Template

Inhaltsverzeichnis

Über arc42

arc42, das Template zur Dokumentation von Software- und Systemarchitekturen.

Erstellt von Dr. Gernot Starke, Dr. Peter Hruschka und Mitwirkenden.

Template Revision: 7.0 DE (asciidoc-based), January 2017

Diese Version des Templates enthält Hilfen und Erläuterungen. Sie dient der Einarbeitung in arc42 sowie dem Verständnis der Konzepte. Für die Dokumentation eigener System verwenden Sie besser die *plain* Version.

1. Einführung und Ziele

Dieser Abschnitt führt in die Aufgabenstellung ein und skizziert die Ziele, die Apollo Auto verfolgt.

1.1. Aufgabenstellung

1.1.1. Was ist Apollo Auto?

Apollo ist eine hochleistungsfähige, flexible Architektur, die die Entwicklung, das Testen und den Einsatz von autonomen Fahrzeugen beschleunigt. Apollo Auto bietet unter Andrem Lösungen für Valet Parking, V2X-Kommunikation und intelligente Lichtsignalanlagen.

1.1.2. Wesentliche Features:

- Valet Parking
  - Software- und Hardware-Integrationslösung. Multifusionslösung bestand aus Fahrzeug, Cloud, HD-Karte und Parkplätzen
  - Bietet hochwertige Dienstleistungen, wie automatische Parkplatzerkennung und autonomes Parken, für Kunden.
- V2X-Kommunikation
  - Interaktionslösung für intelligente Fahrzeuginfrastruktur
  - Apollo V2X umfasst ein intelligentes Transportsystem für Fahrzeug Straßendatenerfassung und intelligente Verarbeitungsanalyse, Verkehrssicherheit und -effizienz
  - Wahrnehmung aller Verkehrsteilnehmer im Sichtfeld und die bereitgestellten straßenseitigen Sensorinformationen können für die Entscheidungsfindung beim autonomen Fahren auf hohem Niveau verwendet werden
  - Wahrnehmung von Verkehrteilnehmern ausserhalb des Sichtfeldes

- Bietet einen vollständigen, kontinuierlichen, multimodalen Datendienst mit niedriger Latenz für L4-Autopilot-Fahrzeuge, die in mehreren Szenarien getestet wurden
- Durch die permanente dynamische Erfassung von Verkehrsinformationen und die Cloud-Integration, wird eine weltweite optimale kollaborative Steuerungsfunktionen für Verkehrsteilnehmer und Verkehrsmanagement erreicht
    - Smart Traffic Signals
- Holographisches Wahrnehmen und Verstehen, basierend auf dem holografischen Wahrnehmungs- und Erkennungssystem
- Status von Fußgängern und Fahrzeugen auf jeder Fahrspur genau erkennen und die Leistung des aktuellen Verkehrsflusses wie Volumen, Warteschlangenlänge, Verspätungen usw.
- Vollständige raum-zeitliche Ableitung und Entscheidungsfindung
- Echtzeitsteuerung der gesamten Szene
- Reduzierung der durchschnittlichen Wartezeit um 20-30% während der Rush Hour
    - Robotaxi
- Die Robotaxis, die aus Chinas erstem werkseitig installierten L4-Passagier-Fahrzeug sind zur Zeit auf öffentlichen Straßen im Testbetrieb
- Sie werden in Kooperation von Baido und FAW an einer gemeinsamen Produktionsline hergstellt
    - Minibus
- Die Minibusse ermöglichen ebenfalls autonomes Fahren der Stufe 4
- Funktionen sind unter Anderem Hinderniserkennung und -vermeidung, zu einem Zielort Fahren und Kreuzungen überqueren

1.2. Qualitätsziele

Die folgende Tabelle beschreibt die zentralen Qualitätsziele von DokChess, wobei die Reihenfolge eine grobe Orientierung bezüglich der Wichtigkeit vorgibt.

Qualitätsziel	Motivation und Erläuterung
Zugängliches Beispiel (Analysierbarkeit)	Apollo Auto ist eine offene Plattform, daher ist es wichtig, dass sich neue Entwickler möglichst schnell in die Architektur, Entwurf und Implementierung einarbeiten können
Echzeitsteuerung von einzelnen Fahrzeugen und Verkehrströmen	Apollo Auto übernimmt zuverlässig und sicher die autonome Steuerung von Fashrzeugen auf Level 4
Echtzeit Umfelderkennung	Für die Steuerung von Fahrzeugen wird ein Modell des Umfelds benötigt. Aus den Sensoprdaten wird ein digitales Abbild des Fahrweges, von beweglichen und unbeweglichen Hindernissen und von Signalen geschaffen
Prediction	Um die Fahrsicherheit weiter zu erhöhen, wird auf die Sensor- und Zustandsdaten von anderen Verkehrsteilnehmern in Echtzeit zugegriffen

1.3. Stakeholder

Die folgende Tabelle stellt die Stakeholder von Apollo Auto und ihre jeweilige Intention dar.

Rolle	Interesse, Bezug
Softwarearchitekten	Wollen ein Gefühl bekommen, wie Architekturdokumentation für ein konkretes System aussehen kann. Möchten sich Dinge (z.B. Form, Notation) für Ihre tägliche Arbeit abgucken. Gewinnen Sicherheit für Ihre eigenen Projekte.
Entwickler	Nehmen Architekturaufgaben im Team wahr. Brauchen ein generelles Verständnis für die Architektur.
OEM & Lieferanten	Entwickeln neue Produkte auf Grundlage von Apollo Auto. Wollen Anregungen für eigene Produkte finden.
Gesetzgeber & Genehmigungsbehörden	Entwickeln einen gesetzlichen Rahmen zur Zulassung von fahrerlosen Fahrzeugen im öffentlichen Straßenverkehr. Etablieren Prüfvorschriften und Tests für Genehmigungsverfahren.
Universitäten	Entwickeln eigene Forschungsprojekte auf Grundlage von Apollo Auto. Wollen Anregungen für weitere Forschungsprojekte und studentische Arbeiten finden.
Studenten	Interessieren sich aufgrund ihres Studiums für die verschiedenen Aspekte einer Architekturdokumentation. Setzen eigene Projekte (z.B. Masterarbeit) zum Thema autonomes Fahren mit Apollo Auto um. Schreiben eine Architekturdokumentation zu Apollo Auto.

2. Randbedingungen

Beim Einsatz von Apollo sind verschiedene Randbedingungen zu beachten. Dieser Abschnitt stellt sie dar und erklärt auch – wo nötig – deren Motivation.

2.1. Technische Randbedingungen

- Für den Einsatz von Apollo Auto wird eine anspruchsvolle Hardwareausstattung benötigt. Eine Umsetzung mit einem marktüblichen Standard-Notebook allein ist nicht möglich.
- Es wird ein Fahrzeug benötigt, dass mit By-Wire-Systemen ausgestattet ist, zum Beispiel Brake-by-Wire, Steering-by-Wire, Throttle-by-Wire oder Shift-by-Wire (Apollo wird derzeit auf Lincoln MKZ getestet).
- Ein Rechner mit einem 4-Kern-Prozessor und mindestens 8 GB Speicher (16 GB für Apollo 3.5 und höher)
- Ubuntu 18.04
- Zusätzlich wird eine umfangreiche Sensorik benötigt die Bild- und Abstandsinformationen aus dem Umfeld aufnehmen
- Arbeitskenntnisse über Docker

2.2. Organisatorische Randbedingungen

Randbedingung	Erläuterungen, Hintergrund
github	*Quellcode ist über github verfügbar.*
Bereitstellung von Daten	*Alle Daten müssen in einem Format hochgeladen werden, das den Apollo-Datenspezifikationen entspricht.*
Speicherung von Daten	*Daten, die in China gesammelt wurden, dürfen nur auf Servern in China gespeichert werden. Daten, die in anderen Ländern und Regionen erhoben werden, unterliegen den Beschränkungen der Datenspeicherung, die durch die Gesetze der jeweiligen Länder festgelegt sind.*
Bereitstellung von Daten	*Als Initiator dieser Plattform stellt Baidu die Ausgangsdaten für diese Plattform bereit. Die Daten stehen allen Partnern dieser Plattform offen. Das Prinzip der fairen Daten stellt sicher, dass Partner mit größeren eigenen Beiträgen mehr Daten und Dienste von dieser Plattform erhalten.*
Datenschutz	*Jeder Partner kann seine eigenen Daten anzeigen und die Datenschutzeigenschaften der Daten als privat oder öffentlich festlegen. Die von Partnern hochgeladenen Daten gelten standardmäßig als privat.*
Endscheidungsführung	*Grundsätzlich handelt es sich bei Apollo Auto um eine offene Plattform. Allerdings wirde Baidu um die architektonische Integrität, die Systemzuverlässigkeit und die schnelle Entwicklung von Apollo zu gewährleisten, bei Bedarf wichtige Entscheidungen treffen, während die aktive Beteiligung der breiteren Gemeinschaft erhalten bleibt.*

2.3. Konventionen

Konvention	Erläuterungen, Hintergrund
Dokumentation	*Terminologie und Gliederung nach dem deutschen arc42-Template in der Version 6.0*
Spezifische Datenformate und Frameworks für autonomes Fahren	*Verwendung etablierter Standards für autonomes Fahren. zum Beispiel sind alle Softwaremodule als ROS(Robot Operating System)-Knoten zu behandeln.*

3. Kontextabgrenzung

Dieser Abschnitt beschreibt das Umfeld von Apollo Auto. Für welche Benutzer ist es da, und mit welchen Fremdsystemen interagiert es?

3.1. Fachlicher Kontext

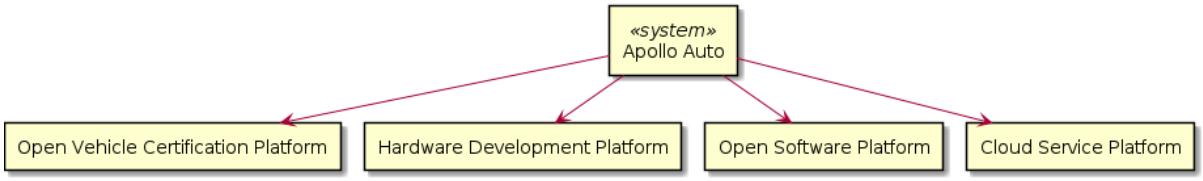


Abbildung 1. Fachlicher Kontext von Apollo Auto

3.1.1. Open Vehicle Certification Platform

Die Open Vehicle Certification Platform schlug eine standardisierte Schnittstelle zwischen dem Autonomen Fahrsystem und dem Fahrzeug vor. Durch diese Plattform können die Fahrzeuganbieter das Fahrzeug einfach mit der offenen Plattform von Apollo verbinden, mehr Entwickler für autonomes Fahren abdecken und die Entwicklung beschleunigen.

3.1.2. Hardware Development Platform

Unter der Hardware Development Platform werden einige durch Apollo Auto unterstütze Hardwarekomponenten zusammengefasst.

Diese sind beispielsweise Sensoren wie Kamera und Lidar sowie Rechen- und Adaptersysteme.

3.1.3. Open Software Platform

Die Open Software Platform fasst alle Softwaremodule zusammen, die von Apollo Auto verwendet werden, um ein autonomes Fahren zu ermöglichen.

Hierzu zählen beispielsweise Module zur Routenplanung, Vorhersage und Wahrneh

3.1.4. Cloud Service Platform

Die Cloud Service Platform fasst ergänzende cloud-basierte Module zusammen die von Apollo Auto verwendet werden.

Hierzu zählen beispielsweise Security, V2X und OTA.

3.2. Technischer Kontext

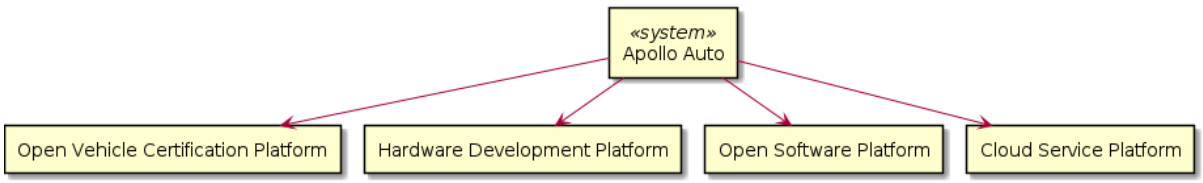


Abbildung 2. Technischer Kontext von Apollo Auto

4. Lösungsstrategie

Dieser Abschnitt enthält einen stark verdichteten Architekturüberblick. Eine Gegenüberstellung der wichtigsten Ziele und Lösungsansätze.

irgendwas zu interoperabilität weil versch. sensoren etc verwendet werden können

5. Bausteinsicht

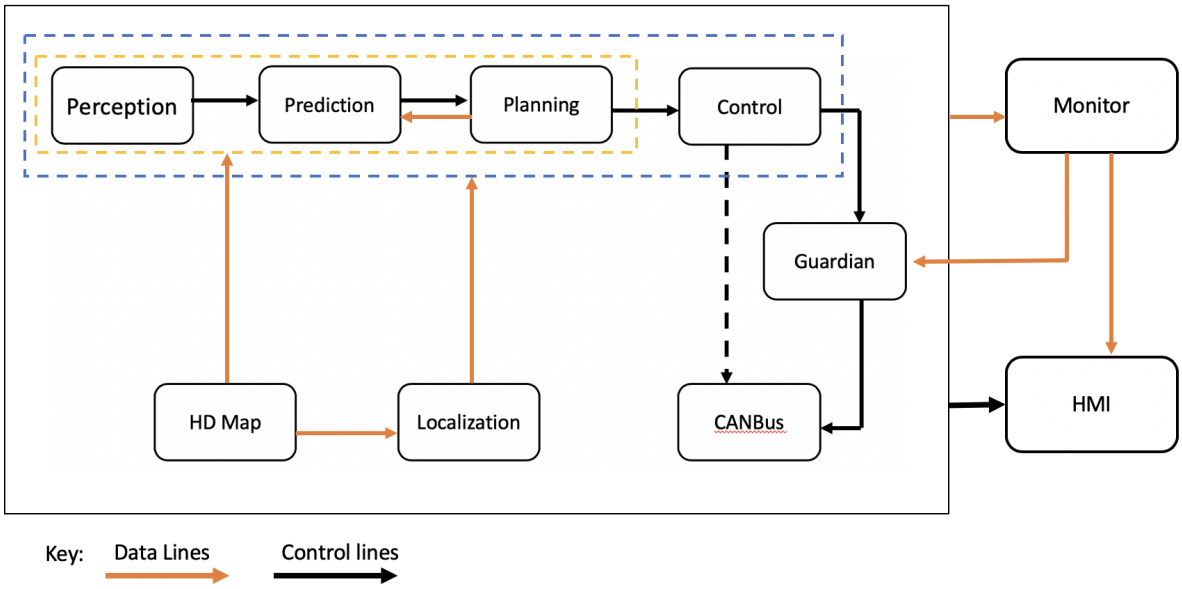
Dieser Abschnitt beschreibt die Zerlegung von Apollo Auto in Module, wie sie sich auch in der Paketstruktur des Java-Quelltextes widerspiegelt. Module der ersten Zerlegungsebene bezeichnen wir in DokChess als Subsysteme. Die → Bausteinsicht, Ebene 1 stellt sie inklusive ihrer Schnittstellen dar.

Für das Subsystem Engine enthält dieser Überblick auch eine detailliertere Zerlegung in → Ebene 2.

5.1. Whitebox Gesamtsystem

DokChess zerfällt wie in Bild unten dargestellt in vier Subsysteme. Die gestrichelten Pfeile stellen fachliche Abhängigkeiten der Subsysteme untereinander dar ("x → y" für "x ist abhängig von y"). Die Kästchen auf der Membran des Systems sind Interaktionspunkte mit Außenstehenden (→ 3.2 Kontextabgrenzung).

Subsystem Kurzbeschreibung XBoard-Protokoll Realisiert die Kommunikation mit einem Client mit Hilfe des XBoard-Protokolls. Spielregeln Beinhaltet die Schachregeln und kann z.B. zu einer Stellung alle gültigen Züge ermitteln. Engine Beinhaltet die Ermittlung eines nächsten Zuges ausgehend von einer Spielsituation. Eröffnung Stellt Züge aus der Eröffnungsliteratur zu einer Spielsituation bereit. Tabelle: Überblick über Subsysteme von DokChess



Begründung

<Erläuternder Text>

Enthaltene Bausteine

Name	Verantwortung
Guardian	Ein Sicherheitsmodul, das die Funktion eines Aktionszentrums übernimmt und eingreift, wenn Monitor einen Fehler erkennt.
Monitor	Das Überwachungssystem aller Module im Fahrzeug inklusive Hardware.
HMI/Dreamview	Human Machine Interface oder DreamView in Apollo ist ein Modul zur Anzeige des Fahrzeugstatus, zum Testen anderer Module und zur Steuerung der Funktion des Fahrzeugs in Echtzeit.
CANBus	Der CanBus ist die Schnittstelle, die Steuerbefehle an die Fahrzeughardware weitergibt. Außerdem gibt er die Fahrwerksinformationen an das Softwaresystem weiter.
Control	Das Control-Modul führt die geplante räumlich-zeitliche Trajektorie aus, indem es Steuerbefehle wie Gas, Bremse und Lenkung erzeugt.
Planning	Das Planning-Modul plant die räumlich-zeitliche Trajektorie, die das autonome Fahrzeug fahren soll.
Routing	Das Routing Modul sagt dem autonomen Fahrzeug, wie es sein Ziel über eine Reihe von Fahrspuren oder Straßen erreichen kann.
Localization	Das Localization-Modul nutzt verschiedene Informationsquellen wie GPS, LiDAR und IMU, um zu schätzen, wo sich das autonome Fahrzeug befindet.

Name	Verantwortung
Prediction	Das Prediction-Modul antizipiert die zukünftigen Bewegungstrajektorien der wahrgenommenen Hindernisse.
Perception	Das Perception-Modul identifiziert die Welt, die das autonome Fahrzeug umgibt. Innerhalb Perception gibt es zwei wichtige Teilmodule: Hinderniserkennung und Ampelerkennung.

Wichtige Schnittstellen

<Beschreibung wichtiger Schnittstellen>

5.2. Guardian

5.2.1. Zweck/Verantwortung

Ein Sicherheitsmodul, dass die Funktion eines "Action Centers" übernimmt und eingreift, wenn das Modul Monitor einen Fehler erkennt. ==== Schnittstelle(n) ==== Qualitäts-/Leistungsmerkmale

- Alle Module funktionieren einwandfrei - Guardian lässt den Steuerfluss normal funktionieren. Steuersignale werden an den CANBus gesendet, als ob Guardian nicht vorhanden wäre.
- Wenn ein Fehler von Monitor erkannt wird, verhindert Guardian, dass Steuersignale den CANBus erreichen und bringt das Auto zum Stillstand. Es gibt 3 Möglichkeiten, wie Guardian entscheidet, wie das Fahrzeug anzuhalten ist. Dazu verwendet Guardian die Ultraschallsensoren
  - Wenn der Ultraschallsensor einwandfrei funktioniert, ohne ein Hindernis zu erkennen, bringt Guardian das Auto langsam zum Stehen
  - Reagieren die Ultraschallsensoren nicht, bremst Guardian hart ab, um das Auto sofort zum Stillstand zu bringen.
  - Ein Sonderfall liegt vor, wenn das HMI den Fahrer über einen drohenden Unfall informiert und der Fahrer 10 Sekunden lang nicht

eingreift, führt Guardian eine Vollbremsung durch, um das Fahrzeug sofort zum Stehen zu bringen.

5.2.2. Ablageort/Datei(en)

https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/guardian

5.3. Monitor

5.3.1. Zweck/Verantwortung

Das Überwachungssystem für alle Module im Fahrzeug einschließlich der Hardware. Monitor empfängt Daten von verschiedenen Modulen und leitet sie an die HMI weiter, damit der Fahrer sie sehen und sicherstellen kann, dass alle Module ohne Probleme funktionieren. Im Falle eines Modul- oder Hardwarefehlers sendet Monitor eine Warnung an Guardian, das dann entscheidet, welche Maßnahmen ergriffen werden müssen, um einen Unfall zu verhindern.

5.3.2. Schnittstelle(n)

5.3.3. Ablageort/Datei(en)

https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/monitor

5.4. HMI/Dreamview

5.4.1. Zweck/Verantwortung

Human Machine Interface oder DreamView in Apollo ist eine Web-Anwendung, die: - die aktuelle Ausgabe relevanter autonomer Fahrmodule visualisiert, z. B. Planung der Trajektorie, Fahrzeuglokalisierung, Fahrwerkstatus usw. - eine Mensch-Maschine-Schnittstelle bietet, über die der Benutzer den Hardwarestatus einsehen, Module ein- und ausschalten und das autonom fahrende Auto starten kann. - bietet Debugging-Tools, wie z. B. PnC-Monitor, zur effizienten Verfolgung von Modulproblemen.

5.4.2. Schnittstelle(n)

5.4.3. Qualitäts-/Leistungsmerkmale

5.4.4. Ablageort/Datei(en)

https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/dreamview

5.5. CANBus

5.5.1. Zweck/Verantwortung

Der CanBus ist die Schnittstelle, die Steuerbefehle an die Fahrzeughardware weitergibt. Außerdem gibt er die Fahrwerksinformationen an das Softwaresystem weiter.

5.5.2. Schnittstelle(n)

- OnControlCommand - ein ereignisbasierter Publisher mit einer Callback-Funktion, die ausgelöst wird, wenn das CANBus-Modul Steuerbefehle empfängt
- OnGuardianCommand - ein ereignisbasierter Publisher mit einer Callback-Funktion, die ausgelöst wird, wenn das Guardian-Modul Steuerbefehle empfängt

5.5.3. Ablageort/Datei(en)

https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/canbus

5.6. Control

5.6.1. Zweck/Verantwortung

Die Control nimmt die geplante Trajektorie als Eingabe und generiert den Steuerbefehl zur Weitergabe an den CANBus.

5.6.2. Schnittstelle(n)

- OnPad
- OnMonitor
- OnChassis
- OnPlanning
- OnLocalization

OnPad und OnMonitor sind Routineinteraktionen mit der PAD-basierten Benutzeroberfläche und Simulationen.

5.6.3. Ablageort/Datei(en)

https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/control

5.7. Planning

5.7.1. Zweck/Verantwortung

Apollo 3.5 verwendet mehrere Informationsquellen, um eine sichere und kollisionsfreie Trajektorie zu planen, daher interagiert das Planning-Modul mit fast jedem anderen Modul.

Zunächst nimmt das Planning-Modul die Vorhersageausgabe. Da die Vorhersageausgabe das ursprünglich wahrgenommene Hindernis umschließt, abonniert das Planning-Modul die Ausgabe der Ampelerkennung und nicht die Ausgabe der wahrgenommenen Hindernisse.

Dann nimmt das Planning-Modul die Routing-Ausgabe. In bestimmten Szenarien kann das Planning-Modul auch eine neue Routing-Berechnung auslösen, indem es eine Routing-Anforderung sendet, wenn der aktuellen Route nicht treu gefolgt werden kann.

Schließlich muss das Planning-Modul den Standort (Lokalisierung: wo bin ich) sowie die aktuellen autonomen Fahrzeuginformationen (Fahrwerk: wie ist mein Status) kennen.

5.7.2. Schnittstelle(n)

5.7.3. Qualitäts-/Leistungsmerkmale

5.7.4. Ablageort/Datei(en)

https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/planning

5.8. Routing

5.8.1. Zweck/Verantwortung

Das Routing-Modul muss den Start- und Endpunkt des Routings kennen, um die Durchfahrtsspuren und Straßen zu berechnen. Normalerweise ist der Startpunkt der Standort des autonomen Fahrzeugs. Die RoutingResponse wird wie unten gezeigt berechnet und veröffentlicht. ==== Schnittstelle(n) ==== Qualitäts-/Leistungsmerkmale ==== Ablageort/Datei(en) https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/routing

5.9. Localization

5.9.1. Zweck/Verantwortung

Das Lokalisierungsmodul aggregiert verschiedene Daten, um das autonome Fahrzeug zu lokalisieren. Es gibt zwei Arten von Lokalisierungsmodi: OnTimer und Multiple SensorFusion.

Die erste Lokalisierungsmethode ist RTK-basiert, mit einer Timer-basierten Callback-Funktion OnTimer, wie unten gezeigt.

Die andere Lokalisierungsmethode ist die Multiple Sensor Fusion (MSF)-Methode, bei der eine Reihe von ereignisgesteuerten Callback-Funktionen registriert werden, wie unten gezeigt. ==== Schnittstelle(n) ==== Qualitäts-/Leistungsmerkmale ==== Ablageort/Datei(en) https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/localization

5.10. Prediction

5.10.1. Zweck/Verantwortung

Das Prediction-Modul schätzt die zukünftigen Bewegungstrajektorien für alle wahrgenommenen Hindernisse. Die ausgegebene Vorhersagemeldung beinhaltet die Informationen zur Hinderniserkennung. Prediction abonniert Lokalisierungs-, Planungs- und Wahrnehmungs-Hindernis-Nachrichten wie unten dargestellt. Wenn ein Lokalisierungsupdate empfangen wird, aktualisiert das Prediction-Modul seinen internen Status. Die eigentliche Vorhersage wird ausgelöst, wenn Perception ihre Perception-Hindernismeldung aussendet.

5.10.2. Schnittstelle(n)

5.10.3. Ablageort/Datei(en)

https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/prediction

5.11. Perception

5.11.1. Zweck/Verantwortung

Das Perception-Modul verfügt über die Fähigkeit, 5 Kameras (2 vorne, 2 seitlich und 1 hinten) und 2 Radare (vorne und hinten) zusammen mit 3 16-Linien-LiDARs (2 hinten und 1 vorne) und 1 128-Linien-LiDAR zu verwenden, um Hindernisse zu erkennen und ihre individuellen Spuren zu einer endgültigen Spurliste zu verschmelzen. Das Hindernis-Submodul erkennt, klassifiziert und verfolgt Hindernisse. Dieses Teilmodul sagt auch die Bewegung und Positionsinformationen des Hindernisses voraus (z. B. Richtung und Geschwindigkeit). Für die Fahrspur werden Fahrspurinstanzen durch Nachbearbeitung von Fahrspur-Parsing-Pixeln konstruiert und die relative Position der Fahrspur zum Ego-Fahrzeug berechnet (L0, L1, R0, R1, usw.). ==== Schnittstelle(n) ==== Ablageort/Datei(en) https://github.com/ApolloAuto/apollo/tree/r5.5.0/modules/perception

5.11.2. <Name Schnittstelle 1>

...

5.11.3. <Name Schnittstelle m>

5.12. Ebene 2

5.12.1. Whitebox *<Baustein 1>*

<Whitebox-Template>

5.12.2. Whitebox *<Baustein 2>*

<Whitebox-Template>

...

5.12.3. Whitebox *<Baustein m>*

<Whitebox-Template>

5.13. Ebene 3

5.13.1. Whitebox <_Baustein x.1_>

<Whitebox-Template>

5.13.2. Whitebox <_Baustein x.2_>

<Whitebox-Template>

5.13.3. Whitebox <_Baustein y.1_>

<Whitebox-Template>

Hardware Development Platform

STUFF

- ASU
  - Apollo Sensor Unit (ASU) is designed to work with Industrial PC (IPC) to implement sensor fusion, vehicle control and network access in Apollo's autonomous driving platform.
  - The ASU system provides sensor interfaces to collect data from various sensors, including cameras, Radars, and Ultrasonic Sensors. The system also utilizes pulse per second (PPS) and GPRMC signals from GNSS receiver to enable synchronization for the camera and LiDAR sensors.
  - The communication between the ASU and the IPC is through PCI Express Interface. ASU collects sensor data and passes to IPC via PCI Express Interface, and the IPC uses the ASU to send out Vehicle Control commands in the Controller Area Network (CAN) protocol.

AXU Apollo Extension Unit (AXU) is designed to boost computation capability and expand storage capacity by enabling developers to plug-in additional accelerators including GPU, FPGA modules, and etc.

ACU Apollo Computing Unit

- Introduction
  - Integrated Autosar software
  - ASIL-D functional safety level with special hardware safety island design
  - 100% Auto-grade components
  - IATF16949 design with PPAP supply chain and production management
- Features:
  - Power supply8~16V
  - Max Power Consumption28WStatic Power Consumption<0.1mA
  - Computing PowerUp to 1.5TOPS

- SOC/MCU：Xilinx ZU5/ Aurix TC297
- Operating temperature：-40~85℃
- OS：Linux/QNX & AUTOSAR
- Size：
- 200 x 170 x 36mm（Working temperatures 85C，or
- 200 x 120 x 36mm（Working temperatures 70C)
- Cooling: Natural Cooling
- Interface：
- 5：GMSL Video Input - support 1.3 megapixel and 2 megapixel
- 1：GMSL Video Output
- 4：CAN（support CAN-FD）
- 12：Ultrasonic Rdar Interface
- 1：100BASE-T1
- 3：Analog Switch

CAN-PCIe/402-B4

Nuvo-6108GC Vendor：NeousysApollo Platform Supported Introduction：Nuvo-6018GC is world's first industrial-grade GPU computer supporting high-end graphics cards. It's designed to fuel emerging GPU-accelerated applications, such as artificial intelligence, VR, autonomous driving and CUDA computing, by accommodating NVIDIA® GPU with up to 250W TDP. Link

ProPak6™ Vendor：NovAtelApollo Platform Supported Introduction：ProPak6™ is an enclosure product manufactured by NovAtel. From standalone metre-level to RTK centimetre-level positioning, the ProPak6 is flexible to meet your positioning needs. Reliability is safeguarded as a result of the extremely rugged and water resistant IP67 housing combined with its wide operating temperature range. NovAtel has also assured faster time to market by reducing integration time with standardized software and hardware connections. The ProPak6 offers optional GPRS/HSPA cellular modem and/or heading options to provide a solution for many applications. Link PwrPak 7D

Vendor：NovAtelApollo Platform Supported Introduction：The PwrPak7D is a robust, high precision receiver ideal for ground vehicle, marine or aircraft based systems. Its multi-frequency dual antenna input allows the PwrPak7D to utilize NovAtel CORRECT® with RTK and ALIGN® functionality. The PwrPak7D has a powerful OEM7® Global Navigation Satellite System (GNSS) inside and offers built-in Wi-Fi, on board NTRIP client and server support and 16 GB of internal storage. Link NV-GI120 Vendor：NavTech Inc.Apollo Hardware Development Platform Supported Introduction：NV-GI120 is a position and orientation system for automatic drive of NAV Technology. With the high-precision GNSS board card and high-precision MEMS gyro, it has the real-time attitude and position resolving ability while transmitting the original data of the sensor and board card for post-processing high-precision resolution. Newton-M1 Vendor：Starneto Introduction：Newton series MEMS inertial/satellite integrated navigation products not only have compact structure , rich interface resources, but also highly cost-effective. Moreover, they can realize high frequency and precision position, speed detection and attitude determination for various vehicles. Link

MARS Vendor：ON SemiconductorApollo Hardware Development Platform supported Introduction：The Modular Automotive Reference System (MARS) is a complete imaging solution for camera system developers and software developers working on automotive imaging applications. MARS gives engineers and software developers the fundamental building blocks needed to create next generation imaging systems, while reducing the design effort and resources required to develop a working solution. Link Vendor：Wissen TechnologiesApollo Hardware Development Platform Supported

- Introduction：
  - 30mm x 30mm coax camera module
  - 1080p FHD YUV422 data
  - HDR function(High Dynamic Range) higher than 100dB
  - support external trigger function

Link LI-USB30-AR023ZWDR Vendor：Leopard Imaging Inc.Apollo Platform

Supported

- Introduction
  - Key Features
  - USB 3.0 Super Speed support
  - Support register access function
  - ON Semiconductor AR023Z 1080p HD Sensor
  - Support CS lens
  - Pixel Size: 3.0um x 3.0um
  - Provide customization services
  - YUV output without compression
  - USB +5VDC powered device
  - UVC compliance
  - Built in AP0202 ISP
  - Support External Trigger, Software Trigger
  - Compact Size: 30mm x 30mm

ARS408-21 VendorContinentalApollo Platform Supported IntroductionThe ARS408 realized a broad field of view by two independent scans in conjunction with the high range functions like Adaptive Cruise Control, Forward Collision Warning and Emergency Brake Assist can be easily implemented. Its capability to detect stationary objects without the help of a camera system emphasizes its performance. The ARS408 is a best in class radar, especially for the stationary target detection and separation. Link B01HC VendorRacobitApollo Hardware Development Platform Supported IntroductionThe 77GHz millimeter-wave automotive anti-collision radar developed by RACO (Beijing Racobit Electronic Information Technology Co., Ltd) utilizing MIMO virtual aperture technology achieves higher precision, finer angle resolution and smaller volume, which is compatible with long-and-mid-range detection function. It enables real-time detection of the vehicle’s driving environment as well as other vehicle targets in various working environments, which is the core sensor of the driverless

technology and ADAS system.

VLS-128 Vendor：VelodyneApollo Platform Supported

- Introduction：
  - 360° Horizontal FOV
  - +15° to -25° Vertical FOV
  - Up to 300m Range
  - Minimum Angular Resolution: 0.11°
  - Up to 4 Return Modes
  - Up to ~9.6 Million Points per Second
  - Environmental Protection: IP67
  - Connectors: RJ45 / M12
  - High Volume, Automotive Grade Contract Pricing Link Scala 2 Vendor：ValeoApollo Hardware Development Platform supported Introduction：Valeo provides its laser scanner to Apollo. The Valeo SCALA® is the first 3D laser scanner compliant with the fierce requirements for automotive mass production. SCALA® offers an unique combination of wide field of view, large detection range and high precision, capable of detecting both stationary and moving objects during day and night. Link M16-LSR Vendor：LeddarTechApollo Hardware Development Platform supported Introduction：The Leddar M16 Sensor Modules are advanced, solid-state LiDAR solutions that combine wide-beam flash illumination with 16 independent detection segments to simultaneously deliver rapid, continuous and precise detection and ranging for multiple objects along with excellent lateral discrimination. Based on the patented Leddar Technology, LeddarTech's off-the-shelf solid-state LiDAR modules for mobility applications are ready for integration into specific projects for R&D, proof-of-concept, field validation, and platform seeding. Link LEDDARVU (VU8) Vendor：LeddarTechApollo Hardware Development Platform supported Introduction：Leddar Vu8 is an affordable, versatile solid-state LiDAR

sensor module that delivers exceptional detection and ranging performance in a small, robust package. LeddarVu modules provide the ability to detect and track multiple objects simultaneously over eight distinct segments with superior lateral discrimination capabilities. Based on the patented Leddar Technology, LeddarTech's off-the-shelf solid-state LiDAR modules for mobility applications are ready for integration into specific projects for R&D, proof-of-concept, field validation, and platform seeding. Link HDL-64E S3 Vendor□VelodyneApollo Platform Supported Introduction□The HDL-64E LiDAR sensor is designed for obstacle detection and navigation of autonomous ground vehicles and marine vessels. Its durability, 360° field of view and very high data rate makes this sensor ideal for the most demanding perception applications as well as 3D mobile data collection and mapping applications. The HDL-64E's innovative laser array enables navigation and mapping systems to observe more of their environment than any other LiDAR sensor. Link ULTRA Puck VLP-32C Vendor□VelodyneApollo Hardware Development Platform supported Introduction□Velodyne LiDAR's ULTRA Puck™ VLP-32C is the newest long-range LiDAR sensor in its product portfolio that combines best-in-class performance with a small form factor. A high-resolution LiDAR sensor that is cost- effective when compared to similar performance sensors but developed with automotive applications in mind to ensure reliability while delivering the performance demanded by the market. The VLP-32C retains the innovative breakthroughs in 3D LiDAR such as 360° surround view along with real-time 3D data that includes distance and calibrated reflectivity measurements along with rotational angles. Link PUCK VLP-16, PUCK Hi-Res, PUCK LITE Vendor□VelodyneApollo Hardware Development Platform Supported Introduction□Velodyne's new PUCK™ (VLP-16) sensor is the smallest, newest, and most advanced product in Velodyne's 3D LiDAR product range. Vastly more cost-effective than similarly priced sensors, and developed with mass production in mind, it retains the key features of Velodyne's breakthroughs in LiDAR: Real-time, 360°, 3D distance and calibrated reflectivity measurements. Link Pandora Vendor□HesaiApollo Platform

Supported Introduction：Pandora is an all-in-one sensor kit for environmental sensing for self-driving cars. It integrates cameras, LiDAR and data processing ability (from Baidu Apollo) into the same module, with advanced synchronization and calibration solutions. Link Vendor：InnovusionApollo Hardware Development Platform Supported

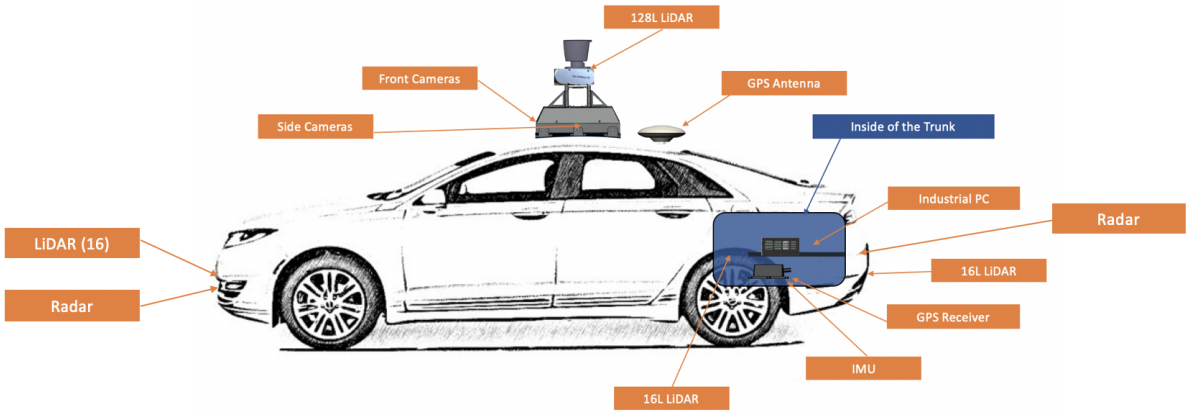
- Introduction：
  - Resolution: provides near picture quality with over 300 lines of resolution and several hundred pixels in both the vertical and horizontal dimensions.
  - Range: detects both light and dark objects at distances up to 150 meters away which allows cars to react and make decisions at freeway speeds and during complex driving situations.
  - Sensor fusion: fuses LiDAR raw data with camera video in the hardware layer which dramatically reduces latency, increases computing efficiency and creates a superior sensor experience.
  - Accessibility: enables a compact design which allows for easy and flexible integration without impairing vehicle aerodynamics. Innovusion's products leverage components available from mature supply chain partners, enabling fast time-to-market, affordable pricing and mass production. C16 Series Vendor：LeiShen Intelligent SystemApollo Hardware Development Platform Supported Introduction：LeiShen's developing 3D Multi-channel LiDARs including 2/4/8/16/32-channel have excellent cost performance ratio and wide range of applications. Link Rs-LiDAR-16 Vendor：RobosenseApollo Hardware Development Platform Supported Introduction：The compact housing of RS-LiDAR-16 mounted with 16 laser/detector pairs rapidly spins and sends out high-frequency laser beams to continuously scan the Surrounding environment. Advanced digital signal processing and ranging algorithms calculate point cloud data and reflectivity of objects to enable machine to 'see' the world and providing reliable data for localization, navigation and obstacle avoidance.

6. Laufzeitsicht

TBD

7. Verteilungssicht

7.1. Infrastruktur Ebene 1



Begründung

<Erläuternder Text>

Qualitäts- und/oder Leistungsmerkmale

<Erläuternder Text>

Zuordnung von Bausteinen zu Infrastruktur

<Beschreibung der Zuordnung>

7.2. Infrastruktur Ebene 2

7.2.1. *<Infrastrukturelement 1>*

<Diagramm + Erläuterungen>

7.2.2. *<Infrastrukturelement 2>*

<Diagramm + Erläuterungen>

...

7.2.3. *<Infrastrukturelement n>*

<Diagramm + Erläuterungen>

8. Querschnittliche Konzepte

Dieser Abschnitt beschreibt allgemeine Strukturen und Aspekte, die systemweit gelten. Darüber hinaus stellt er verschiedene technische Lösungskonzepte vor.

8.1. Unter-der-Haube

Die Dynamik von autonomen Fahrzeugen (AV) wird von der Planungs-Engine über den Controller Area Network-Bus (CAN-Bus) gesteuert. Die Software liest Daten aus Hardware-Registern und schreibt sie zurück, genau wie wir es in Assembler tun würden. Für hochpräzise Berechnungen arbeiten die Lokalisierungs-, Wahrnehmungs- und Planungsmodule als unabhängige Eingabequellen, während die Ausgabequellen über das Peer2Peer (P2P)-Protokoll zusammenarbeiten. P2P wird durch die RPC-Netzwerkanwendung unterstützt.

Apollo Auto verwendet ROS1 als zugrunde liegendes Netzwerk, was bedeutet, dass Apollo Auto das Master-Nodes-Framework von ROS1 entlehnt. Da xmlRPC von ROS1 wirklich alt ist (im Vergleich zu den neueren brpc und grpc), hat Baidu seine eigene protobuf-Version von RPC entwickelt.

8.2. *<Konzept 2>*

<Erklärung>

...

8.3. *<Konzept n>*

<Erklärung>

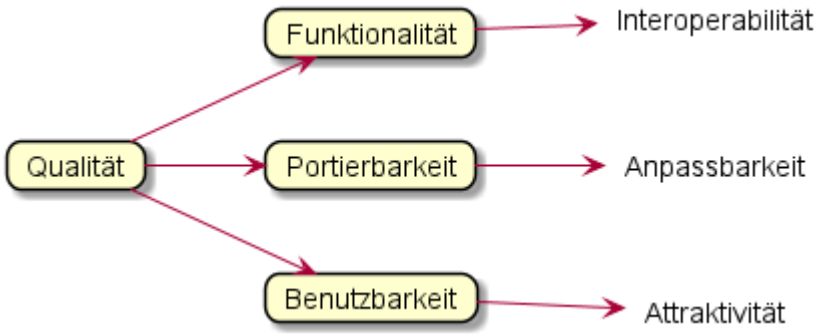
9. Entwurfsentscheidungen

10. Qualitätsanforderungen

Dieser Abschnitt beinhaltet konkrete Qualitätsszenarien, welche die zentralen Qualitätsziele, aber auch andere geforderte Qualitätseigenschaften besser fassen. Sie ermöglichen es, Entscheidungsoptionen zu bewerten.

10.1. Qualitätsbaum

Das folgende Bild gibt einen Überblick über die relevanten Qualitätsmerkmale und den ihnen jeweils zugeordneten Szenarien.



10.2. Qualitätsszenarien

11. Risiken und technische Schulden

Die folgenden Risiken wurden zu Beginn des Vorhabens identifiziert. Sie beeinflussten die Planung der ersten drei Iterationen maßgeblich. Seit Abschluss der dritten Iteration werden sie beherrscht. Dieser Architekturüberblick zeigt die Risiken inklusive der damaligen Eventualfallplanung weiterhin, wegen ihres großen Einflusses auf die Lösung.

12. Glossar

Das folgende Glossar erklärt Begriffe aus dem Bereich Autonomes Fahren.

Begriff	Definition
By-Wire	*Bezeichnung für (zumindest partielles) Fahren oder Steuern von Fahrzeugen ohne mechanische Kraftübertragung der Bedienelemente zu den entsprechenden Stellelementen wie etwa Drosselklappen. Das By-Wire-Konzept umfasst dabei zumindest zwei oder mehr der „X-by-Wire“- Systeme wie etwa Brake-by-Wire (Bremssteuerung) und Steer-by-Wire (Lenkung)..*
LIDAR	*ist eine dem Radar verwandte Methode zur optischen Abstands- und Geschwindigkeitsmessung sowie zur Fernmessung. Statt der Radiowellen wie beim Radar werden Laserstrahlen verwendet.*
RADAR	*ist die Bezeichnung für verschiedene Erkennungs- und Ortungsverfahren und -geräte auf der Basis elektromagnetischer Wellen im Radiofrequenzbereich (Funkwellen).*
P2P	*Peer-to-Peer und Rechner-Rechner-Verbindung sind synonyme Bezeichnungen für eine Kommunikation unter Gleichen, hier bezogen auf ein Rechnernetz.*
RPC	*Remote Procedure Call ist eine Technik zur Realisierung von Interprozesskommunikation. Sie ermöglicht den Aufruf von Funktionen in anderen Adressräumen. Im Normalfall werden die aufgerufenen Funktionen auf einem anderen Computer als das aufrufende Programm ausgeführt.*

13. Anhang

13.1. Apollo

13.2. Arc42

- die Dokumentation muss bereits zu Beginn der Arbeiten an der Architektur/am Quellcode erfolgen um stets den aktuellen Stand abbilden zu können.
- die Dokumentation soll mit dem Projekt wachsen!
- die nachträgliche Dokumentation ist eine Sysiphusarbeit
    - Beispiel:
        - Diagramme in der Laufzeitsicht sind effizienter zu gestalten während ein Modul im Entstehen ist. Änderungen können mit jeder Entwicklungsstufe eingepflegt werden.
        - Ein fertiges Modul in ein Diagramm zu überführen erfordert viel Zeit, da alle beteiligten Module und deren Funktionsweise erst identifiziert werden müssen.

13.3. docToolChain

- plantUML ist zwar bei sehr einfachen Diagrammen eine Erleichterung, kommt aber an beispielsweise draw.io oder tikz (in Latex) nicht heran wenn es um komplexe und detailierte Diagramme geht
    - teilweise werden bei der inline-Erstellung bei plantUML Diagrammeinstellungen nicht übernommen
- Schriftgrößen sind nicht einstellbar, was an manchen Stellen - z.B Tabellen nötig gewesen wäre
- Ansonsten ist docToolChain eine interessante Alternative, besonders in Bezug auf den CI-Task in github.
    - damit nicht bei jedem Commit die Dokumentation neu erstellt wird,

bietet github die Möglichkeit über Commitnachrichten, z.B "[no ci]" oder "[skip ci]" den CI-Task auszusetzen. Hier wäre es sinnvoller über Commitnachrichten explizit die Generierung zu starten.