

Carl von Ossietzky  
Universität  
Oldenburg

**BTC**

*embedded  
systems*

Abt. Foundations and Applications of Systems of Cyber-Physical Systems

Abt. Verteilte Regelung in Vernetzten Systemen

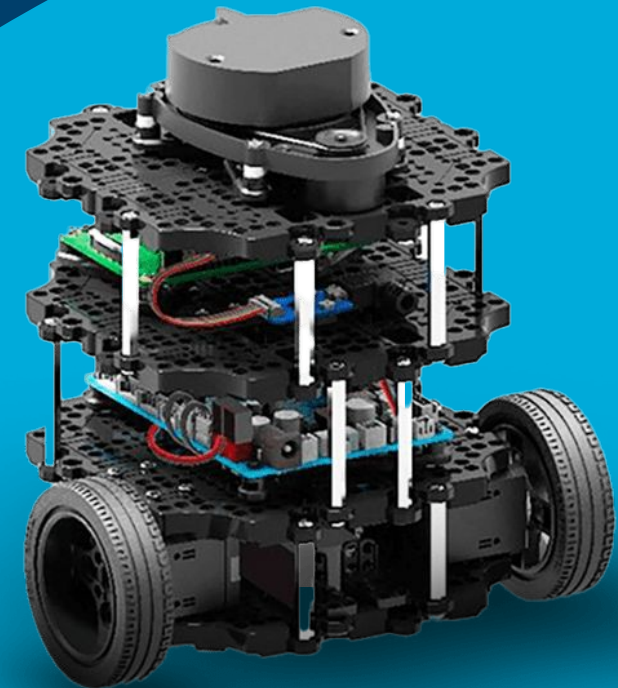


# iTraffic with TurtleBots

Testbasierte Validierung Autonomer  
Fahrfunktionen

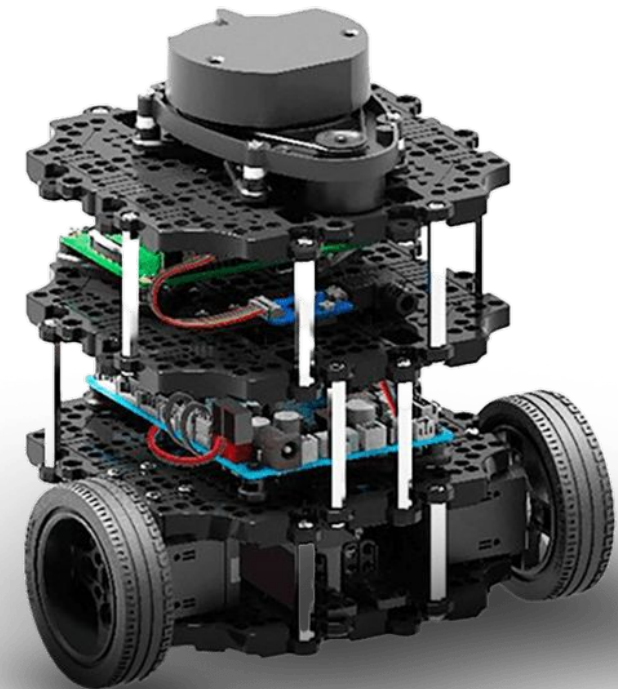
Carl Schneiders, Filip Wojciak, Jan-Magnus Monenschein,  
Julia Debkowski, Lasse Heckelmann, Malte Grave,  
Marie Marken, Nellson Eilers, Paulina Kowalska,  
Stefan Gerber, Simon Struck

28.09.2023



# Agenda

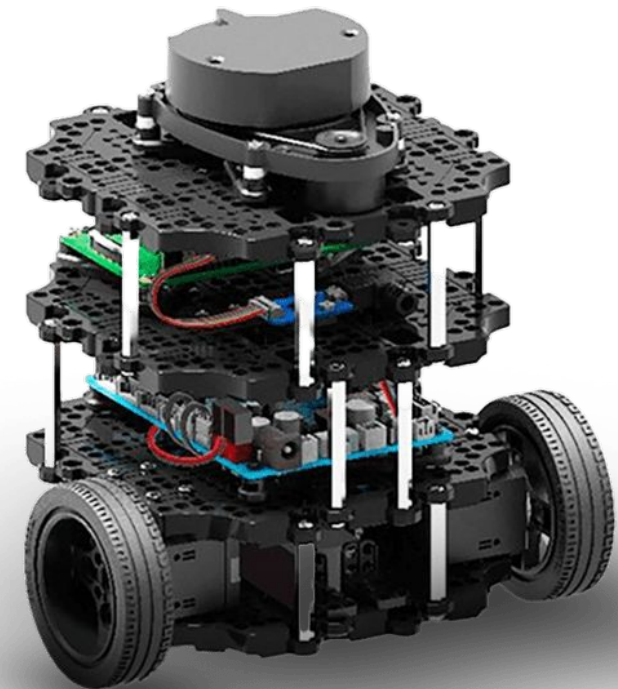
1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion



1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

# Vision

Marie Marken



## Projektziele

- Bewältigung der Herausforderungen autonomen Fahrens
- Simulation unterschiedlicher Autonomiestufen bei gleicher Hardware

Keine Autonomie	Teilautomatisiert	Hochautomatisiert
manuelle Steuerung	Fahrfunktionen <ul style="list-style-type: none"><li>● Tempomat</li><li>● Spurhalteassistent</li><li>● Abstandsregeltempomat</li><li>● Spurwechselassistent</li><li>● Hindernisumfahrung</li><li>● Überholvorgang</li></ul>	Verkehrsschildererkenung <ul style="list-style-type: none"><li>● Geschwindigkeitsbegrenzungen</li><li>● Überholverbote</li></ul> Selbstinitiierte Ausführung der Funktionen

- Entwicklung sicherer und reaktiver Fahrstrategien in Hinblick auf “böartige” Verkehrsteilnehmer
- Experimentelle (simulative) Validierung von Fahrfunktionen in Verkehrsszenarien

## Vision

### Test-Plattform

- Aufbau einer Testplattform für autonome Fahrfunktionen auf Basis von TurtleBots
- Validierung: Methoden zur Definition von Testfällen und deren Ausführung



**Experimentelle Validierung von Fahrfunktionen in Verkehrsszenarien unterschiedlicher Komplexität**

### Entwicklung von Fahrfunktionen

- Aufbau einer Entwicklungsplattform
- Regelungstechnisch-orientierte Umsetzung

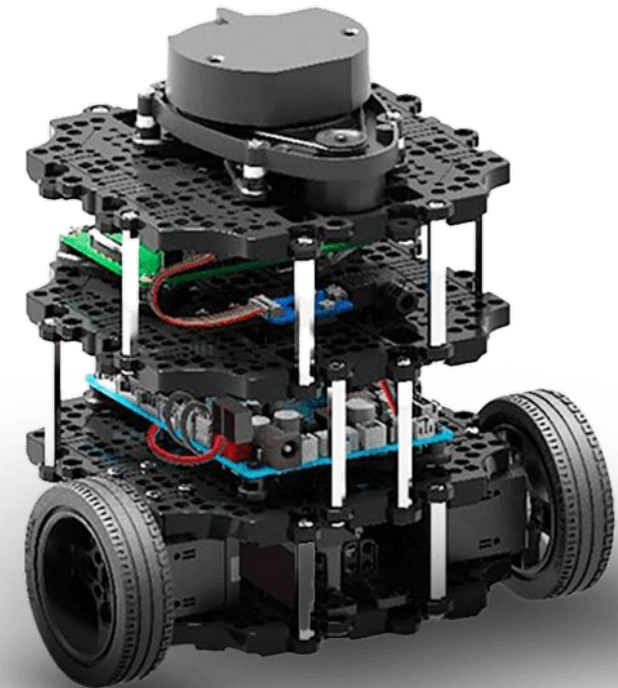


**Bereitstellung der Fahrfunktionen zur Autonomisierung**

1. Vision
2. **State of the Art**
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

# State of the Art

Marie Marken



1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

## Weitere Projektgruppen



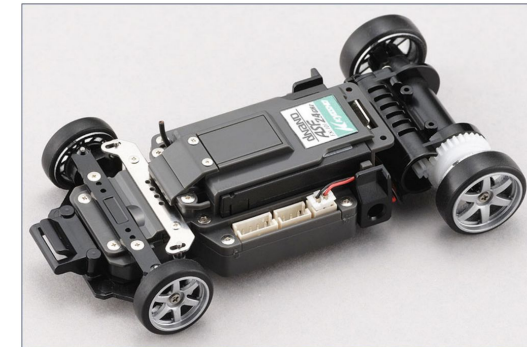
**Umgebung:**  
Rennstrecke

**Szenarien:**  
Unfallfreie Fahrt. Autonome  
Befahrung einer Rennstrecke durch  
ein Fahrzeug.

- Aufbau einer initialen Rennstrecke
- Entwicklung einer Fahrzeugerkennung
- Autonomen Längs- und Querführung für ein Fahrzeug

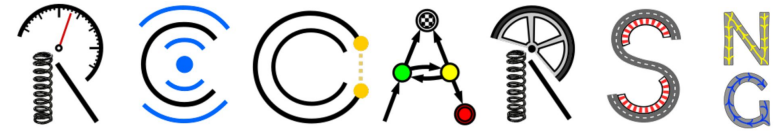


Rennstrecke *Mini-Z Grand Prix Circuit 30*, Marke *Kyosho*



RC-Car Marke *Kyosho*

## Weitere Projektgruppen



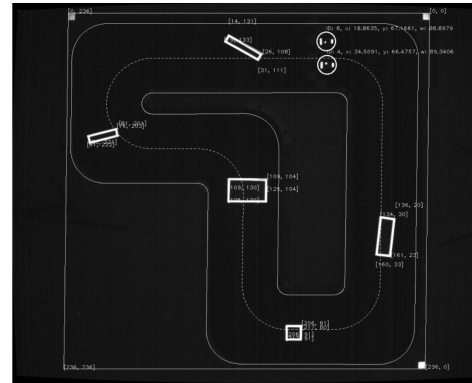
Realtime Controlled Cooperative Autonomous Racing System next generation

### Umgebung: Rennstrecke

### Szenarien:

Unfallfreier Überholvorgang.  
Autonome Befahrung einer  
Rennstrecke durch zwei  
Fahrzeuge.

- Folgen anderer Fahrzeuge
- Überholvorgang
- Hindernis-Integration



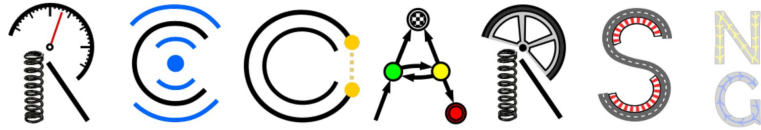
Infrarotaufnahme der  
Rennstrecke mit  
Hindernissen und  
Fahrzeugen



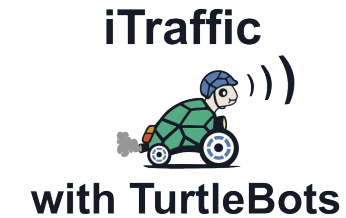
RC-Car mit Markierung



## Abgrenzung zu unserer Projektgruppe



Realtime Controlled Cooperative Autonomous Racing System next generation



- Globales Wissen und externe Berechnungen
- Vorhergehende Trajektorienberechnung
- Statische Hindernisse
- Matlab, C++
- Kamera
- Rennstrecke
- Lokales Wissen und lokale Berechnungen
- Dynamischer Überholvorgang
- Statische und dynamische Hindernisse
- Python
- Kamera, Lidar
- Autobahn

## Weitere Projektgruppen

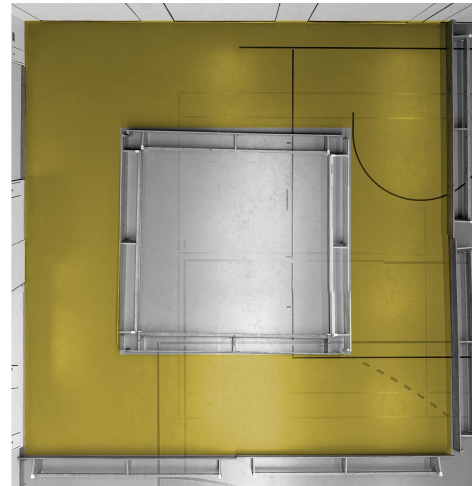


### Umgebung: Parcours

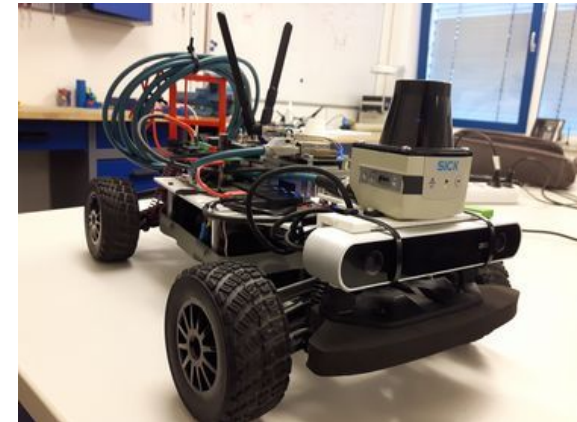
### Szenarien:

Unfallfreie Fahrt. Autonome  
Befahrung eines Parcours durch  
ein Fahrzeug.

- Entwicklung eines echtzeitfähigen Fahrzeugassistenten
- Entscheidungsfindung: Ausweichstrategie oder Notbremsung



vordefinierter Parcours



ferngesteuertes Fahrzeug der Marke Traxxas

## Abgrenzung zu unserer Projektgruppe



- ROS1
- Konstante Sollgeschwindigkeit
- Einen Fahrzeugassistenten
- Lidar
- Parcours

iTraffic



with TurtleBots

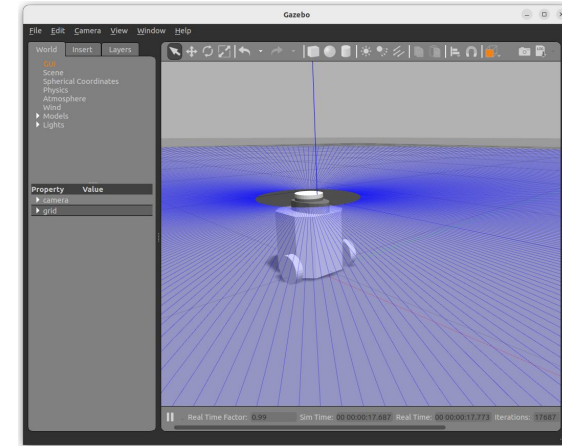
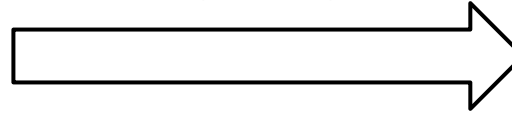
- ROS2
- Variable Geschwindigkeit
- Mehrere Fahrfunktionen
- Lidar, Kamera
- Autobahn

# Vorprojekt: "Forschendes Lernen" - Mobiles Multiagenten-Robotersystem

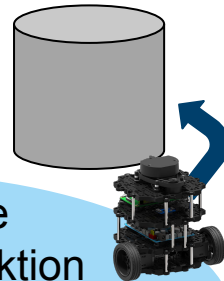
1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion



GAZEBO



**Simulink**



"Wie kann eine autonome Fahrfunktion zum Umfahren eines Hindernisses entwickelt werden?"

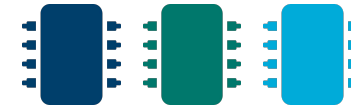
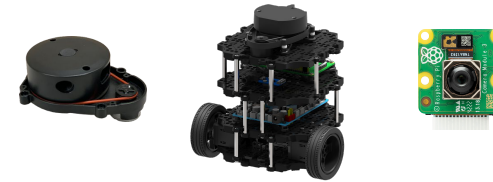
**Python**



"Wie wird realistisches Fahrverhalten simuliert?"

# Unsere Alleinstellungsmerkmale

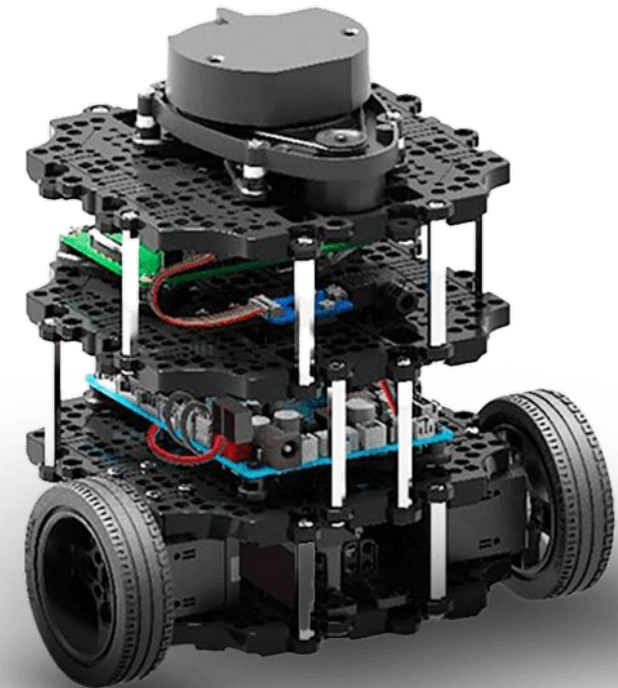
- Kombination mehrerer Sensoren
  - Erweiterbare Hardware
- Realitätsnähe
  - Umgebung: Autobahn
  - Betrachtung des TurtleBots als Auto
- Modulare Entwicklungsplattform:
  - Austauschbare Fahrzeugmodelle
  - Austauschbare Regler
  - Austauschbare Umgebungen
- Automatisierte Test-Plattform



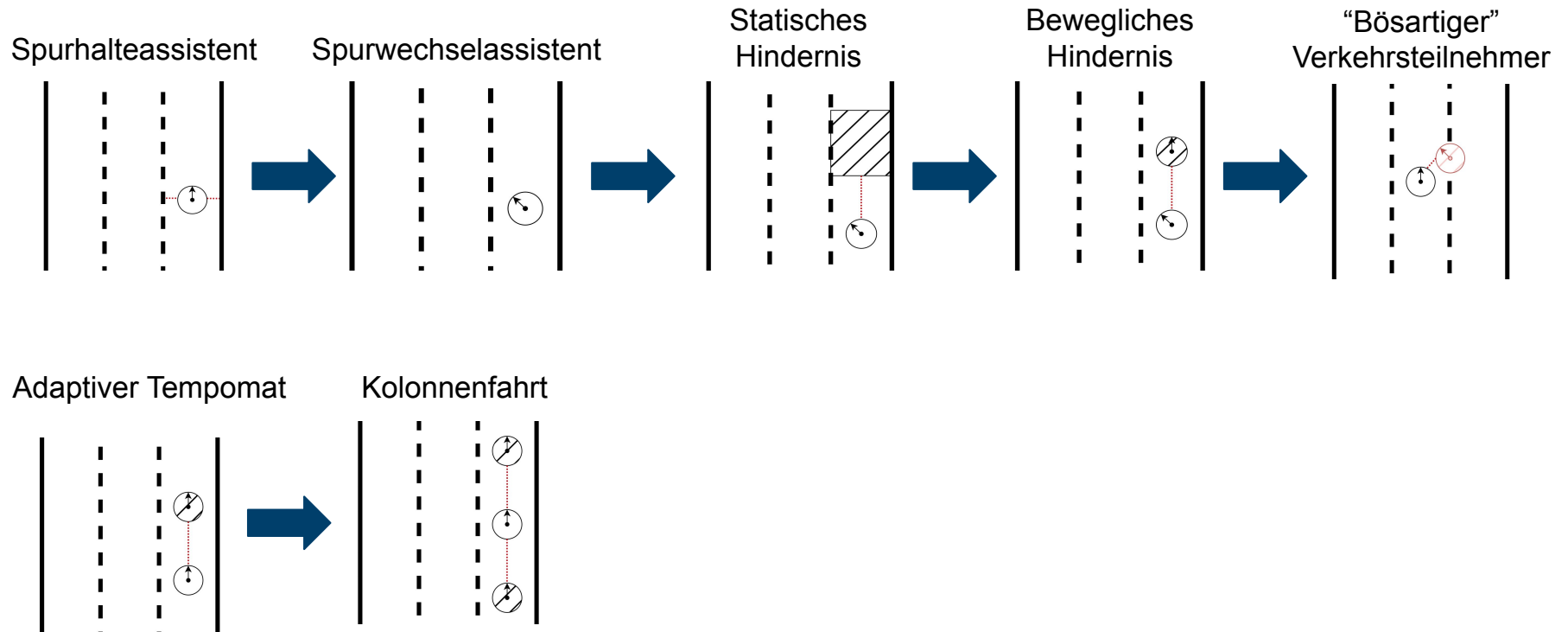
1. Vision
2. State of the Art
3. **Projektmanagement**
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

# Projektmanagement

Carl Schneiders



# Roadmap der Fahrfunktionen



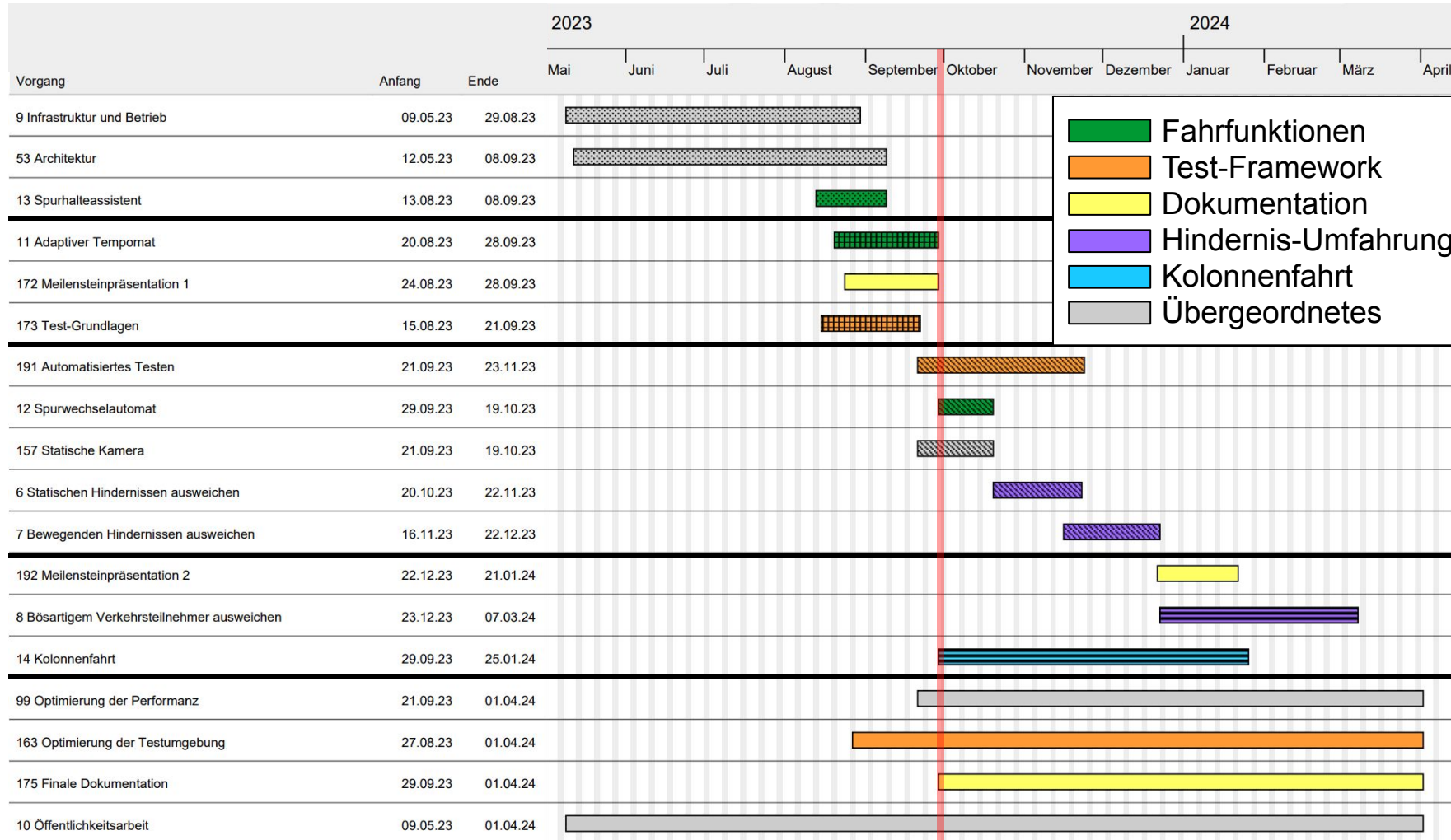
## Zeitplan

Meilenstein	Startdatum	Enddatum
MS 1: Spurhalteassistent und grundlegende Architektur	05. Mai 2023	08. September 2023
MS 2: Adaptiver Tempomat und grundlegende Testbed-Funktionalität	09. September 2023	28. September 2023
MS 3: Hindernisumfahrung, Roboterkamera	29. September 2023	22. Dezember 2023
MS 4: Bössartiger Verkehrsteilnehmer und Kolonnenfahrt	23. Dezember 2023	07. März 2024



- 1. Vision
- 2. State of the Art
- 3. **Projektmanagement**
- 4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
- 5. Ausblick und Reflexion

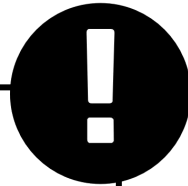
# Zeitplan



## PG iTraffic - Erste Meilensteinpräsentation

Carl Schneiders

## Arbeitsprinzipien



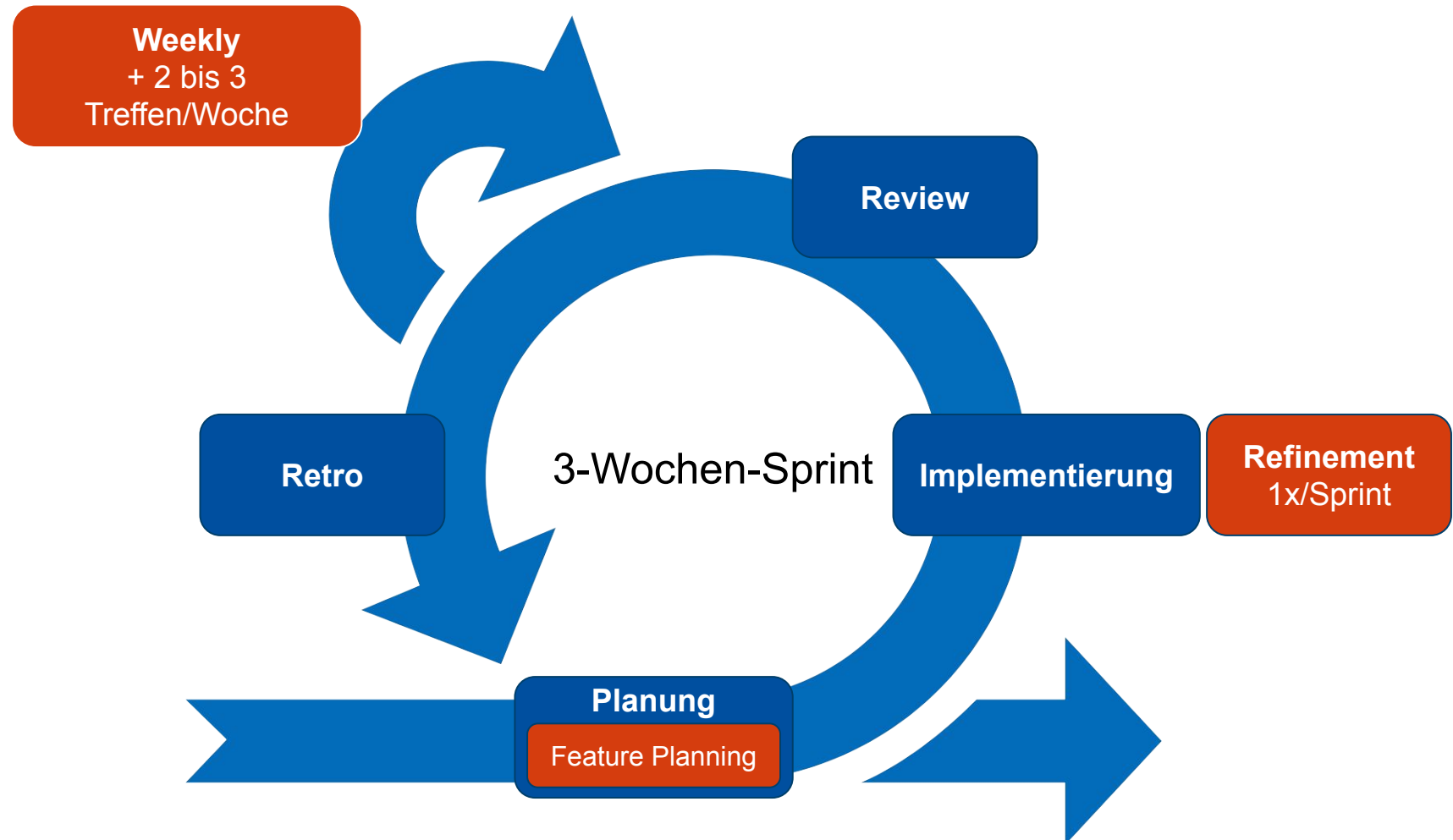
- Gesunde Gestaltung der PG
- Kein Burnout
- Klare und regelmäßige Kommunikation



- Kein Loggen von Zeit
- Review auf Vertrauensbasis
- Urlaub nach Absprache
- Kein Crunch

1. Vision
2. State of the Art
3. **Projektmanagement**
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

## Scrum und Sprints



1. Vision
2. State of the Art
3. **Projektmanagement**
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

# Rollenverteilung

## Fachliche Leitung



**Marie Marken**  
Product Owner



**Lasse Heckelmann**  
Business Engineer

## Entwicklungsteam



**Malte Grave**  
Infrastruktur



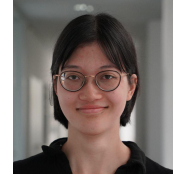
**Jan-Magnus Monenschein**  
Code Stewart



**Filip Wojciak**  
Quality Assurance



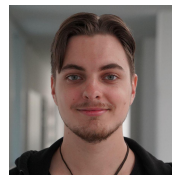
**Stefan Gerber**  
Architektur



**Julia Debkowski**  
Entwicklerin



**Paulina Kowalska**  
Öffentlichkeitsarbeit



**Nellson Eilers**  
Documentation Stewart

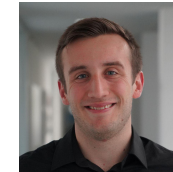


**Simon Struck**  
Technische Leitung



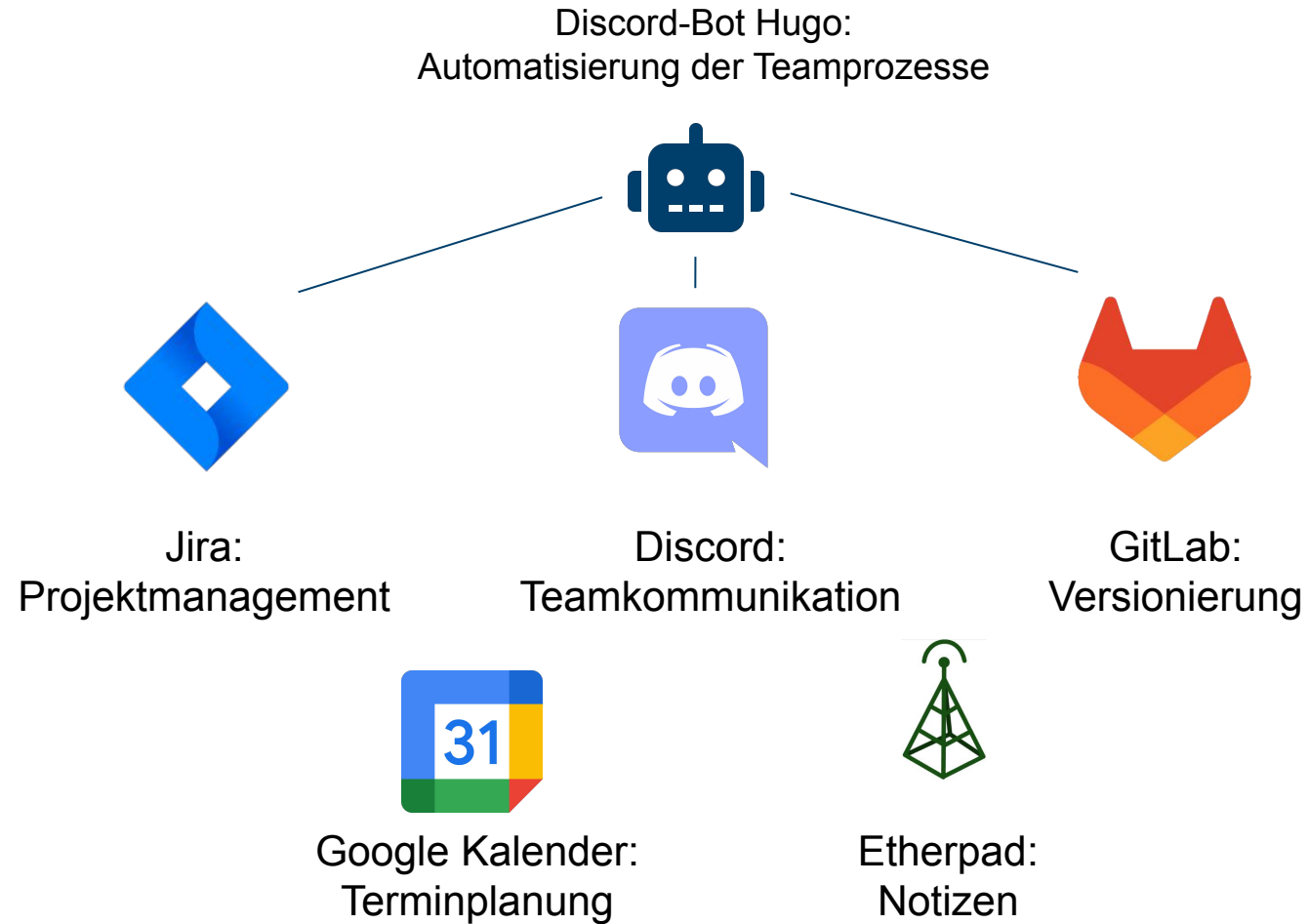
**Marvin Kuhlmann (ehem.)**  
Infrastruktur

## Prozessleitung



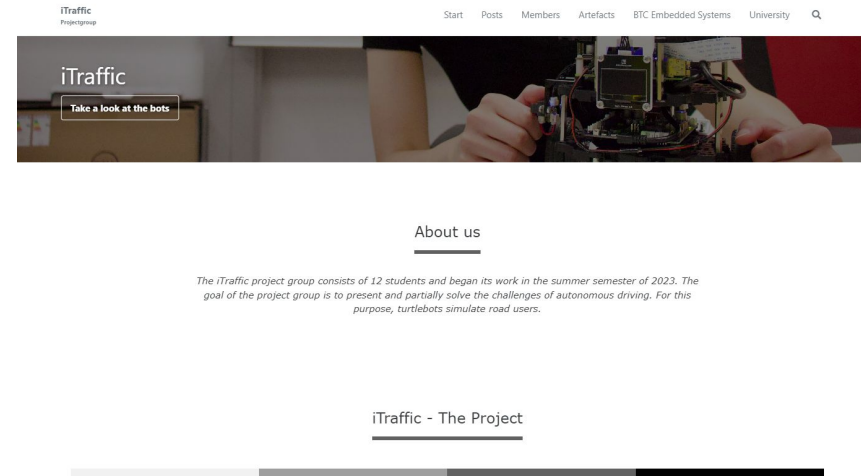
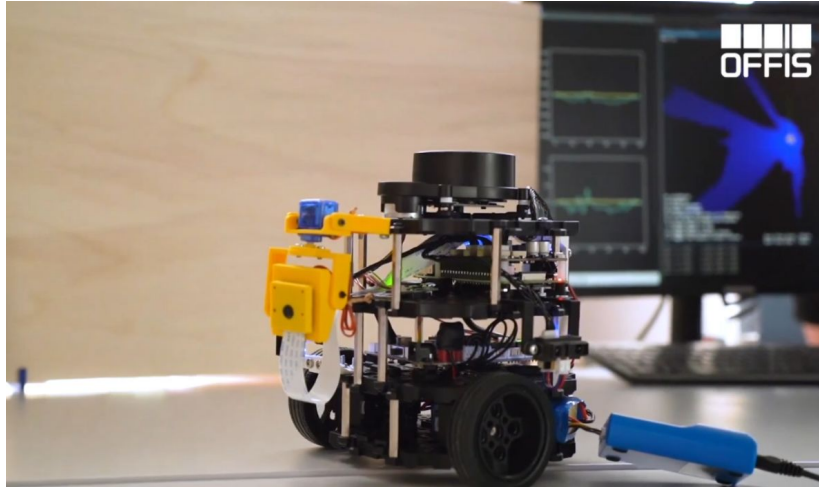
**Carl Schneiders**  
Scrum Master

# Tools



1. Vision
2. State of the Art
3. **Projektmanagement**
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

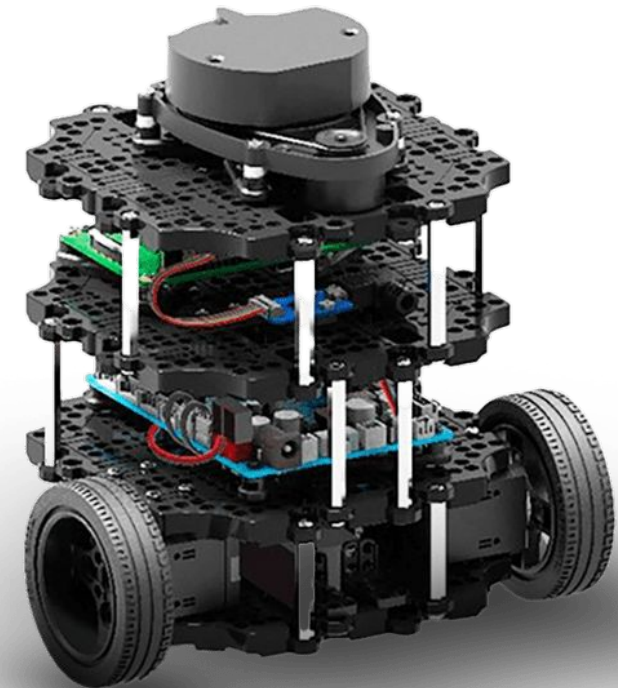
# Öffentlichkeitsarbeit



1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

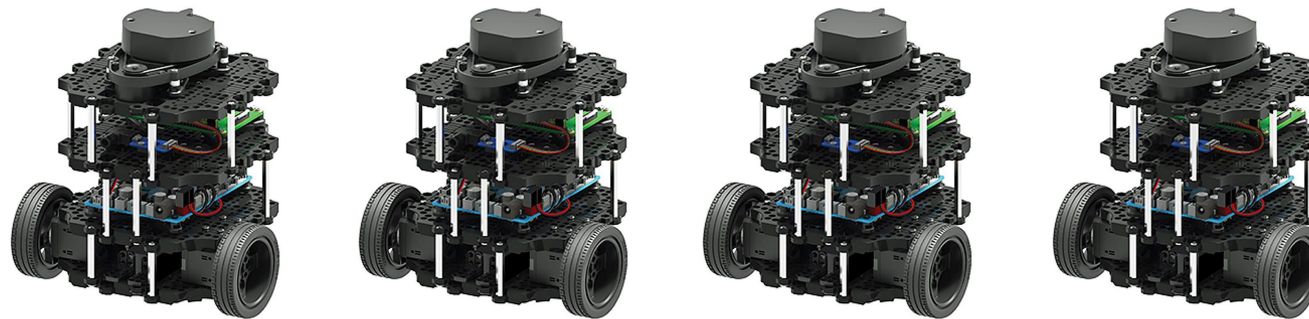
# Umsetzung

Malte Grave  
Jan-Magnus Monenschein  
Nellson Eilers



# Hardware

- TurtleBot 3 Burger Modelle
- Modulares Design
- Sensorik und Aktorik leicht nachrüstbar
- LIDAR, 2x IR, 2x Motoren, 2x Servo, Pi CAM
- 2-Achsen Camera mount

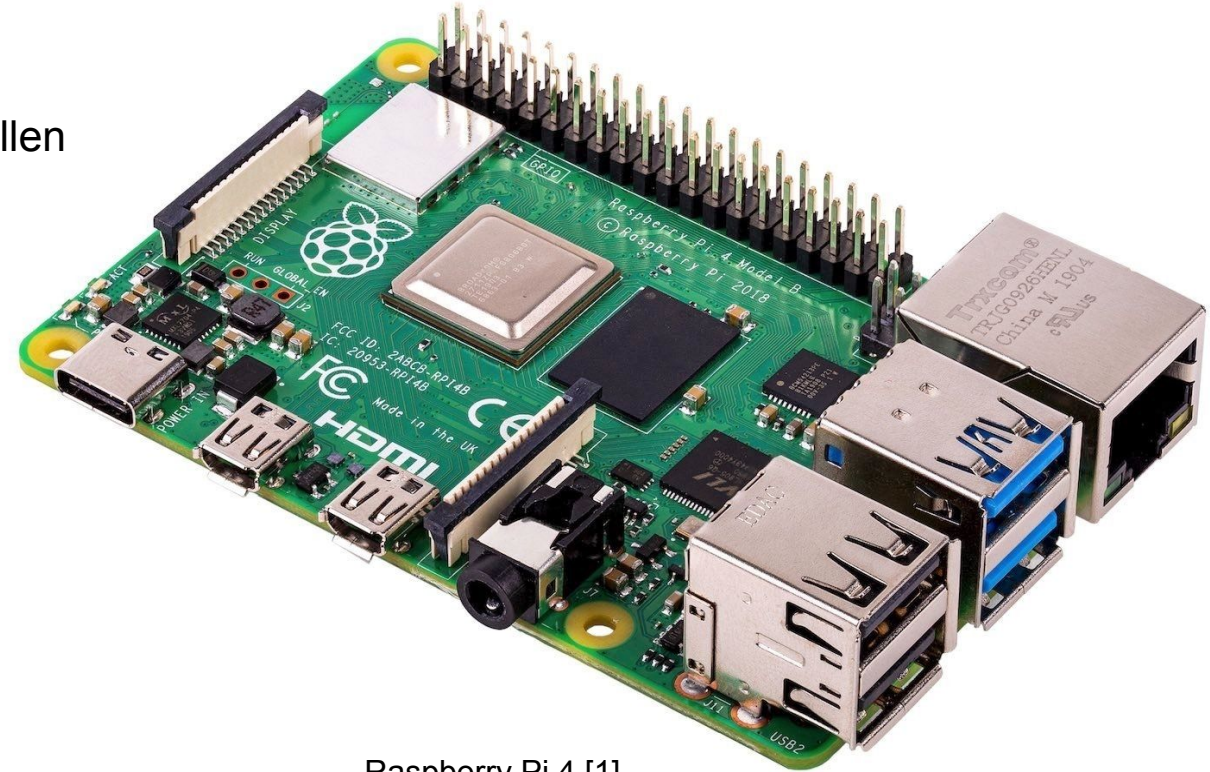


TurtleBot 3 Burger [0]



# Hardware

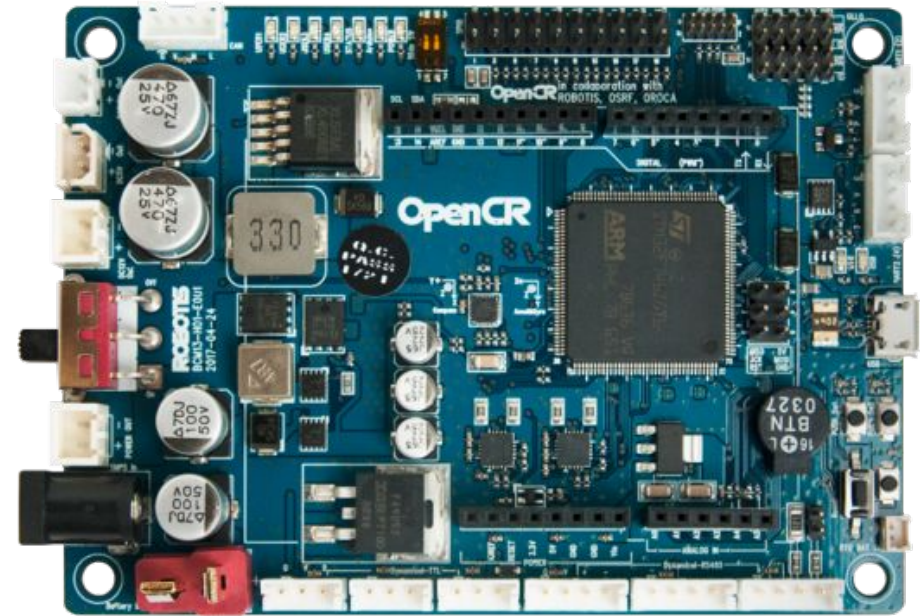
- Raspberry Pi 4 2GB
- Cortex-A72 (ARM v8) 1,5 GHz
- GPIOs, I2C, UART, SPI Schnittstellen
- emBRICK Header für Anschlüsse



Raspberry Pi 4 [1]

# Hardware

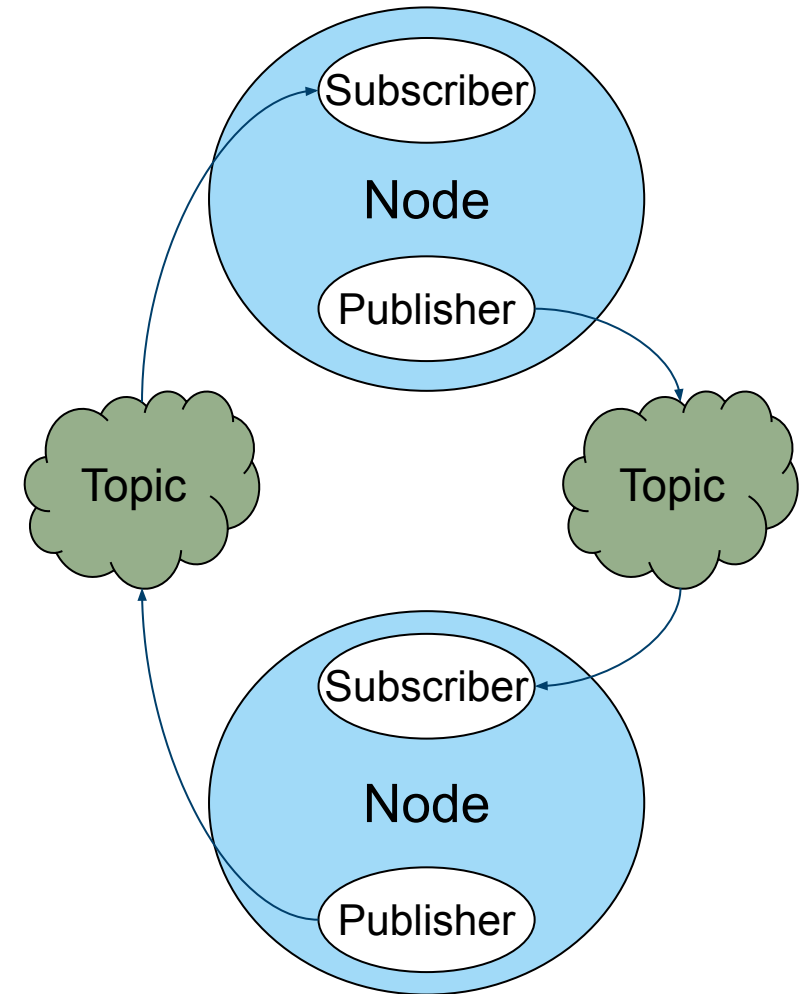
- OpenCR Board 1.0
- Agiert als Treiber
- STM32 / 32-bit ARM Cortex-M7 216MHz
- Kommunikation über USB bzw. UART
- ROS Abstraktionslayer



OpenCR Board 1.0 [2]

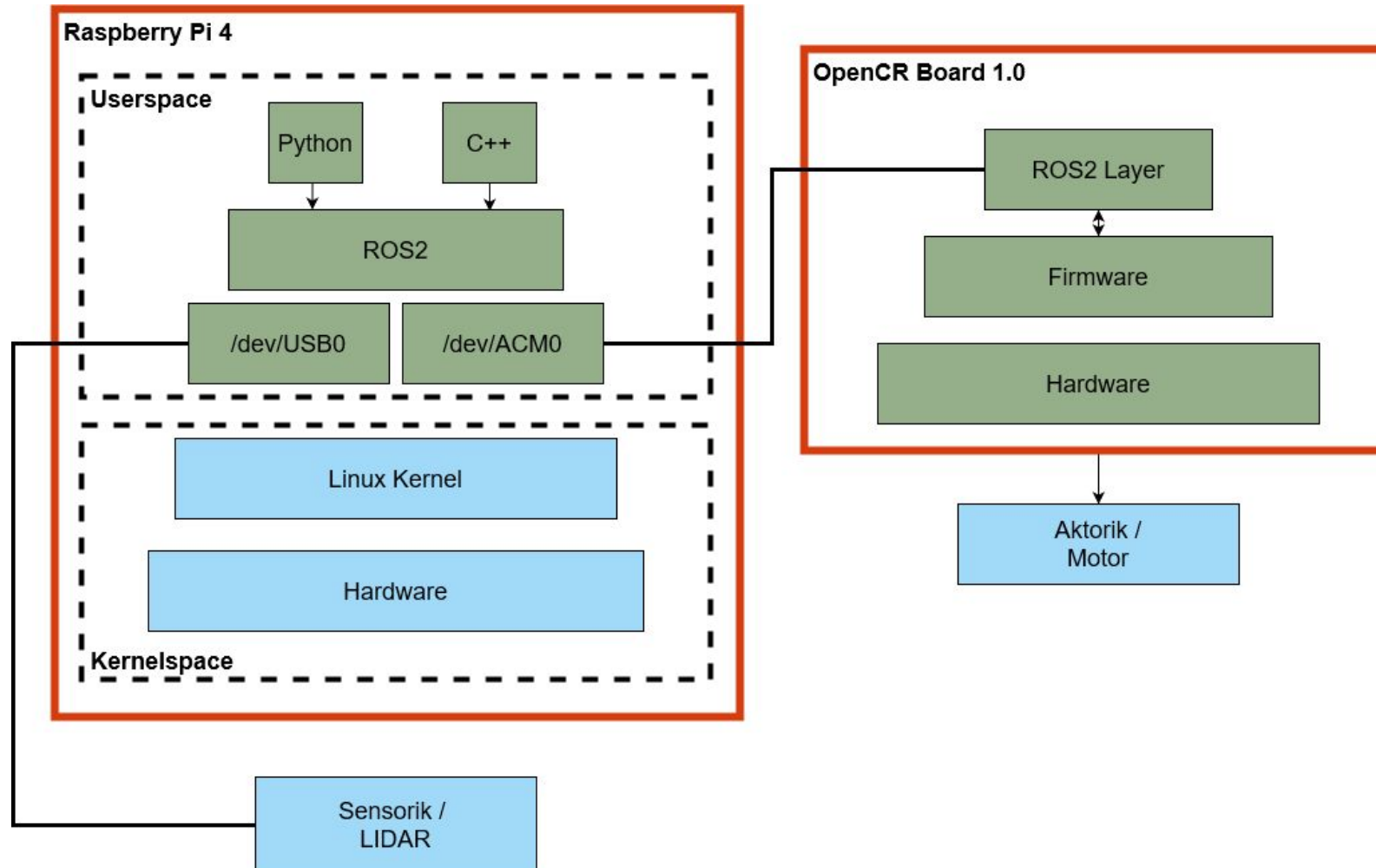
## ROS Grundlagen

- Kommunikation via DDS
  - Topics als zentrales Konzept
  - Nutzt eine Domain ID
  - Insbesondere über Netzwerke
- Publisher und Subscriber greifen darauf zu
  - Werden in Nodes zusammengefasst
- Services und Action als spezielle “Topics”



1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion




# Kommunikation



# MS 1: Spurhalteassistent und grundlegende Architektur

- Infrastruktur und Betrieb
- Implementierung von Fahrfunktionen ermöglichen
- Spurhalteassistent

## Unsere Produkte

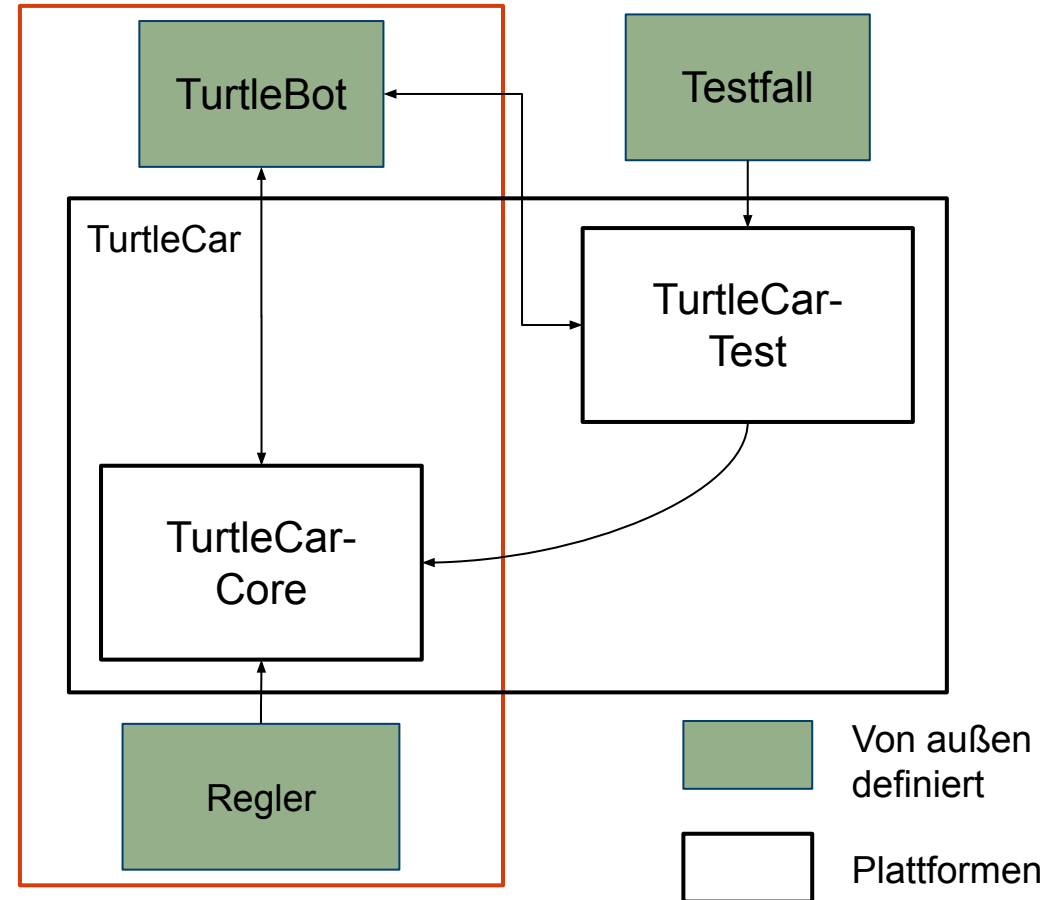
<b>TurtleCar-Core</b> 	<b>TurtleCar-Test</b> 	<b>Fahrfunktionen</b> 
<p>Entwicklungsplattform für autonome Fahrfunktionen</p> <ul style="list-style-type: none"><li>● Emulation eines Autos auf dem TurtleBot</li><li>● Definierte Schnittstelle für Sensoren und Aktuatoren in passenden Formaten</li><li>● Entwicklung von Reglern gegen diese Schnittstelle</li></ul>	<p>Testplattform für autonome Fahrfunktionen</p> <ul style="list-style-type: none"><li>● Einfache Definition von Testfällen</li><li>● Simulatives &amp; automatisiertes Testen von Fahrfunktionen</li></ul>	<p>...entsprechend den Szenarien der Vision</p> <ul style="list-style-type: none"><li>● Entwicklung von Reglern gegen die TurtleCar-Schnittstelle</li><li>● Nutzung von Modellprädiktiver Regelung</li><li>● Ggf. Anbindung neuer Sensorik</li></ul>

# Die Plattform TurtleCar

## Ziele für TurtleCar

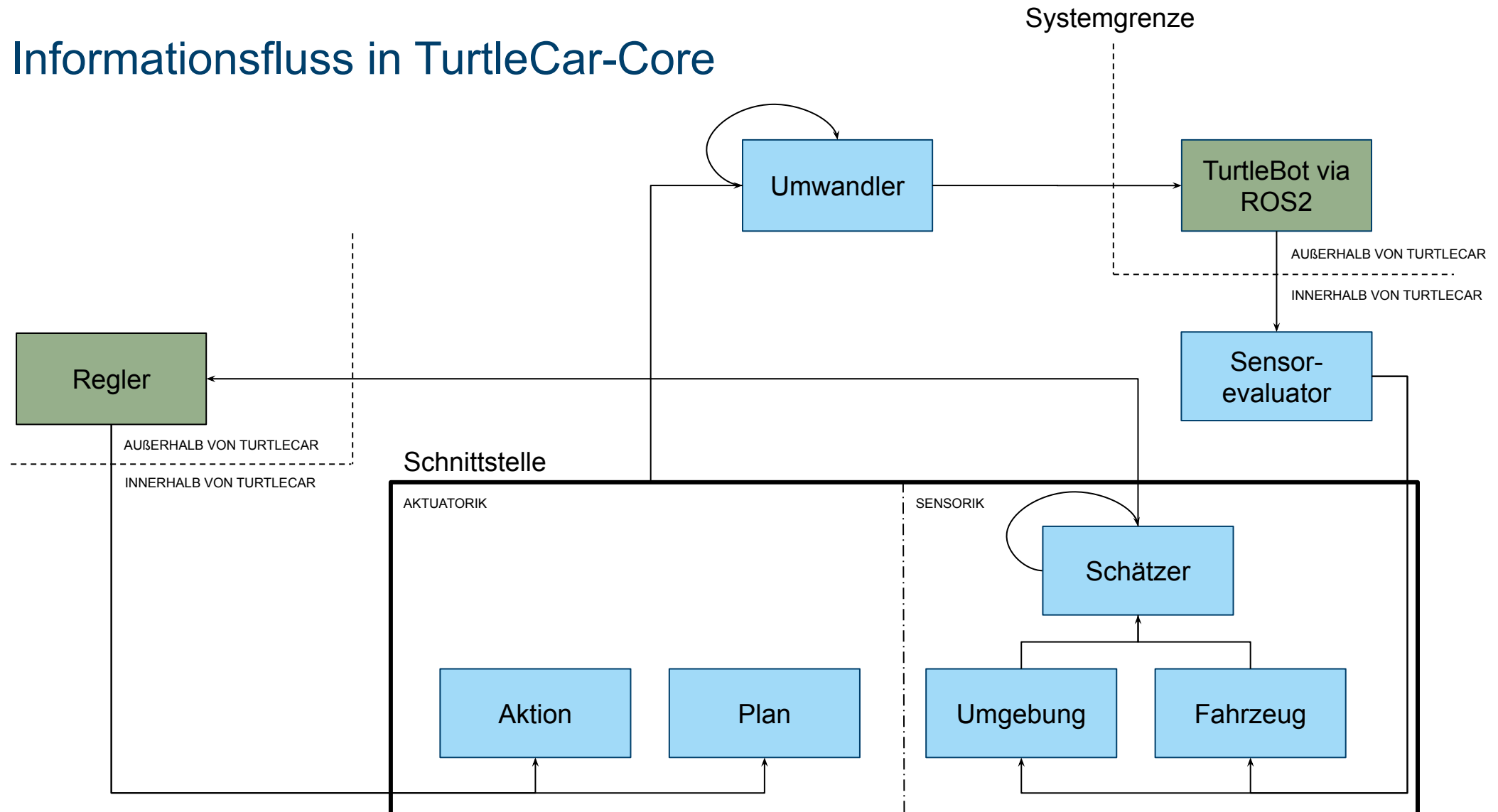
- Bereitstellung von Fahrzeug- und Umgebungsinformationen
- Klare Struktur und Architektur
- Erweiterbarkeit durch Dritte

## Überblick



- 1. Vision
- 2. State of the Art
- 3. Projektmanagement
- 4. Umsetzung
  - a. Hardware
  - b. **Meilenstein 1**
  - c. Meilenstein 2
- 5. Ausblick und Reflexion

# Informationsfluss in TurtleCar-Core

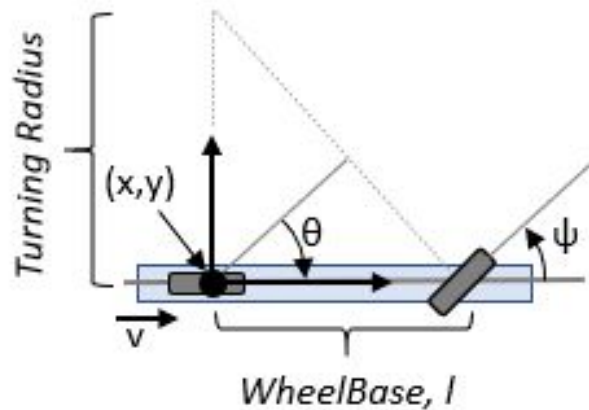




# Fahrzeugmodelle

## Bicycle Modell

- Reduziert Dynamiken auf zwei Achsen
  - Vorderrad: Lenkung
  - Hinterrad: Antrieb
- Berechnungen im Umwandler



[3]

## Simpler Automotor

- Berechnung der Beschleunigung fürs Bicycle Modell
- Geschalteter Gang bestimmt die Antriebskraft des Motors

Antriebskraft:

$$F_t = \frac{T_e \cdot i_x \cdot i_0 \cdot \eta_d}{r_{wd}}$$

$T_e$  = Drehmoment des Motors

$i_x$  = Getriebeübersetzung

$\eta_d, i_0$  = Konstanten aus  
Fahrzeugspezifikationen

$r_{wd}$  = Reifenradius

Beschleunigung:

$$a_v = \frac{1}{m_v} [F_t - (F_r + F_a)]$$

$F_r$  = Reibungswiderstand

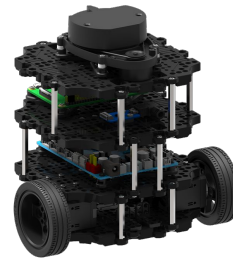
$F_a$  = Luftwiderstand

$m_v$  = Masse des Autos

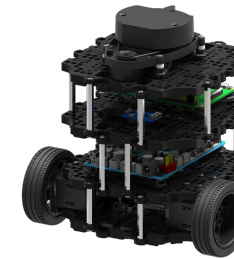
# Fahrzeugmodelle

## Fahrzeugtypen

- Modellierung verschiedener Verkehrsteilnehmer
  - Austauschbar durch Konfigurationsdateien
  - Realistische Szenarien
- Übersetzung der Parameter echter Autos



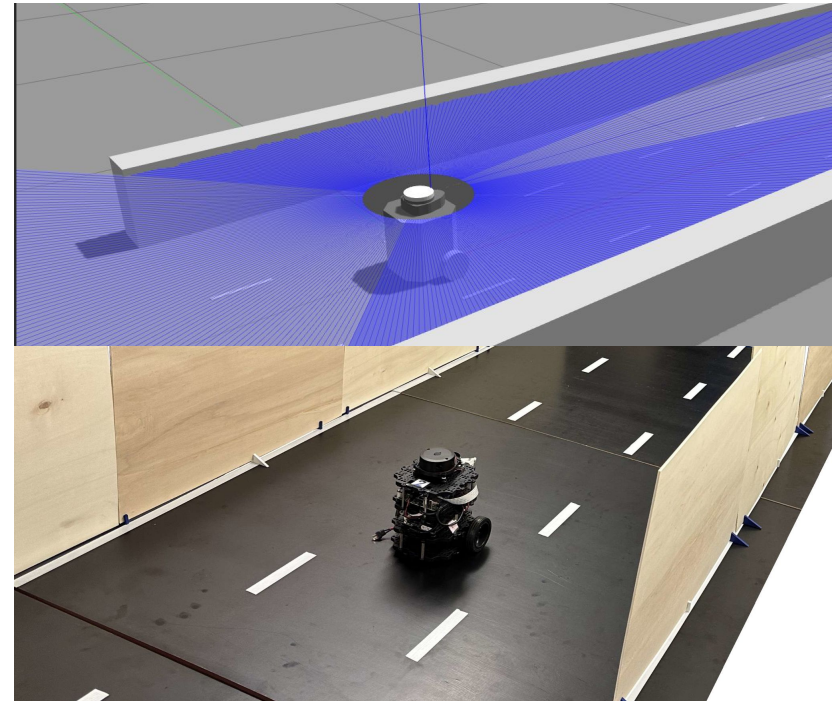
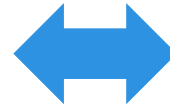
Jaguar F-Type [4]



VW-Golf 7 [5]

1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. **Meilenstein 1**
  - c. Meilenstein 2
5. Ausblick und Reflexion

# Umgebungsmodell



[6]

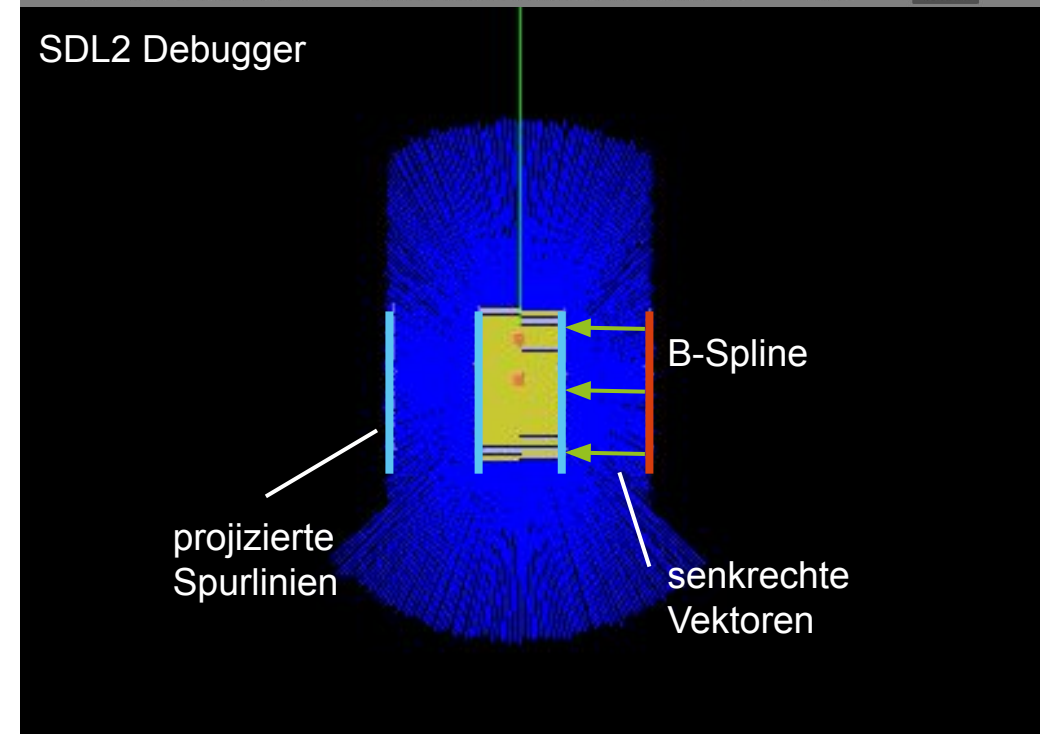
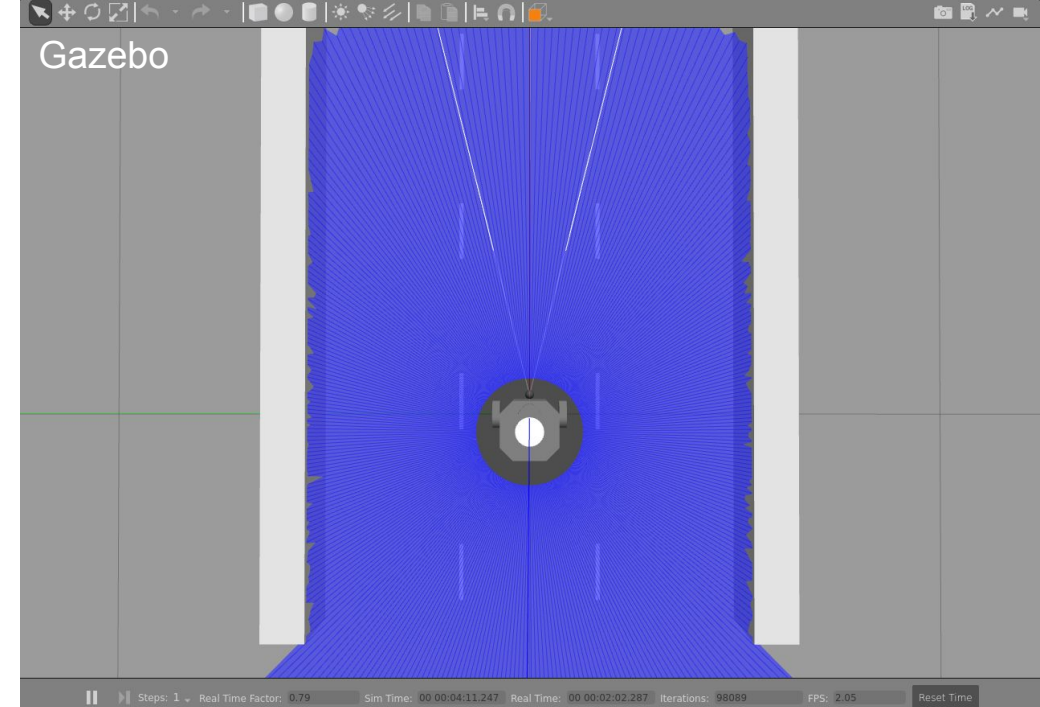
# Spurerkennung mit LIDAR

## Annahme:

- Wand rechts
- Definierte Umgebung wird genutzt

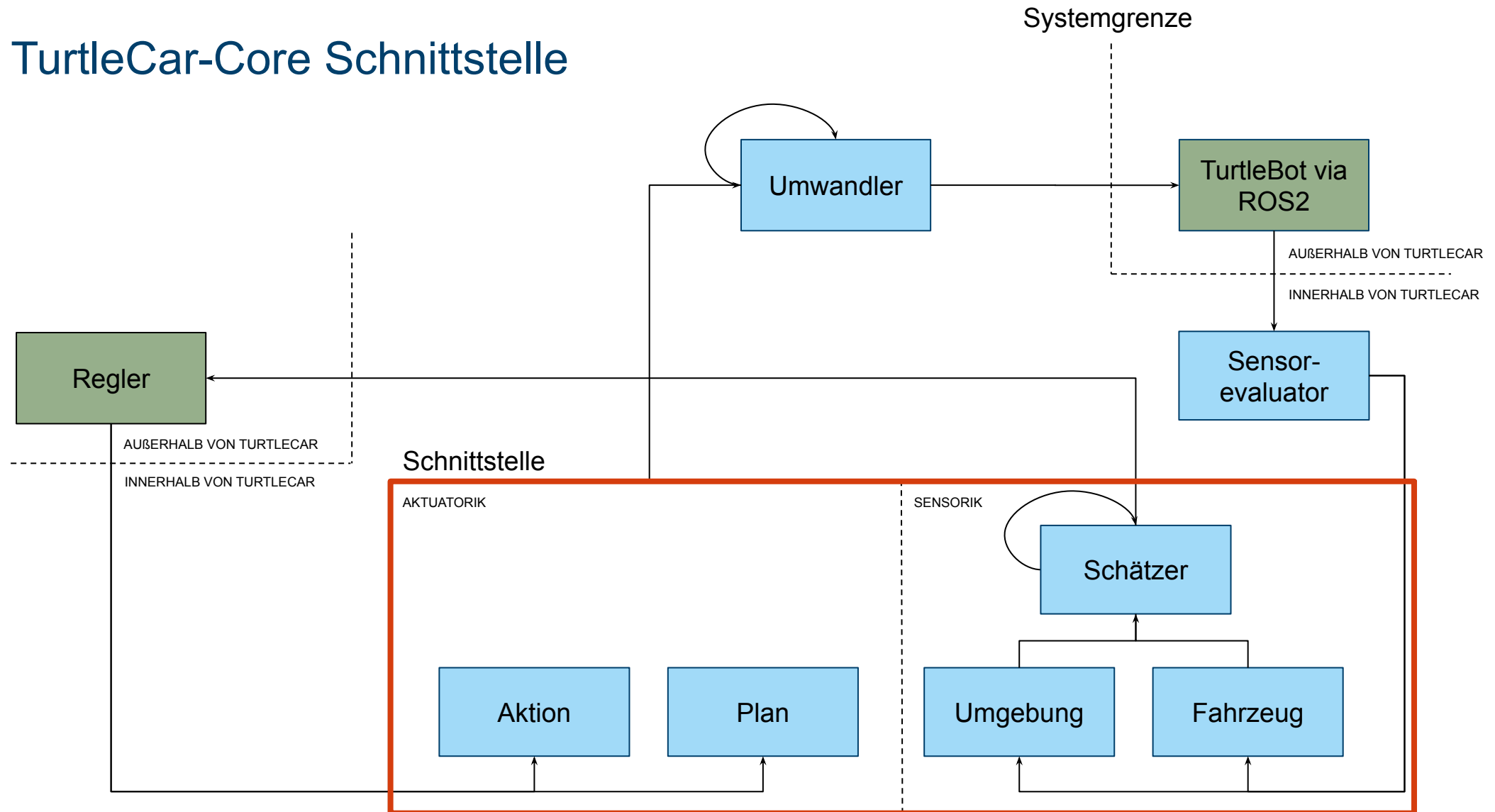
## Funktionsweise:

1. Speicherung von Lidar-Messungen
2. Umrechnung in kartesisches Koordinatensystem
3. Berechnung einer B-Spline
4. Iteration der Messpunkte und je eine Berechnung eines orthogonalen Vektors
5. Projektion weiterer Spurlinien durch Vektoren



1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. **Meilenstein 1**
  - c. Meilenstein 2
5. Ausblick und Reflexion

# TurtleCar-Core Schnittstelle

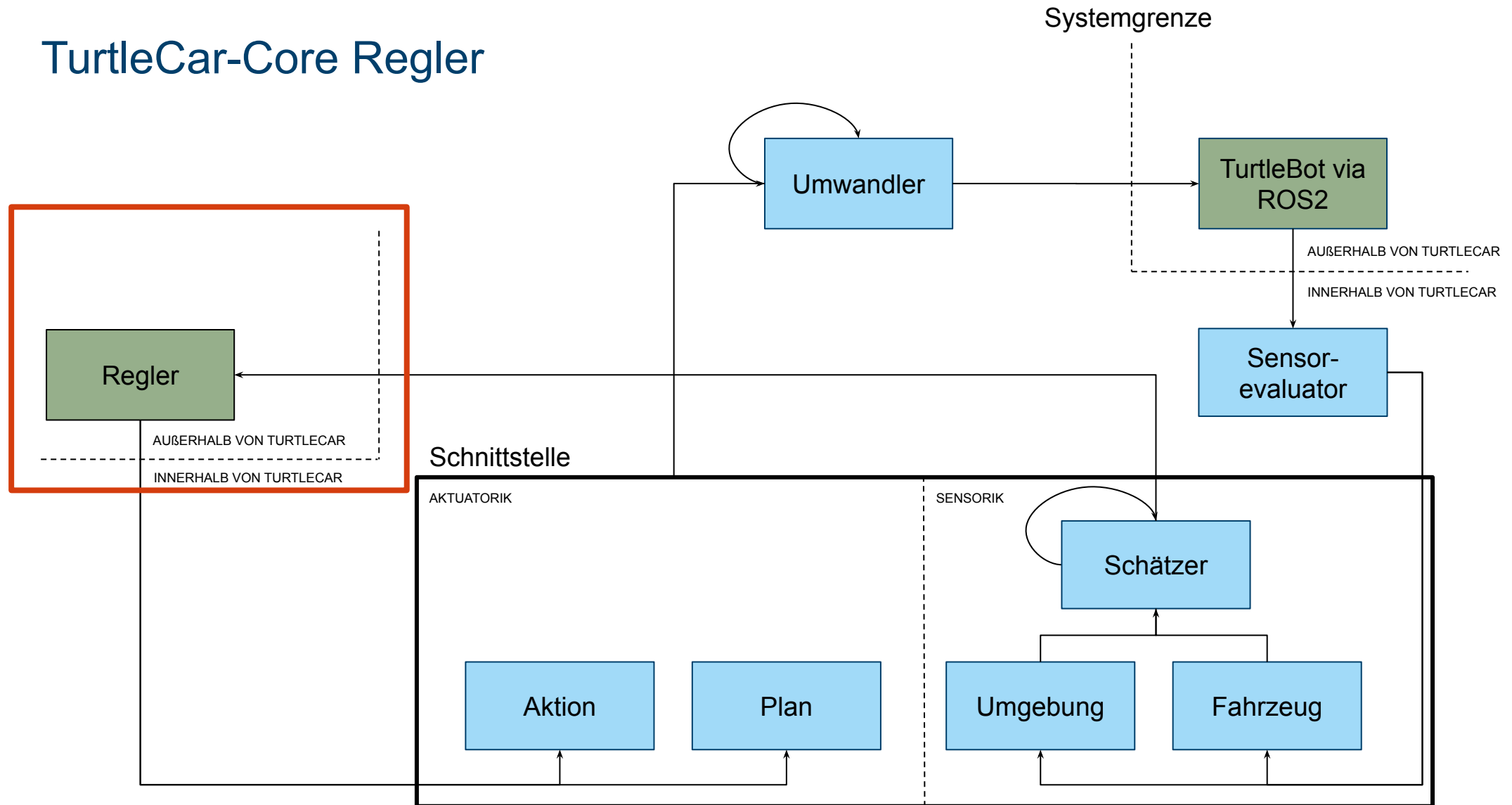


## TurtleCar-Core Schnittstelle

Modul	Verfügbare Daten
Aktion	Schnittpunkt zur Steuerung des Roboters
Plan	Geplanter Pfad in 2D-Punkten
Fahrzeug	<ul style="list-style-type: none"><li>● Drehmoment</li><li>● Fahrwiderstand</li><li>● Geschwindigkeit</li><li>● ...</li></ul>
Umgebung	<ul style="list-style-type: none"><li>● Hinderniskoordinaten</li><li>● Verkehrsschilder</li><li>● Spurdaten</li><li>● ...</li></ul>

1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. **Meilenstein 1**
  - c. Meilenstein 2
5. Ausblick und Reflexion

# TurtleCar-Core Regler



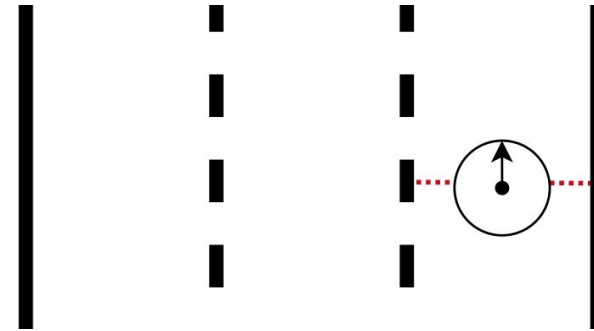
# Spurhalteassistent

## Funktionale Anforderungen:

- Gezielt ein- / ausschalten
- Spurverlauf identifizieren
- Spurmitte folgen

## Nicht funktionale Anforderungen:

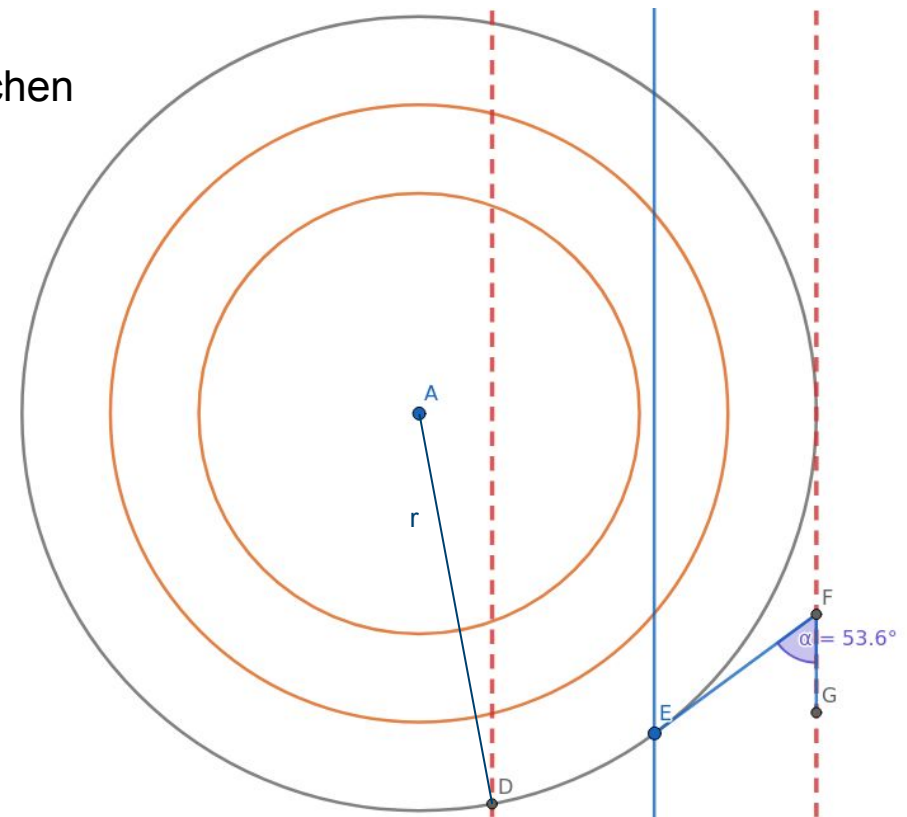
- Modellvorhersage nutzen
- Anwendbar für Geschwindigkeiten von 0 bis 120 km/h





## Initialbedingungen

- Engster möglicher Wendekreis:  $r = \frac{l}{\tan(a)} + \frac{b}{2}$
- Anlegen des Wendekreises an den Fahrbahnrand
- Maximale Lenkorientierung ist der Winkel zwischen Kreis-Tangente und Fahrbahnrand
  - Für Mitte der Fahrbahn:  $53,6^\circ$
- 25% zusätzlicher Spielraum
  - Für Mitte der Fahrbahn:  $40,2^\circ$



# Modellprädiktive Regelung

## Bicycle-Modell

$$\dot{x}_1 = \dot{Y} = v \cdot \sin(x_2)$$

$$\dot{x}_2 = \dot{\theta} = \frac{v}{l} \cdot \tan(u_1)$$

## Linearisierung

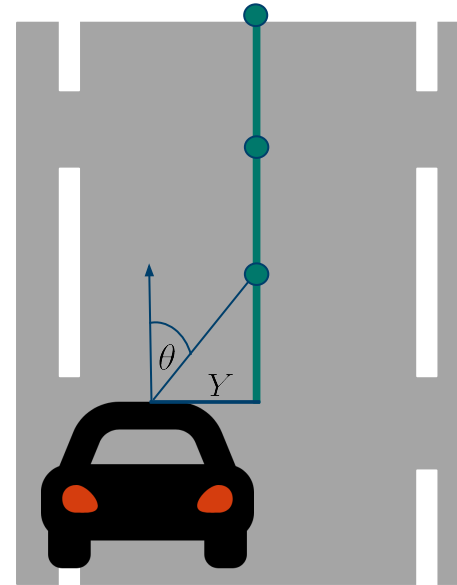
$$Y_s = 0, \theta_s = 0$$

$$\dot{x} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix} \cdot x + \begin{bmatrix} 0 \\ \frac{v}{l} \end{bmatrix} \cdot \left( - [k_1 \quad k_2] \cdot \begin{bmatrix} Y \\ \theta \end{bmatrix} \right)$$

## Eigenwertplatzierung

$$\mathcal{R}\{\lambda\} \leq -2$$

für robuste Stabilität bei variabler  
Geschwindigkeit  $v$



## Nutzungsbereich

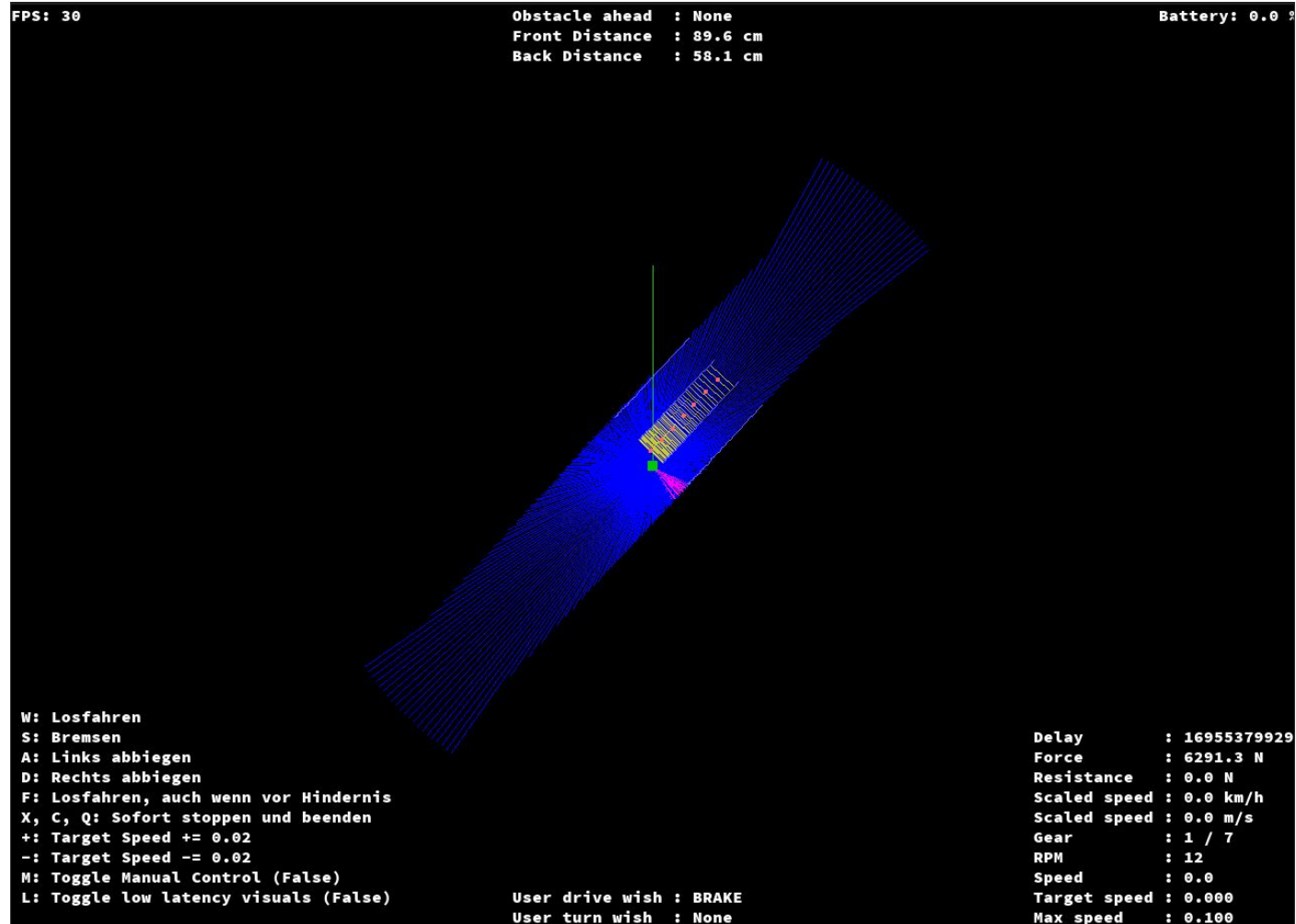
$$v \in [0.5, 34] \text{ m/s} \Rightarrow k_2 \in [0.05, 1.4]$$

## Gewählte Regelkonstante

$$k_2 = \frac{l}{v}$$

1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

# Spurhalteassistent



# MS 2: Adaptiver Tempomat und grundlegende Testbed Funktionalität

- Ausführung von Integrationstests über das Testbed
- Umsetzung der Fahrfunktion: “Adaptiver Tempomat”

# Validierung von Fahrfunktionen

## Simulativ:

- Einfaches Setup
- Fokus auf Realitätsnähe

## Experimentell:

- Manuelles Testen in der physischen Realität
- Visuelle Evaluation

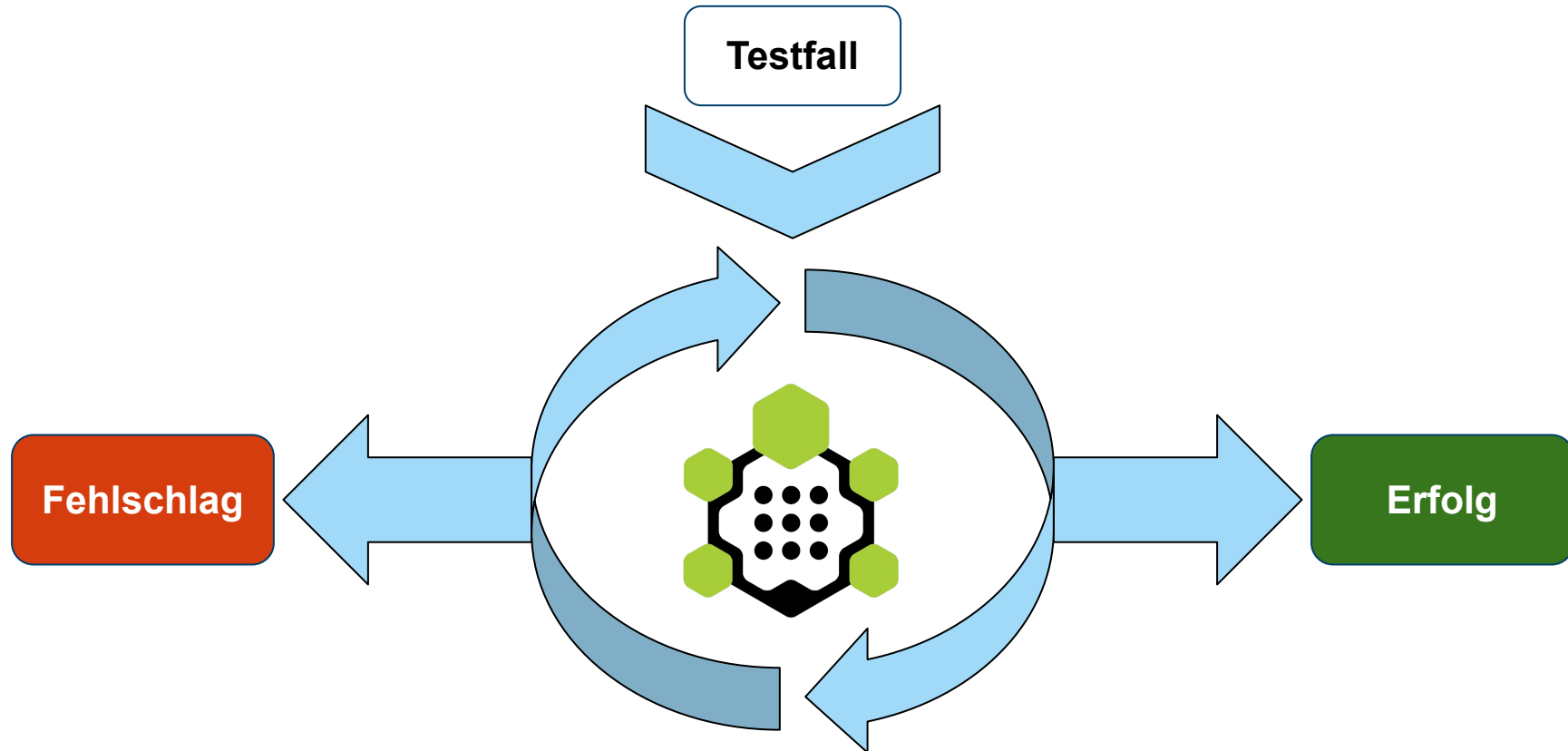
Evaluation der  
erfüllten  
Anforderungen

## Formale Tests:

- Integration in Szenarien
- Definition von formalen Kriterien
- Auswertung einzelner Testfälle

Validierung der  
erfüllten  
Anforderungen

# Grundlagen Test-Plattform



# Definition von Testfällen

## Startbedingungen:

- Anzahl Roboter
- Startposition / -rotation
- Fahrzeugmodell

## Eingabebefehle:

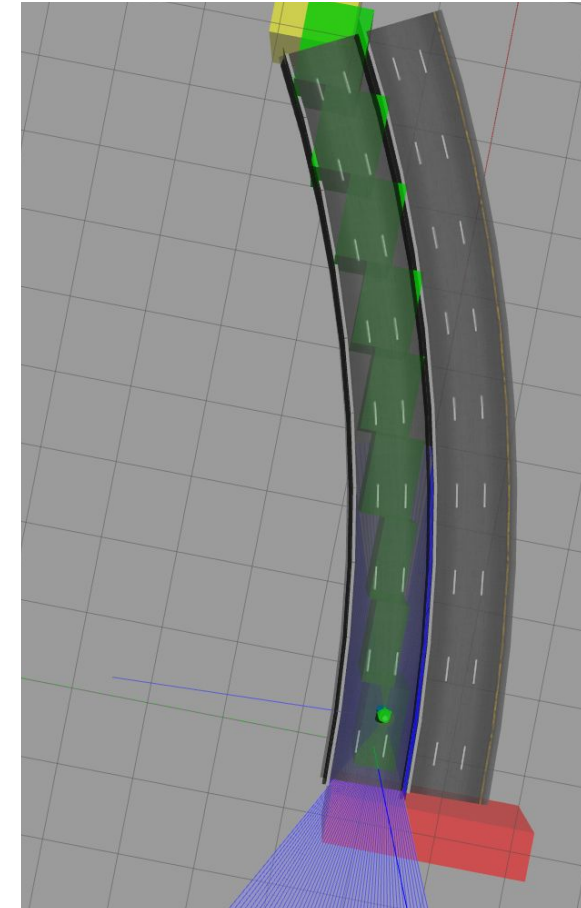
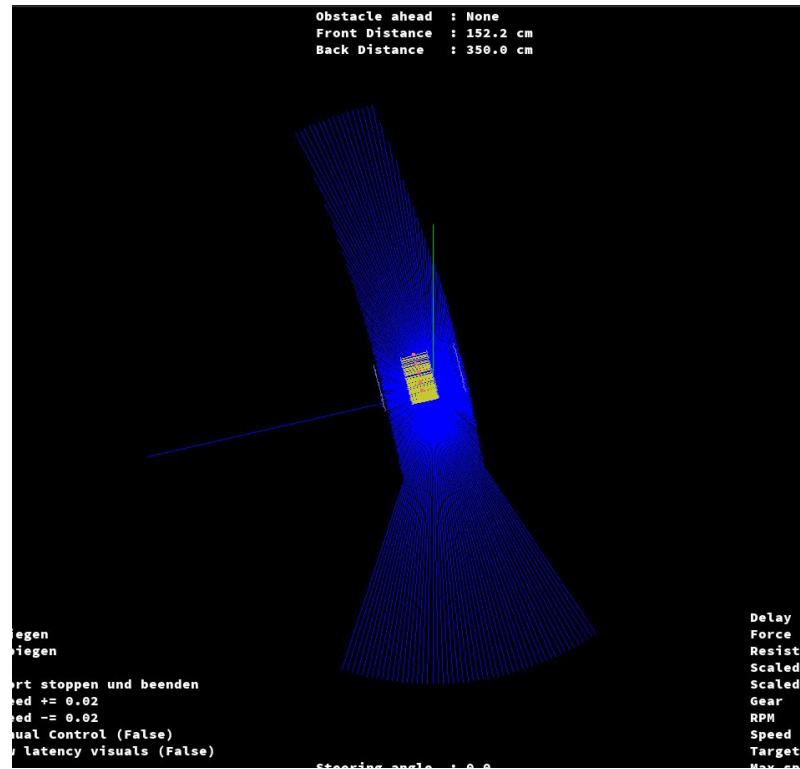
- Beschleunigen
- Lenken
- Bremsen

## Formale Kriterien:

- Timeout
- Position
  - Erreichbare Flächen
  - Beschränkende Flächen
  - Verbotene Flächen
- Weitere Restriktionen
  - Rotation
  - Geschwindigkeit

1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. **Meilenstein 2**
5. Ausblick und Reflexion

## Testfall Spurhalteassistent





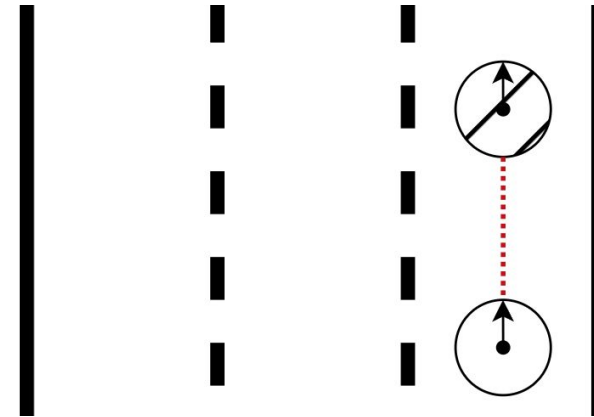
# Adaptiver Tempomat

## Funktionale Anforderungen:

- Gezielt ein- / ausschalten
- Sicherheitsabstand einhalten
- Automatische Geschwindigkeit

## Nicht funktionale Anforderungen:

- Modellvorhersage nutzen
- Erfüllt die StVO-Anforderungen an Sicherheitsabstände



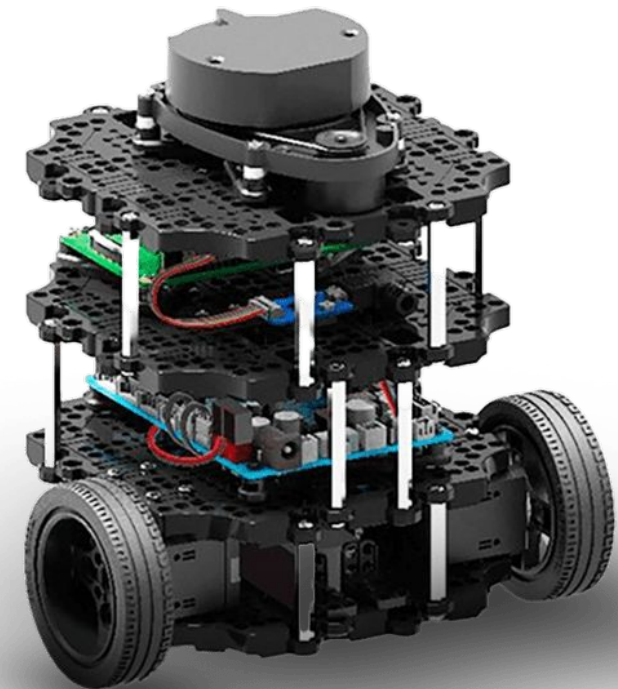
# Adaptiver Tempomat

- Der adaptive Tempomat verwendet eine Modellvorhersage (linearisiertes Bicycle-Modell)
- Der minimale Sicherheitsabstand basiert auf Recherche (StVO §2 & §4)
  - Faustformel 1: Halber Tacho in Meter
    - 50 km/h → 25 m
    - 100 km/h → 50 m
  - Faustformel 2: Innerorts 1 Sekunde, Außerorts 2 Sekunden
    - 50 km/h → 13,9 m
    - 100 km/h → 55,6 m

1. Vision
2. State of the Art
3. Projektmanagement
4. Umsetzung
  - a. Hardware
  - b. Meilenstein 1
  - c. Meilenstein 2
5. Ausblick und Reflexion

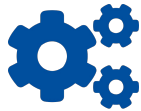
# Ausblick und Reflexion

Carl Schneiders



# Zusammenfassung

## Bisher erreicht:



TurtleCar-Core:  
Entwicklungsplattform für  
Fahrfunktionen



TurtleCar-Test: Testplattform  
zur automatisierten Validierung



Aufbau von Testszenarien



Fahrspurerkennung



Spurhalteassistent

## Nächste Schritte:



Abstandstempomat



Spurwechselassistent



Ausweichen von stationären  
und beweglichen Objekten



Sicheres Verhalten bei  
böartigen Verkehrsteilnehmern



Kolonnenfahrt

## Reflexion

Interne Prozesse im Turtlebot  
schlecht sichtbar

Fehlende Odometrie-Daten

Lange Sprintplanung

Mangelnder übergreifender Fokus

Zeitweise Personalmangel bei  
hoher Arbeitslast

Umfassendes grafisches Debug-Tool

Rekonstruktion aus vorhandenen Daten

Einführung der Refinements

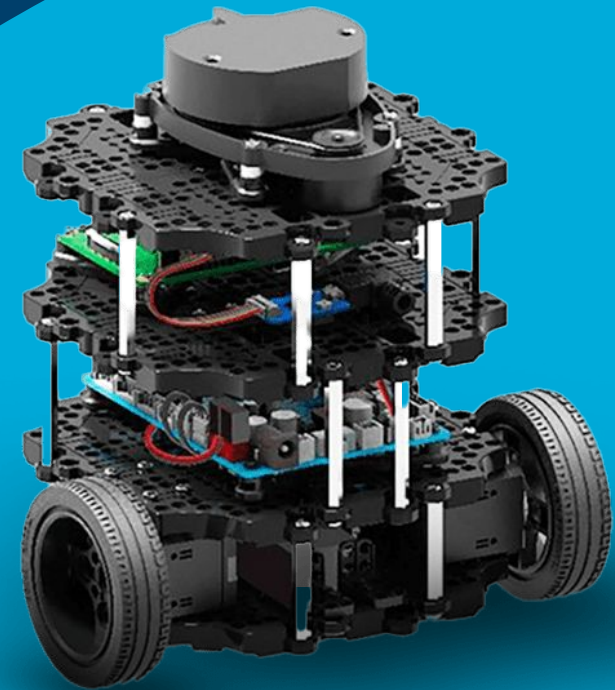
Klare Definition von Meilensteinen und  
Deadlines über Sprintgrenzen hinaus

Urlaub enger mit dem Team abstimmen

Carl von Ossietzky  
Universität  
Oldenburg

# Live-Demo

im PG Raum U104



## Quellen

- [0]: <https://www.elektor.de/robotis-turtlebot3-burger-incl-raspberry-pi-4>
- [1]: <https://www.berrybase.de/raspberry-pi-4-computer-modell-b-2gb-ram>
- [2]: <https://emmanual.robotis.com/docs/en/parts/controller/opencr10/>
- [3]: <https://de.mathworks.com/help/nav/ref/bicyclekinematics.html>
- [4]: <https://auer-gruppe.de/wp-content/uploads/2019/11/Jaguar-F-TYPE-Fahrzeugbild.jpg>
- [5]: <https://www.pngegg.com/en/png-ppmhv>
- [6]: <https://pixabay.com/photos/road-car-vehicle-transportation-1658810/>