

# Stream-Processing für ROS2-basierte Anwendungsentwicklung

## Zielsetzung

ROS2-basierte Applikationen durch den Einsatz von Stream-Processing zu vereinfachen

Literatur-Recherche

Erstellung einer Integrationsplattform

Bewertung von Integrationskonzepten

Umsetzung eines industrienahen Szenarios

## Herausforderungen

### ROS2

- Networking
- Multi-Robot-Communication
- Turtlebot4-Architektur



### Stream-Processing

- Implementierungsoverhead bei Serialisierung



### Bridge

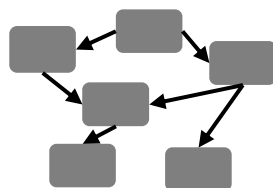
- Transformation von Datentypen
- Topic-Mapping

## Vorteile

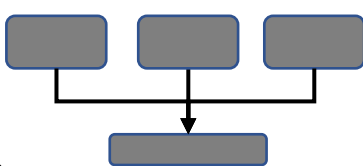
### Schnelle Entwicklung



### Abbilden von Abläufen



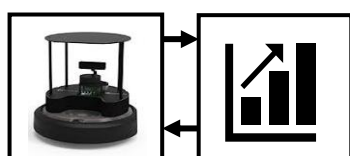
### Sensor-Fusion



### Kooperatives Arbeiten



### Decoupled Development

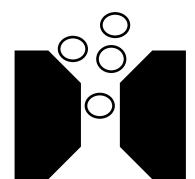


## Nachteile

### Erhöhte Latenz



### Bottleneck (Server)



### Konfigurationsaufwand



INM Forschungsprojekt SS 23  
Alexander Grüneberg, Alexander Mattes,  
Lukas Mendel, Julian Sobott

# Stream-Processing für ROS2-basierte Anwendungsentwicklung

## Technische Umsetzung

ROS2



### Registrierung

- Mechanismus zum Anmelden und Initialisieren von Robotern im System
- Dynamisches, einfaches Hinzufügen von Robotern zum System

### Wiederholung von Nachrichten

- Knoten, der unveränderte Befehle wiederholt versendet
- Reduziert Anzahl der gesendeten Nachrichten

### Weiterleitung von Nachrichten

- Knoten, die Nachrichten vom Server and die richtigen Roboter weiterleiten
- Server muss nicht wissen, wo diese sind

ROS-Kafka Bridge

### Konfiguration (Mapping)

- Welche ROS-Topics werden auf welches Kafka-Topic abgebildet und umgekehrt

### Transformation von Datentypen

- ROS-Nachrichten sind typisiert
- ROS-Datentypen werden in generische Repräsentation überführt, um lose Kopplung zu fördern



ROS-Nodes

Cliff Sensor  
Bumper Node  
LED Node

ROS-Topics

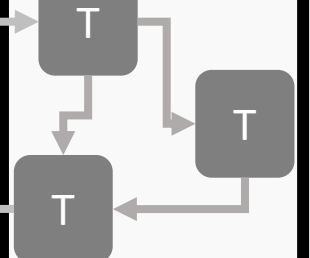
Cliff Distances  
Bumper Status  
LED Commands

ROS-Kafka Bridge

Kafka-Topics

Alert Sensors  
LED Commands

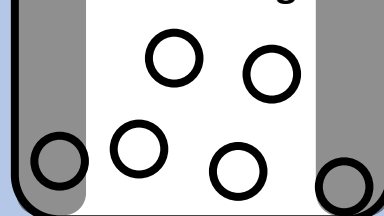
Kafka-Streams



## Complex-Event-Processing mit Kafka Streams

- Auswerten von Sensordaten
- Ableiten des Roboterstatus
- Ableiten von Befehlen für die Roboter
- Nutzen von Aggregationen
  - Windowing: Aktualitätsprüfung
  - Group By: Trennen von geteilten Datenströmen
  - Averaging: Fehlerhafte Sensordaten entfernen
  - Join: Zusammenführen von Sensor-Streams verschiedener Roboter

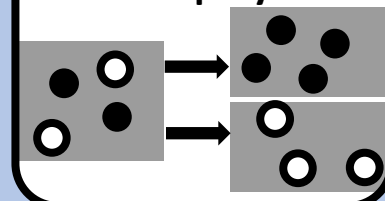
### Windowing



### Averaging



### Group By



### Join

