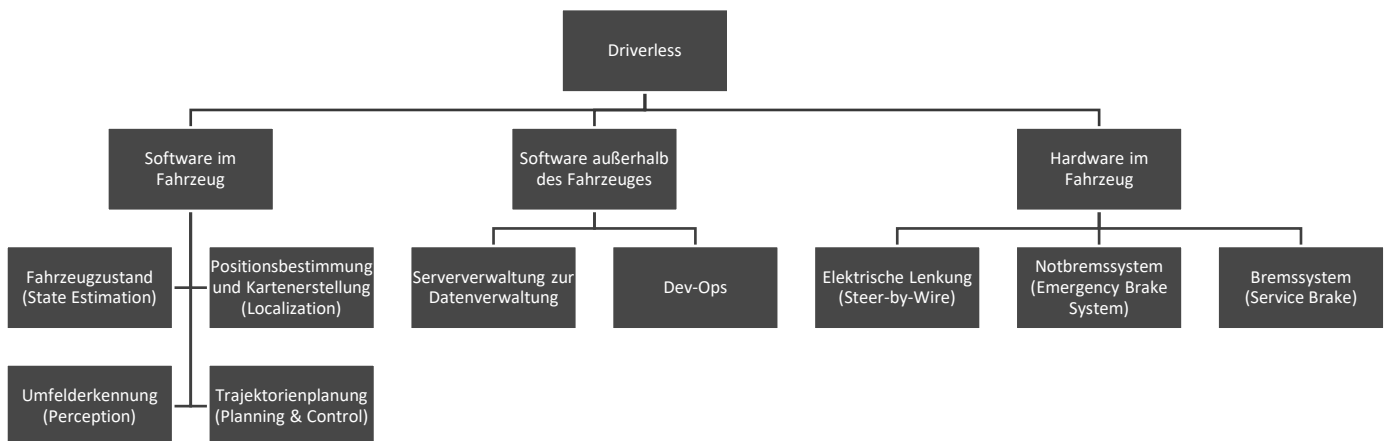


DRIVERLESS

Die Innovationen der Mobilität werden mit dem Modell CASE beschrieben. C für Connectivity, A für Autonomous, S wie Shared und E wie Electric. Elefant Racing e.V. hat bereits 2011 den Umstieg auf den elektrischen Antrieb umgesetzt. Seit 2019 strebt das Team ebenfalls an, das fahrerlose Fahren im eigenen Fahrzeug zu implementieren. Das Subteam Driverless beschäftigt sich dabei mit der Auslegung des fahrerlosen Systems und den Softwarekomponenten. Funktional gliedert es sich wie folgt:



SOFTWAREENTWICKLUNG IM FAHRZEUG

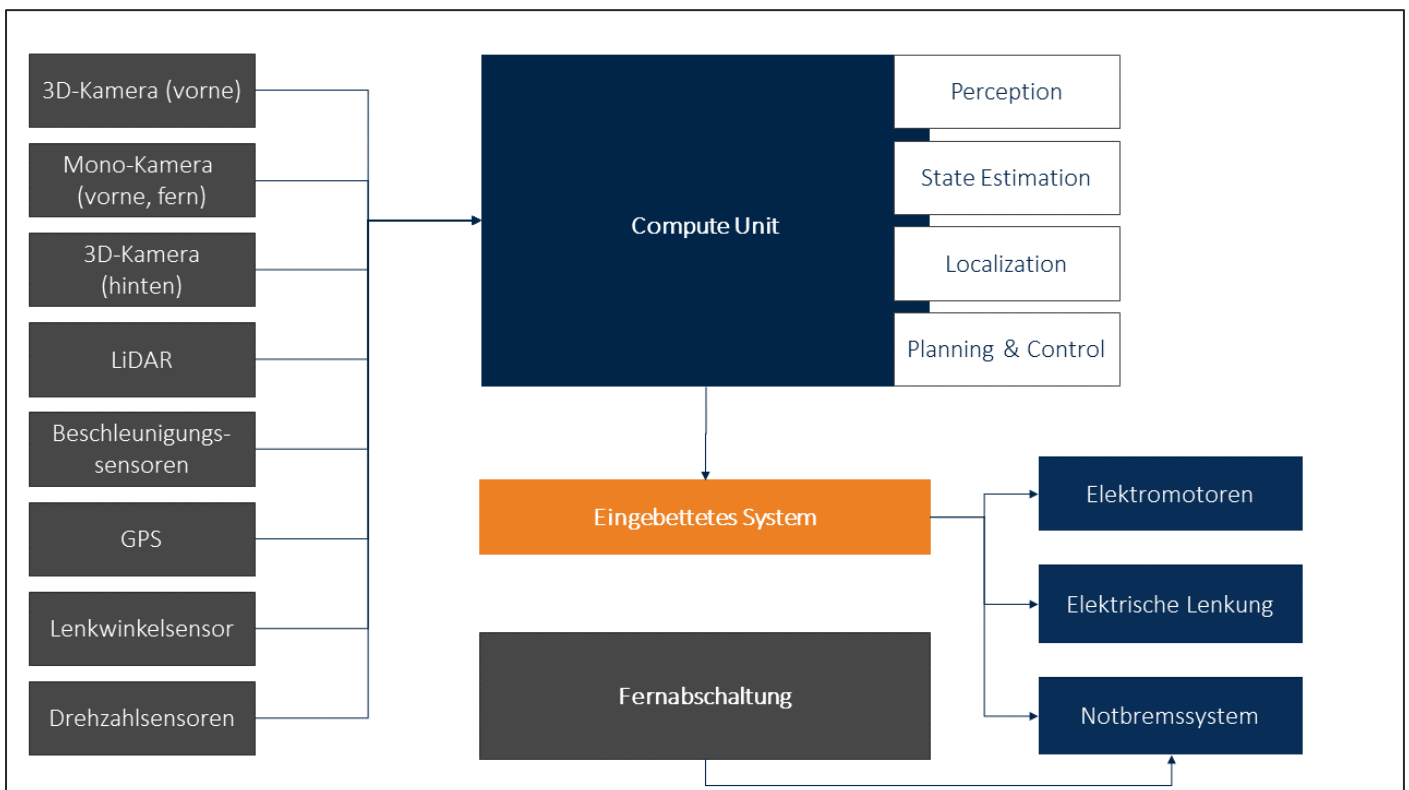
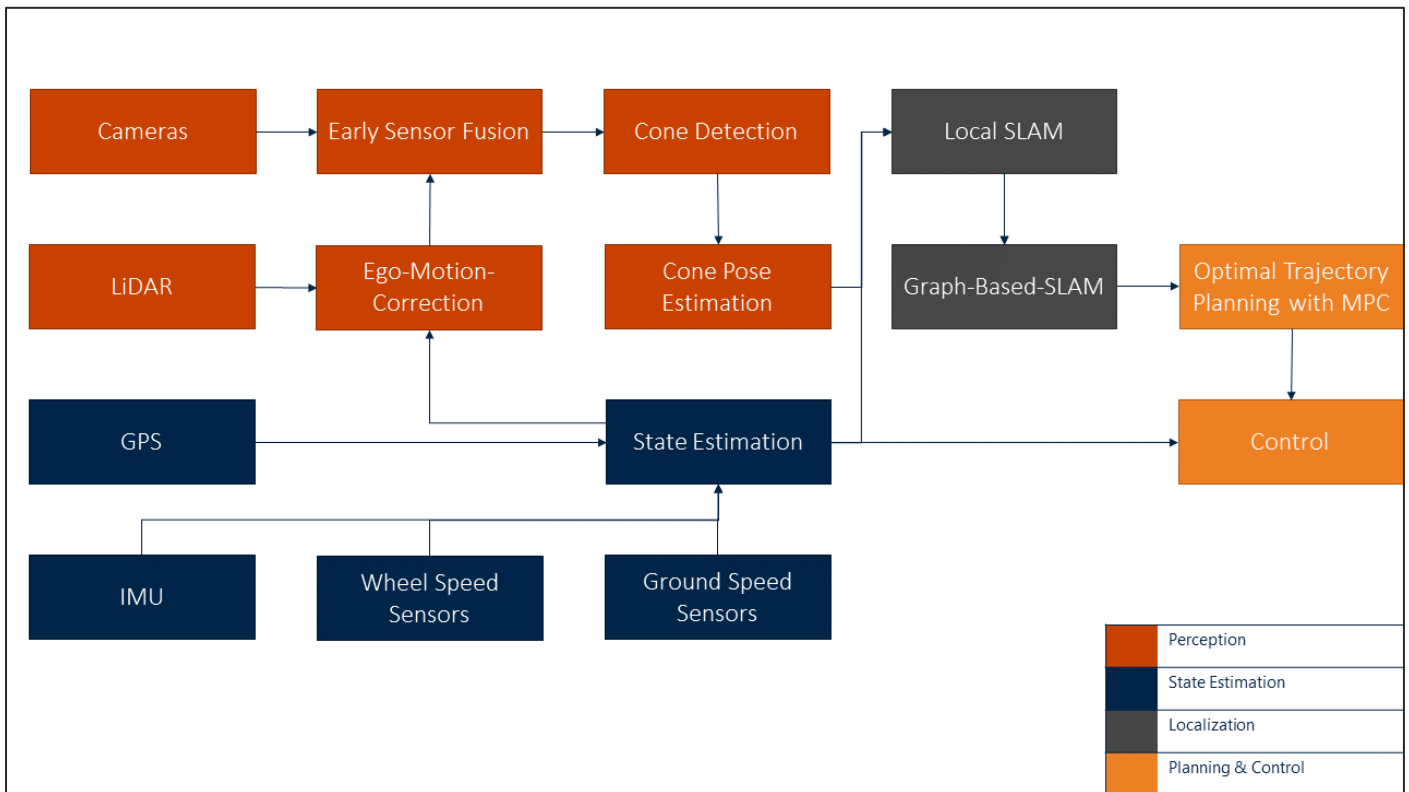
Die Software im Rennwagen basiert auf dem *Robotic Operating System* (ROS) in Linux und besteht aus vier großen Teilbereichen:

Der Bereich **Perception** dreht sich alles um die Umfeldererkennung, d.h. hier dem Erkennen von Leitkegel. Dabei gilt es den Typ und die Position aller Kegel mit (3D)-Kameras und LiDAR zu erkennen. Das Detektieren der Kegel im Kamerabild erfolgt oft durch neuronale Netze. Die Position wird mithilfe der 3D-Daten der Kamera und dem Kameramodell geschätzt. Außerdem wird die Ungenauigkeit der Schätzung bestimmt. Die Position der Kegel kann aus LiDAR-Daten durch klassische Algorithmen (RANSAC, Clustering) bestimmt werden.

Das Ziel des Bereichs **State Estimation** ist den Zustand des Autos zu schätzen. Dieser wird z.B. durch die (lokale) Position und Richtung, Geschwindigkeit, Gierrate und das Schlupfverhältnis der Reifen bestimmt. Diese Größen schätzt man, indem man verschiedenen Messwerte des GPS, der inertielle Messeinheit (inertial measurement unit - IMU) und der Radgeschwindigkeitssensoren zusammenführt. Zu den Werkzeugen in diesem Bereich gehören Kalman-Filter und ein einfaches dynamisches Modell für das Auto. Wie im Bereich **Perception**, gilt es die Sensoren zunächst in das System einzubinden.

Die Softwarekomponente für **Localization** verarbeitet die erkannten Verkehrskegel (Perception) und den Zustand des Autos (State Estimation). Damit wird das Auto auf einer Karte der Strecke lokalisiert. Falls die Strecke unbekannt ist wird gleichzeitig eine Karte mit den Kegeln aufgebaut (simultaneous localization and mapping – SLAM).

Im Bereich **Planning & Control** gilt es eine Trajektorie des Autos zu planen und durch Regelung das Auto auf dieser Trajektorie zu halten. Das Team orientiert sich dabei an einfachen Trajektorien, wie der Mittellinie auf der Strecke. In Zukunft soll eine optimale Bahnplanung und Regelung mittels Modle-Predictive-Control (MPC) durchgeführt werden.



SOFTWARE AUßERHALB DES FAHRZEUGES

Beim autonomen Fahren wird eine Vielzahl von Daten gesammelt. Mit eigenem NAS und Backup im IT-Servicezentrum der Universität Bayreuth (ITS) sind große Mengen an Testdaten mit *S3* und *DVC* ("git für Daten") zu verwalten. In Zukunft ist dafür eine webbasierte GUI zu entwickeln. Für das Training der neuronalen Netzwerke pflegt das Team das Annotierungswerkzeug CVAT.

Wir legen Wert auf einen professionellen, modernen und automatisierten Softwareentwicklungs- und verteilungsprozess vom Laptop über den Server bis in da Auto. DevOps-Interessierte optimieren die Prozesse mit Tools wie Gitlab und Docker.

HARDWARE AM FAHRZEUG

Alle Hardwarekomponenten werden in enger Zusammenarbeit mit den anderen Subteams erarbeitet. Von zentraler Bedeutung sind hierbei die elektrische Lenkung (Steer-by-Wire), das pneumatisch-hydraulische Notbremssystem (Emergency Brake System – EBS) mit Fernabschaltung und das normale Bremssystem (Service Brake) mit den schon vorhandenen Antriebsmotoren. Hierbei programmiert das Team die Bestandteile mit Hilfe von Mikrocontrollern.

TEAM & STRUKTUR

Das Subteam wird aktiv durch zwei leitende Funktionen gesteuert:

- Die **technische Leitung** ist für die Weiterentwicklung des gesamten Systems inkl. Softwarearchitektur verantwortlich.
- Die **administrative Leitung** (Product Manager, Scrum Master) kümmert sich um die interne und externe Organisation des Projektes.

Die Programmierung erfolgt je nach Themengebiet in C++, Python oder Julia. Die Syntax von Julia ist ähnlich zu MATLAB und erlaubt es jedem Mitglied sein Wissen einzubringen und das System interaktiv zu entdecken. Damit können sowohl reine Programmierer als auch Domänenexperten aus den Ingenieurwissenschaften eingebunden werden.

Die Probleme des fahrerlosen Fahrens sind aus der Robotik bekannt und werden seit Jahrzehnten erforscht. Deswegen machen Standardliteratur, Tutorials und Bibliotheken den Einstieg einfach.