





# Projektgruppe CHILL

## Abschlussreview

# Agenda



Einleitung & Projektorganisation	Patrick	20 min
Anforderungsanalyse	Jens	15 min
Konzeptionierung	Raphael	20 min
Safetyanalyse	Dominik	15 min
Implementierung	Raphael, Jens, Wiebke, Lynn	60 min
Testmanagement	Johann	15 min
Live-Demonstration	Dominik	20 min
Abschluss	Patrick	10 min

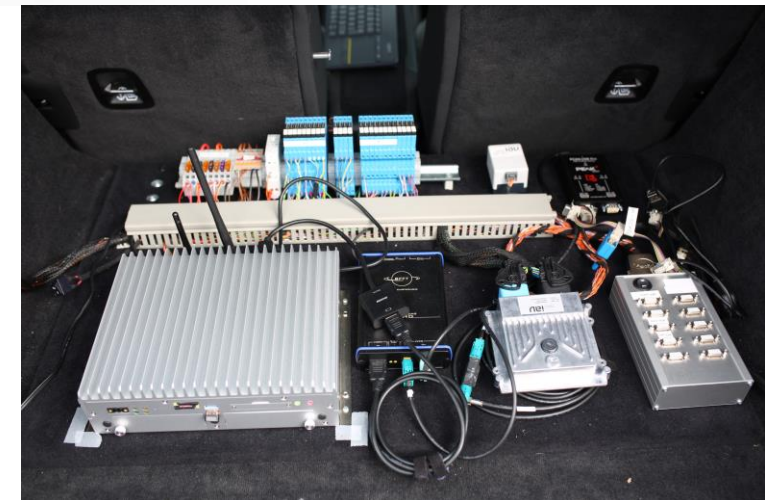
# Einleitung

- Vorstellung der Projektgruppe
- Vision der Projektgruppe
- Projektorganisation
  - Prozessmodel
  - Risikoanalyse

- Cooling & Heating, Independent Learning Limousine
- Abteilungen
  - „Hybride Systeme“ von Prof. Fränzle
  - „Sicherheitskritische Eingebettete Systeme“ von Prof. Damm
- Kooperation mit OFFIS e. V. und IAV GmbH
  - 3 Betreuer Universität/OFFIS
  - externe Ansprechpartner der IAV
- 10 Teilnehmer



- Versuchsträger Porsche Panamera 4 E-Hybrid Sports Turismo
- Ansteuerbar durch die Projektgruppe:
  - 108 Sensoren
    - Regen-Licht-Sensor
    - Ambientebeleuchtung
    - ...
  - 27 Aktoren
    - Fensterheber vorne links
    - Sitzheizung vorne rechts
    - ...



- Vorklimatisierung
- Einstellen von Abfahrtszeiten

(z.B. Volvo/Porsche)

- Steuerung der Vorklimatisierung (wie/wann) per mobiler Applikation

(z.B. Chevrolet/Porsche)

Vorkonditionierung

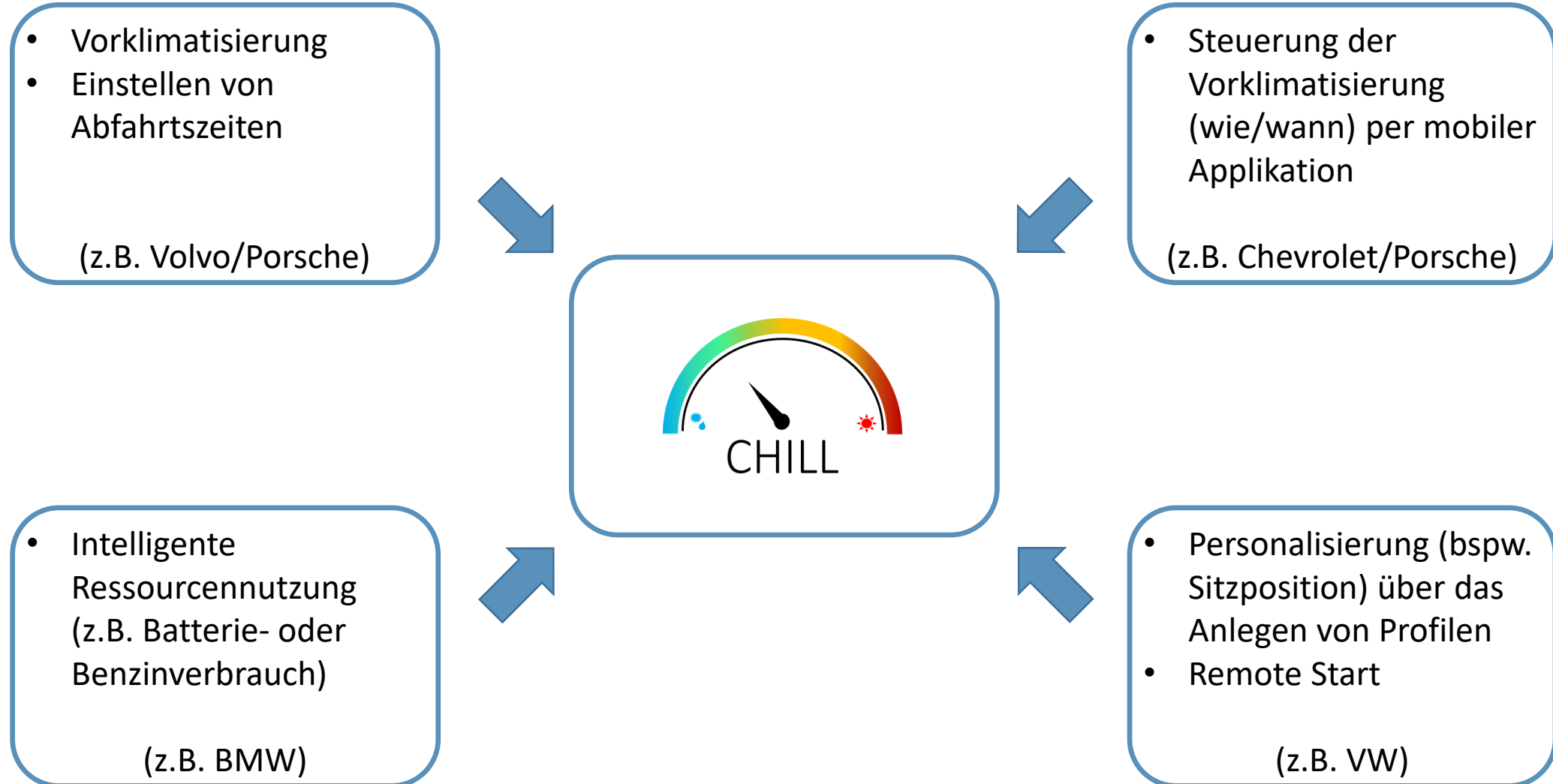
- Intelligente Ressourcennutzung (z.B. Batterie- oder Benzinverbrauch)

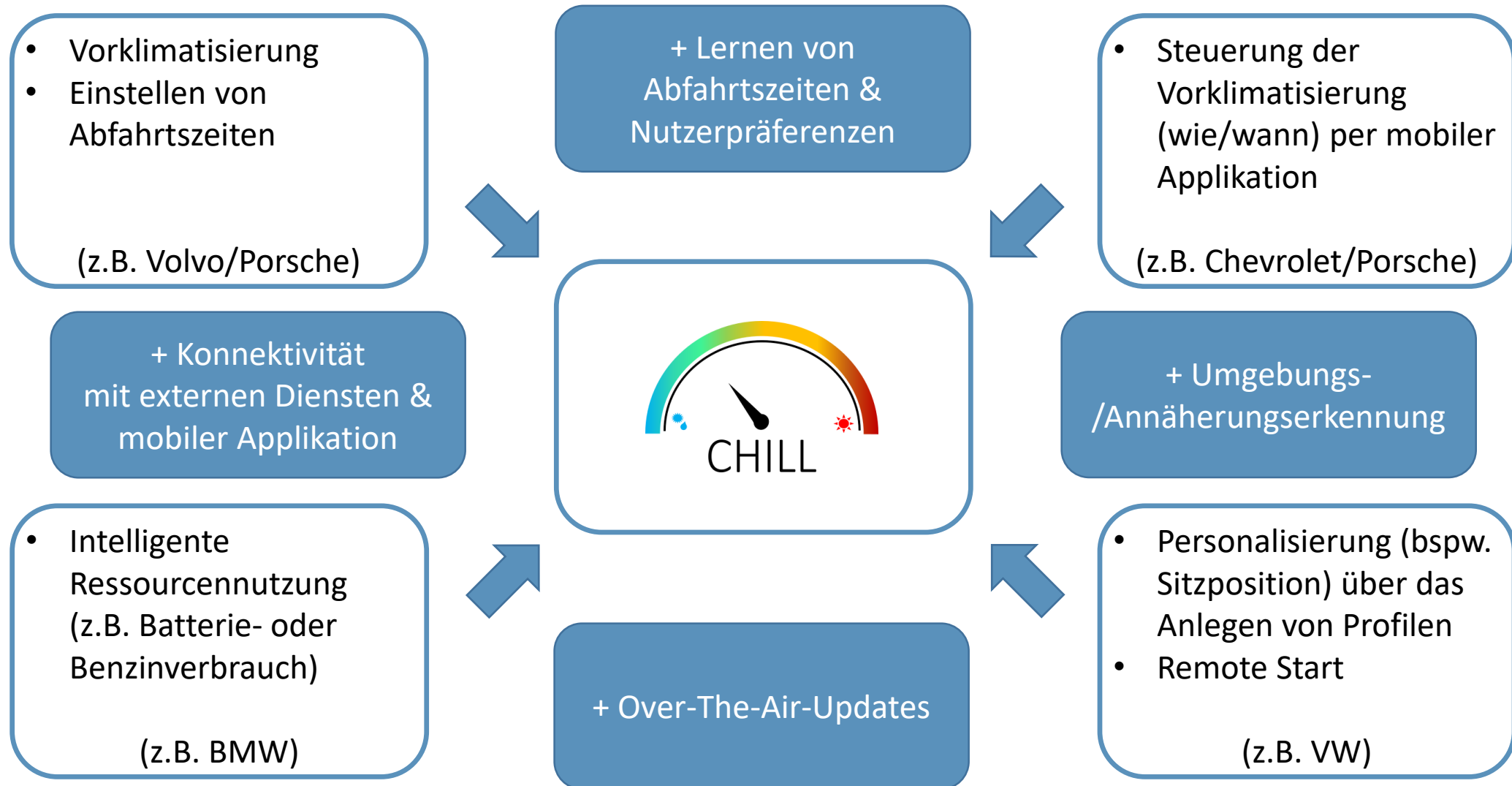
(z.B. BMW)

- Personalisierung (bspw. Sitzposition) über das Anlegen von Profilen
- Remote Start

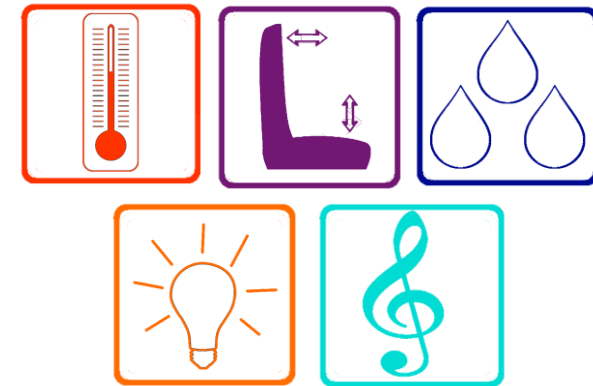
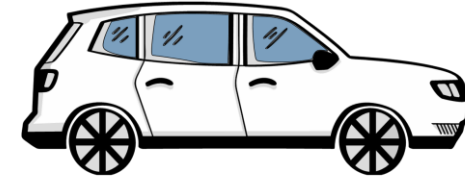
(z.B. VW)







- Automatische Vorkonditionierung
  - Anpassen der Atmosphäre
  - Selbstständig
- Lernen
  - Erfahrungen nutzen
  - Nutzerorientiert
- Konnektivität
  - Aktuelle Wetterdaten
  - Verbindung zu Smart Device



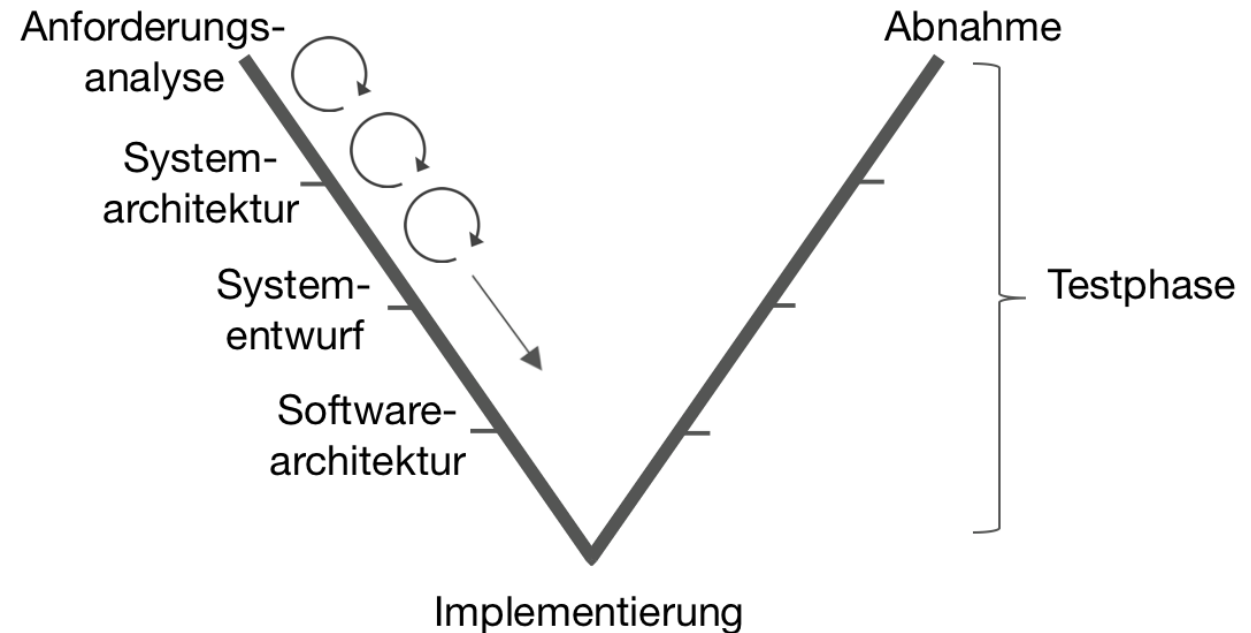
- Sicherheit
  - Kein Schaden für Menschen
  - Schutz der Daten vor Fremdeingriffen
- Over-the-Air Updates
  - Neue Features ohne Werkstattbesuch
  - Bugfixes



# Projektorganisation

- V-SCRUM

- Kombination von V-Modell und SCRUM
- Orientierung an den Phasen des V-Modells
- Klare Zeiteinheiten aus SCRUM
- Nutzung der Struktur Planning, Sprint, Review, Retrospektive
- Arbeiten entlang Zeitplan, Doku-Plan, Implementierungs-Plan



- Geplant:

- Feste Sprintlänge
- Dokumentations-/Konzeptphase
  - Bearbeitung, Review, Überarbeiten
- Implementierungsphase
  - Implementierung, Testen, Überarbeiten
- Gespräche über die Pläne im Plenum
- Arbeiten entlang der Pläne

- Umsetzung:

- Feste Sprintlänge
- Zu Implementierungsbeginn weniger Testen und Überarbeiten
- Gespräche über die Pläne im Plenum wurden später seltener
- Arbeiten entlang Doku-Plan, genereller Zeitplan
- Verschiebung entlang des Implementierungsplan

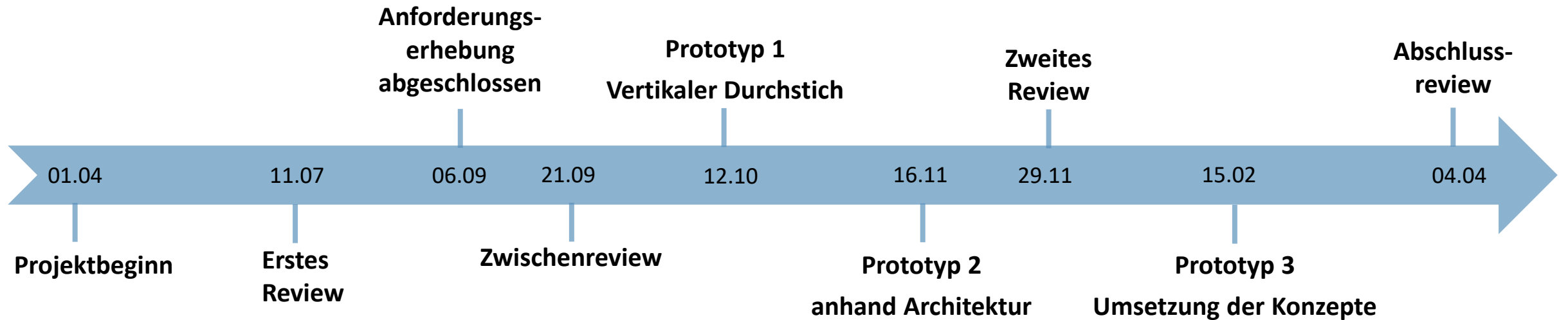


- Geplant:

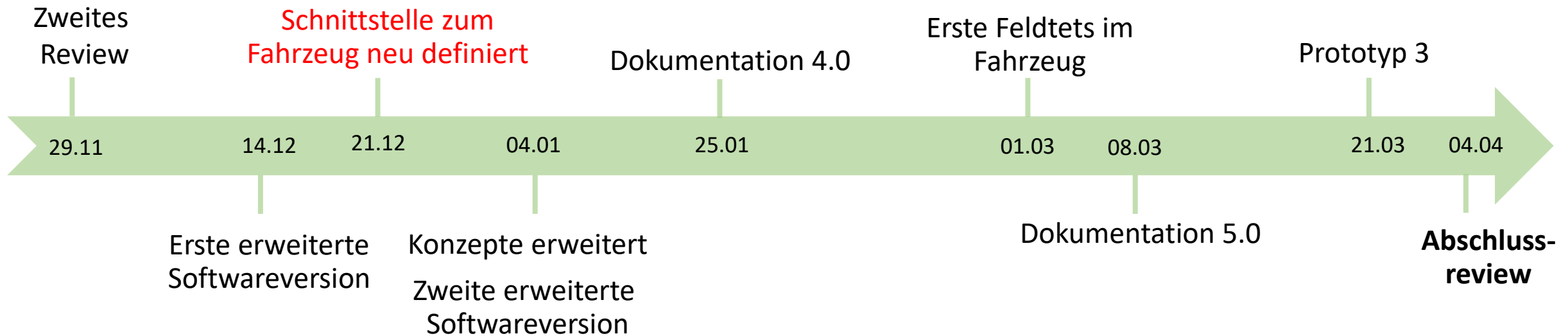
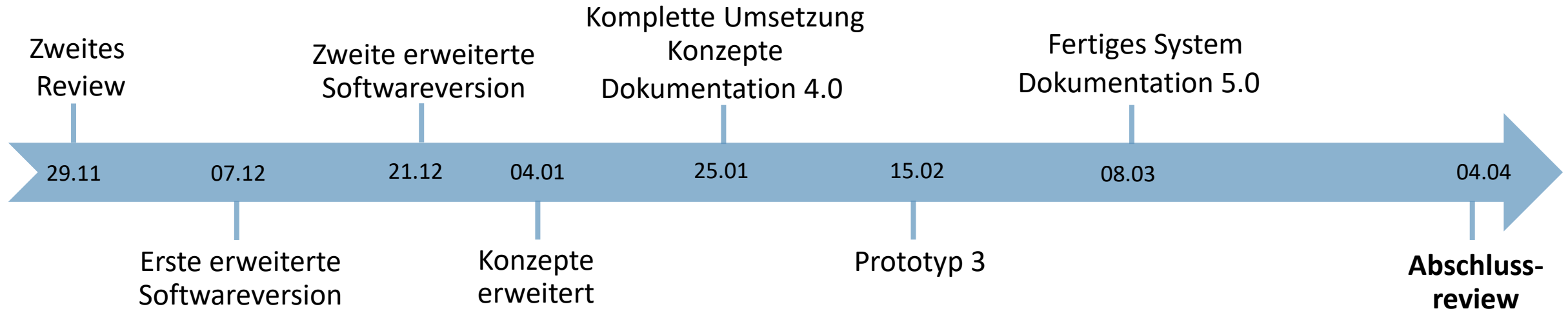
- Arbeiten in Zweiterteams
- Aufbau von Expertenwissen
- Rollenverteilung mit Möglichkeit zum Wechsel
- Nutzung von Jira, Redmine, GanttProject
- Pufferzeiten zum Überarbeiten nutzen

- Umsetzung:

- Größtenteils Einzelarbeit mit Absprachen
- Aufbau von Expertenwissen
- Verbleib in den Rollen
- Nutzung von Jira nur für Doku-Aufgaben, Redmine für Tickets und Anleitungen und GanttProject nur zur Übersicht
- Pufferzeiten zum Arbeiten genutzt



# Zeitplanung 2/2



# Risikoanalyse

- Identifikation, Analyse, Abschätzung von Risiken
- Aufstellen von Maßnahmen

Risiko	Folgen	Maßnahmen
Fehlerhafte Zeitplanung	Verzögerung im Projektplan Nicht erreichen von Projektziel	Einbindung eines Zeitpuffers Ausgleich durch SCRUM
Personelle Ausfälle	Verzögerung im Projektplan Nicht erreichen von Projektziel	Kompensation durch Mehrarbeit Verteilung von Wissen
Technische Probleme	Verzögerung im Projektplan Nicht erreichen von Projektziel	Anpassung der Zeitpläne
Nicht identifizierte Abhängigkeiten zwischen Aufgaben	Verzögerung im Projektplan Nicht erreichen von Projektziel	Kompensation durch Mehrarbeit

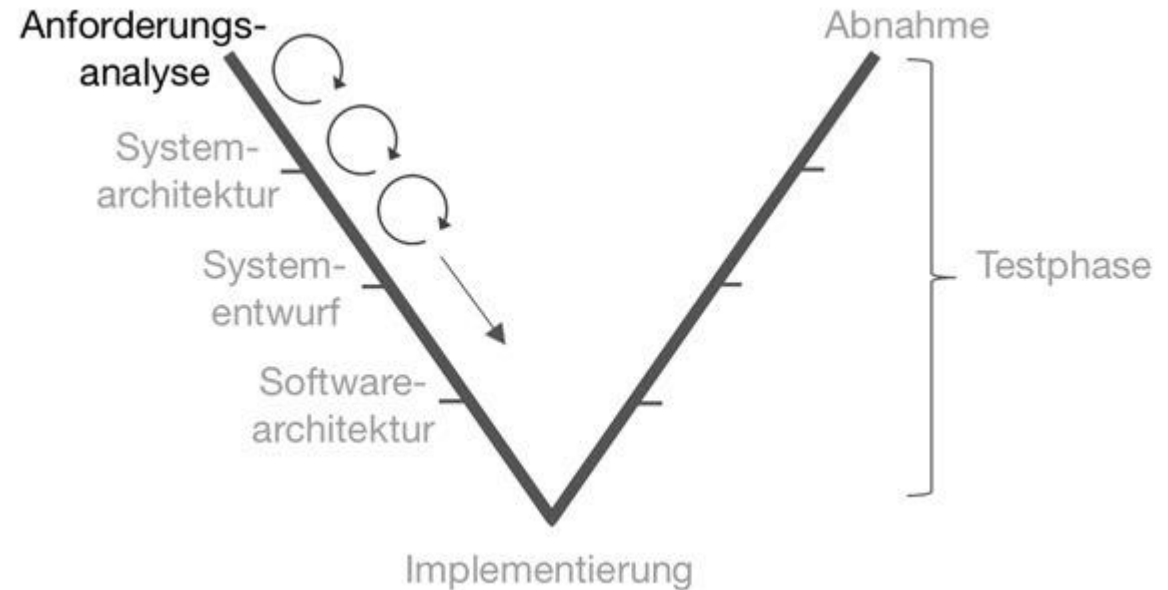
- Arbeiten entlang der Phasen (V-Modell) sehr hilfreich
- Sprints sinnvoll zur Strukturierung
- Rollenverteilung gut
- Ergänzung durch Expertenwissen
- Leben eines industrienahen Prozesses
- Pufferzeiten wurden sinnvoll geplant
- Adäquate Kompensation technischer Probleme
- Abwandlung der Nutzung von projektorganisatorischen Werkzeugen

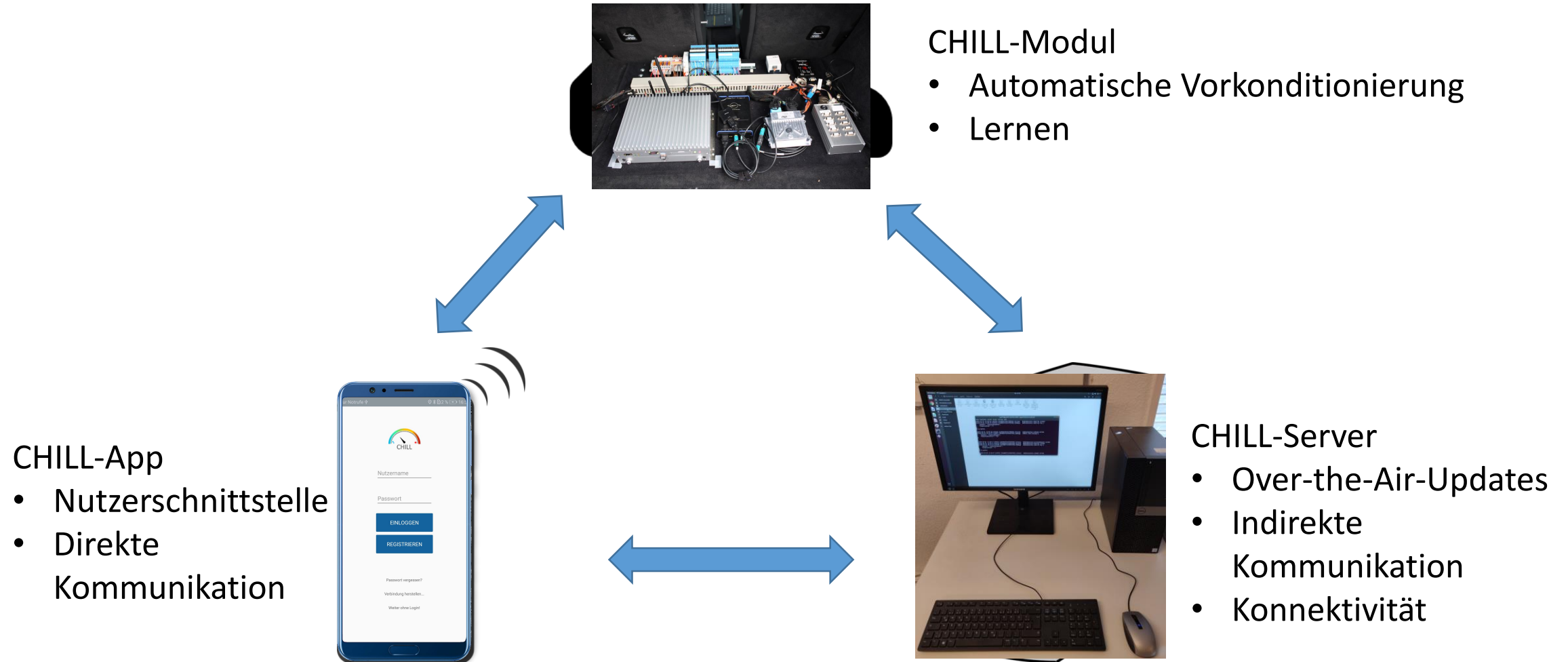
- ✓ Fertigstellen der Dokumentation
  - Fertigstellen von Architektur und Schnittstellen
  - Verfeinern von Konzepten
  - Safetyanalyse erweitern
  - Erfüllung der Anforderungen
  - Prototyp erstellen
  - Vorbereitung auf Veranstaltungen
  - Ausblick auf Erweiterungen



# Von Vision zu Anforderungen

- Grobkonzept
- Vorgehen
- Ergebnisse
  - Top-Anforderungen und Themen
  - Abgeleitete Themen







# Live-Demo

Stakeholderanalyse

4 Gruppen von Stakeholdern indentifiziert:  
Auftraggeber, Nutzer, Entwickler, Außenstehende

User-Stories

55 User-Stories von der Form:  
„Als [Rolle] möchte ich [Eigenschaft], damit [Gewinn]“

Anwendungsfälle

9 konkrete Anwendungsfälle als UML-Diagramme

Anforderungen

# Themen und Top-Level-Anforderungen



## Themen

Konditionierung

Lernen

Selbstständigkeit

App

Server

Modulkommunikation

Fahrererkennung (Näherung)

Umgebungsklassifikation

Safety

OTA-Updates

Absicherung (Fremdeinwirkung)

Datenschutz

Klimamodell

muss

soll

kann

top-fkt-otaupdates

top-nfkt-safety

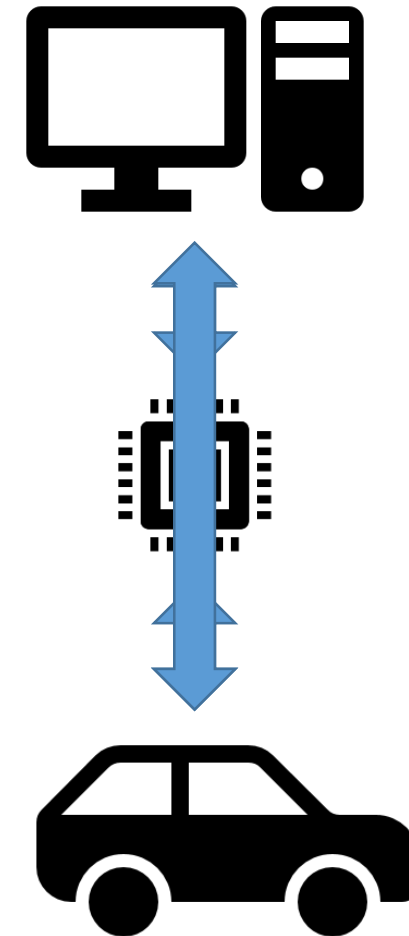
top-nfkt-sicherung

top-nfkt-datenschutz

top-nfkt-benutzerfreundlichkeit

## Konditionierung

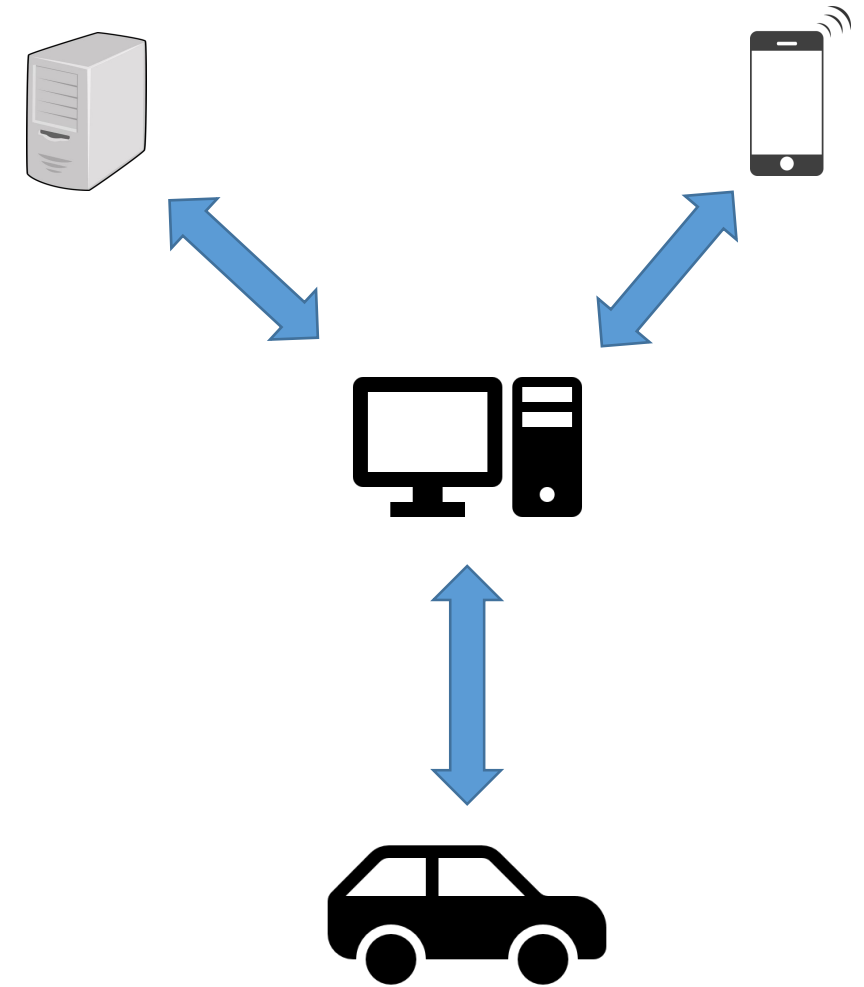
- CHILL-Modul:
  - Setzen relevanter Aktorik
  - Informationen über aktuellen Zustand
- Schnittstelle zum Fahrzeug
  - Versuchsträger: IAV-Gateway





## Selbstständigkeit

- CHILL-Modul:
  - Bestimmung wann welche Aktorik zu setzen ist
  - Sammlung relevanter Informationen:
    - Bekannte Informationen
    - Externe Informationen (App, Server, Fahrzeug)
    - Gelernte Informationen
- Konzept: Bestimmung notwendiger Klimateinstellungen für ein gegebenes Ziel
- Konzept: Bestimmung, ob sich der Fahrer annähert



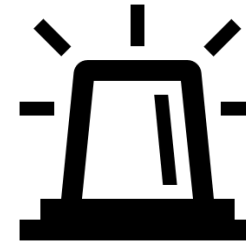
## Lernen

- CHILL-Modul:
  - Observierung des Nutzerverhaltens
    - Konditionierungspräferenzen
    - Abfahrtszeiten
- Konzept: Extrapolation und Vorhersage
  - Tatsächliche Präferenzen unter verschiedenen Bedingungen identifizieren
  - Zukünftige Abfahrtszeiten vorhersagen



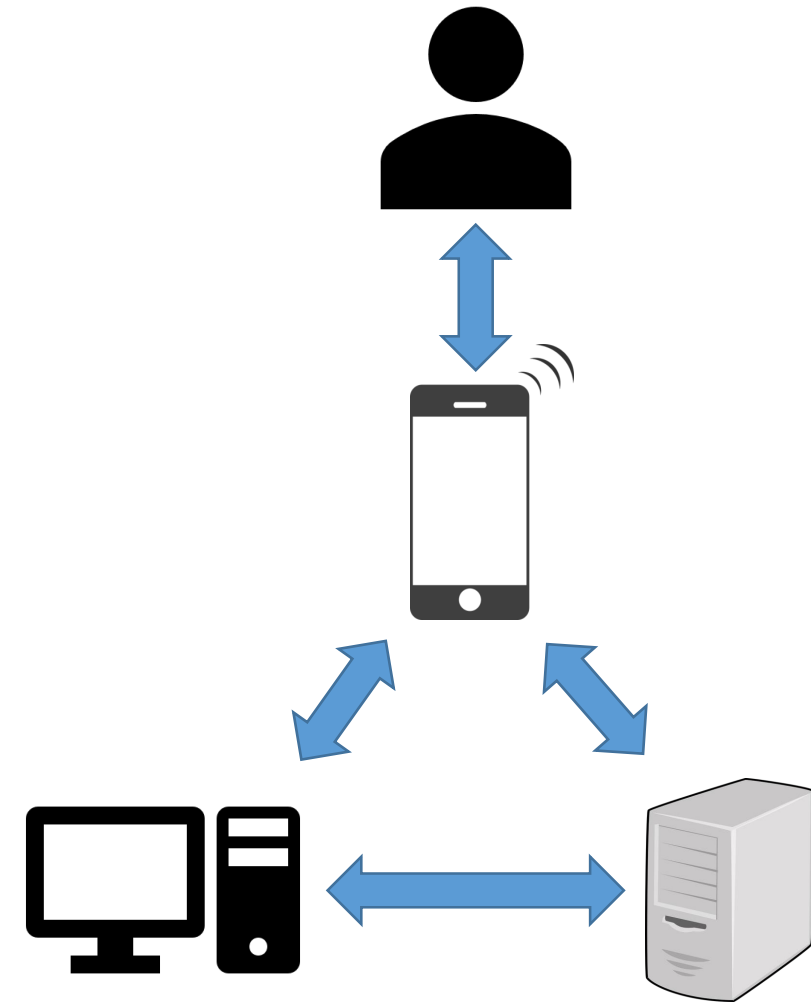
## Safety

- Einbettung in ein sicherheitskritisches System
- CHILL-Modul:
  - Erkennen systembezogener Gefahren
    - Umgebungserkennung
    - Fahrtzustand
  - Umsetzung entsprechender Vorkehrungen und Gegenmaßnahmen
- Konzept: Klassifizierung der Umgebung



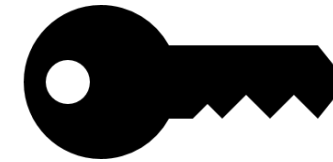
## Nutzerschnittstelle

- CHILL-App:
  - Einstellungen für das CHILL-Modul
  - Planung von Vorkonditionierungen
  - Benachrichtigungen
- CHILL-Server:
  - Sekundärer Kommunikationskanal
- Konzept: Softwareupdates ohne Hardwarezugriff (OTA)



## Datenschutz und Security

- CHILL-App, CHILL-Server, CHILL-Modul:
  - Absicherung von Kommunikationskanälen
  - Authentifizierung und Authorisierung
  - Schutz von Nutzerdaten
    - Möglichkeit der Löschung
    - Keine Erhebung unnötiger Daten

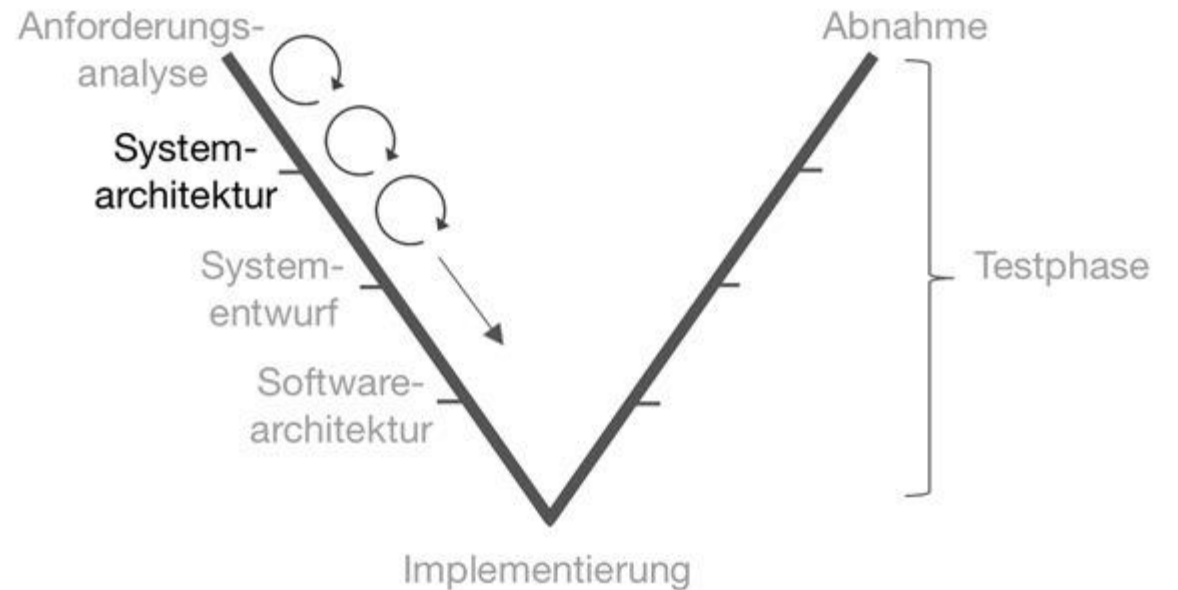


# Konzeption

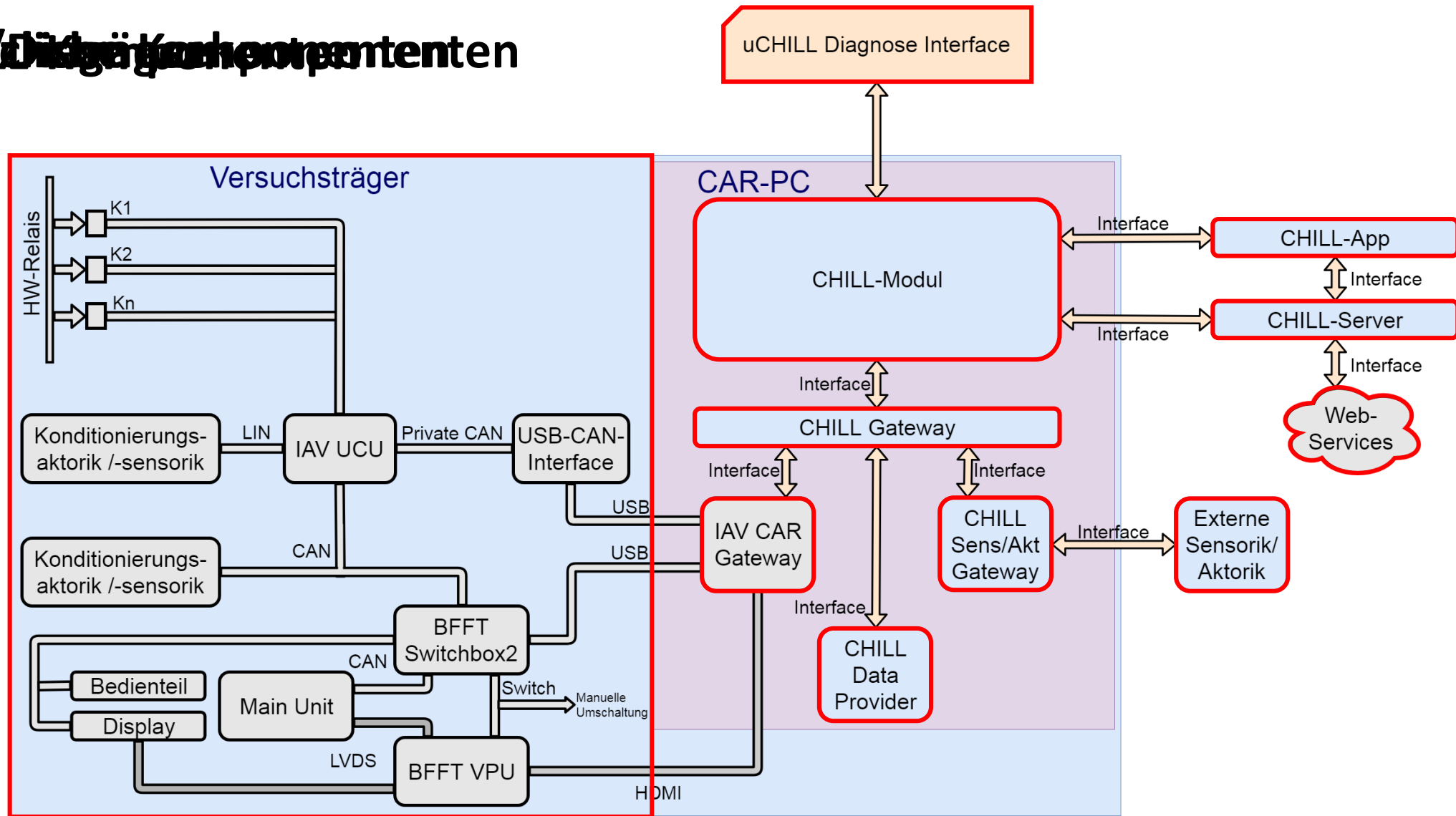
- Architektur und Schnittstellen
- Konzepte
  - Klimamodell
  - Künstliche Intelligenz
  - Umgebungserkennung
  - Annäherungserkennung
  - OTA-Updates



- Architektur und Schnittstellen
- Konzepte
  - Klimamodell
  - Künstliche Intelligenz
  - Umgebungserkennung
  - Annäherungserkennung
  - OTA-Updates



## Test/Diagnosekomponenten

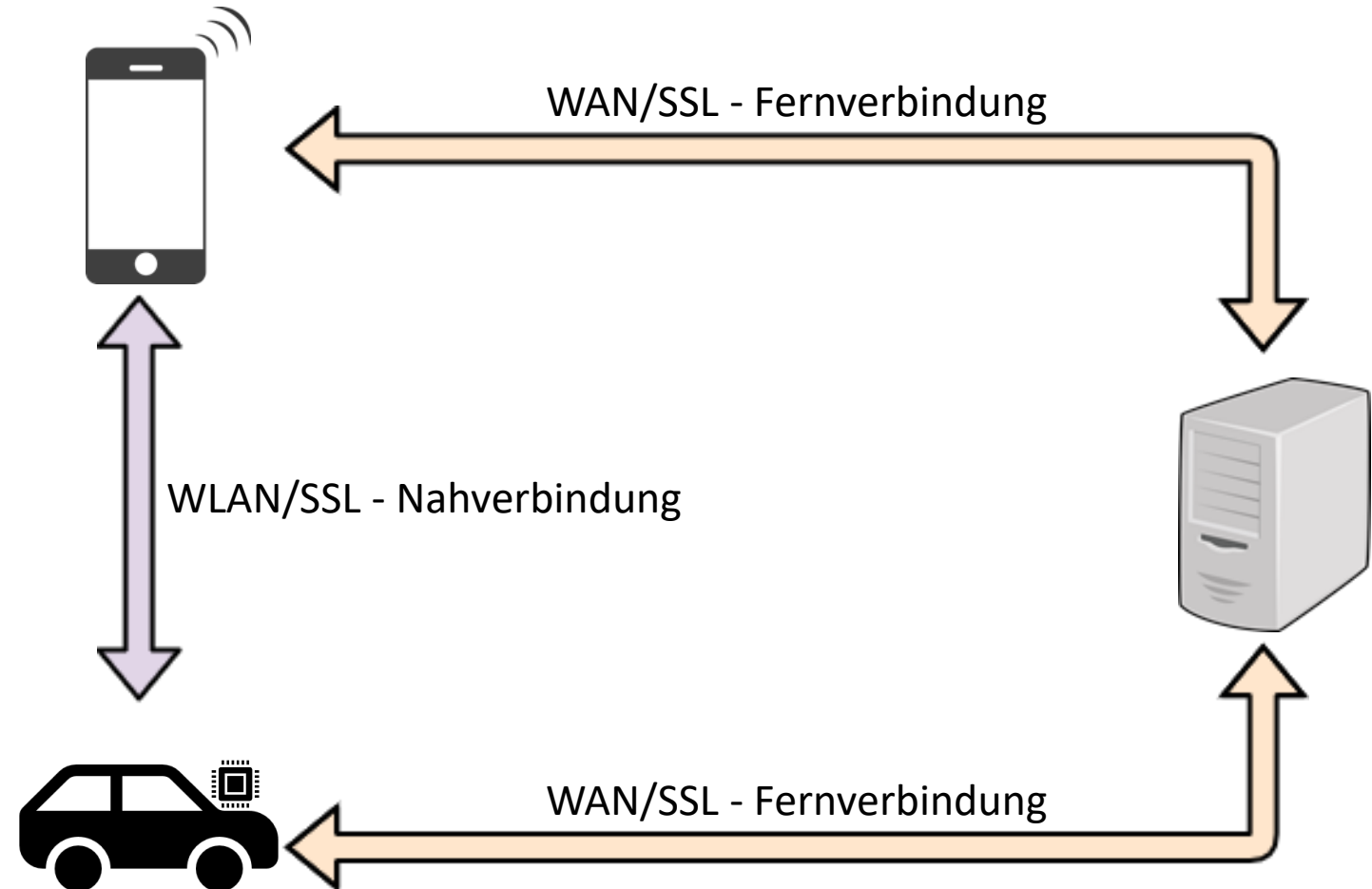


## Nahverbindung

- Zugriff ohne Internet
- Lokales WLAN

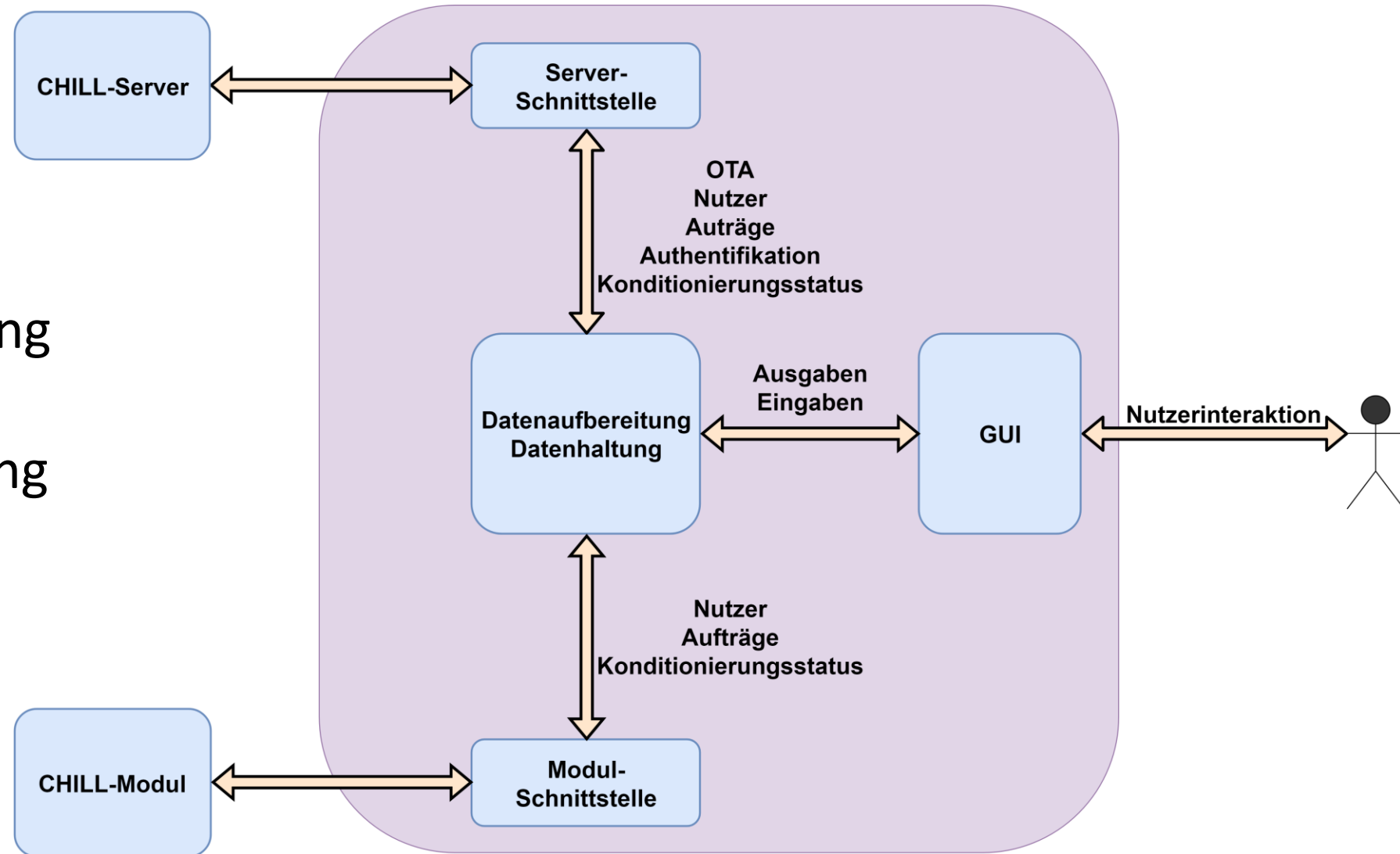
## Fernverbindung

- Umweg über Server
- Fernzugriff



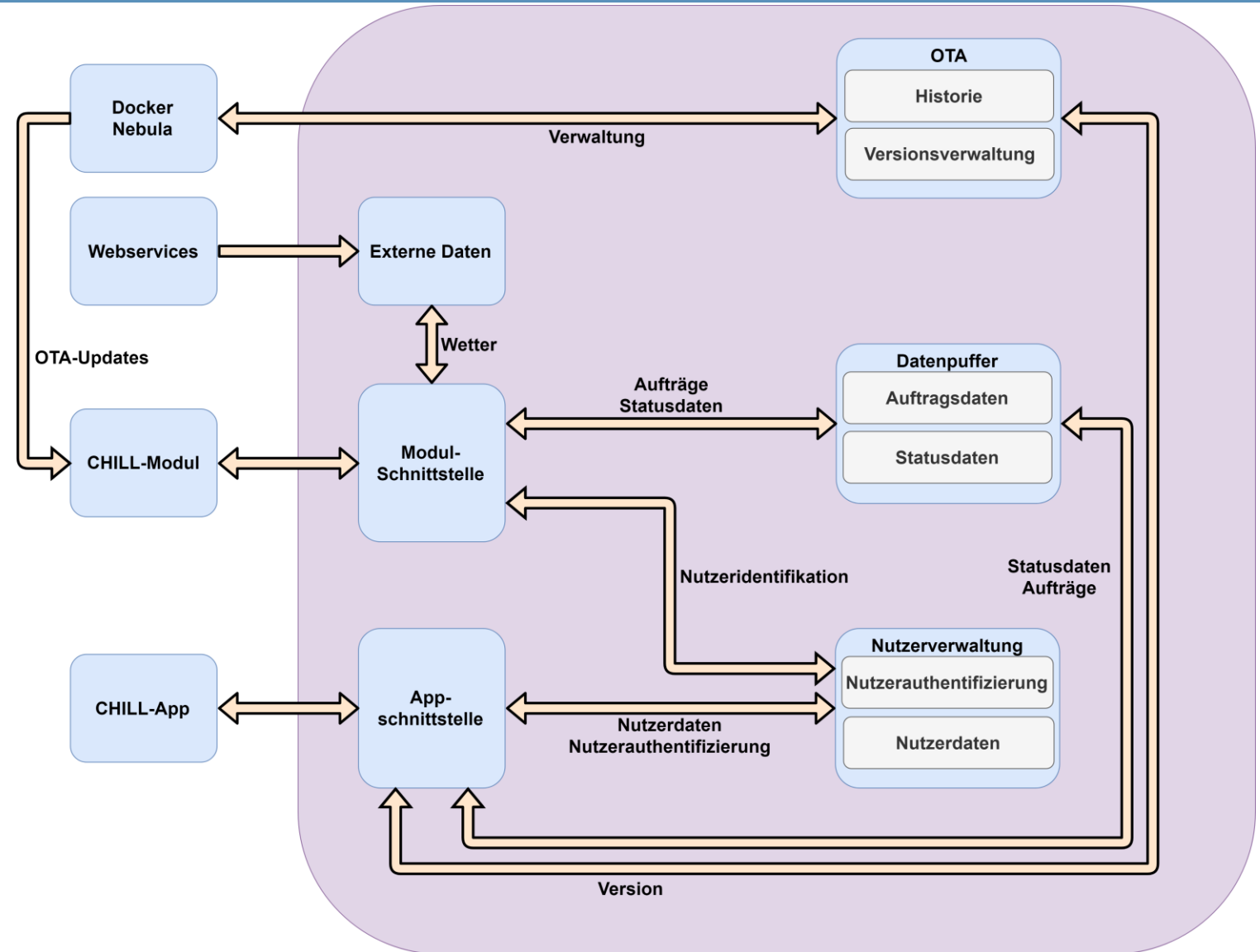
## Aufgaben:

- Nutzerschnittstelle
  - Terminverwaltung
  - Konfiguration
  - Nutzerverwaltung
  - Verwaltung OTA-Updates



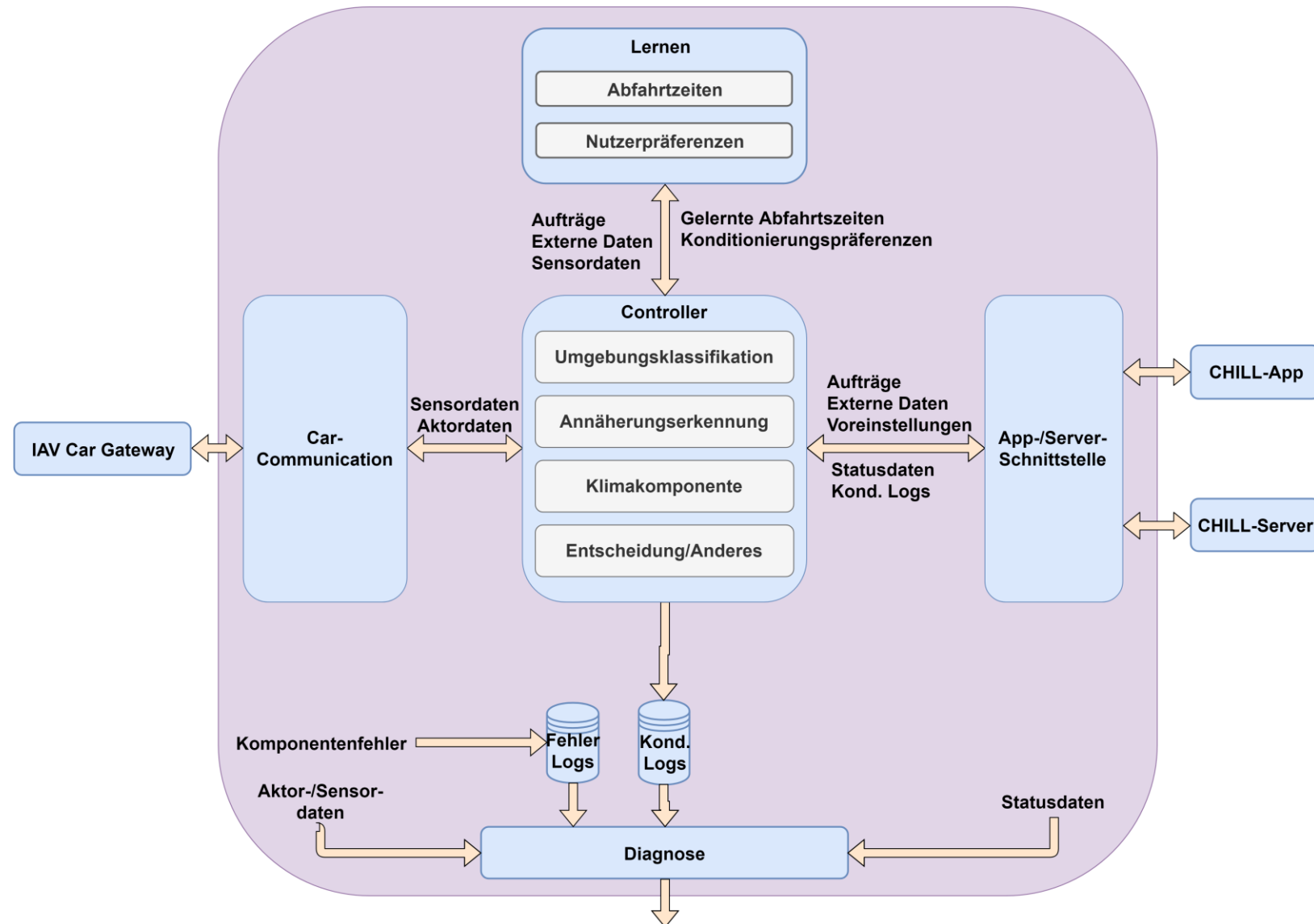
## Aufgaben:

- Durchführung OTA Updates
- Weiterleiten von Aufträgen
- Nutzerverwaltung
- Externe Daten



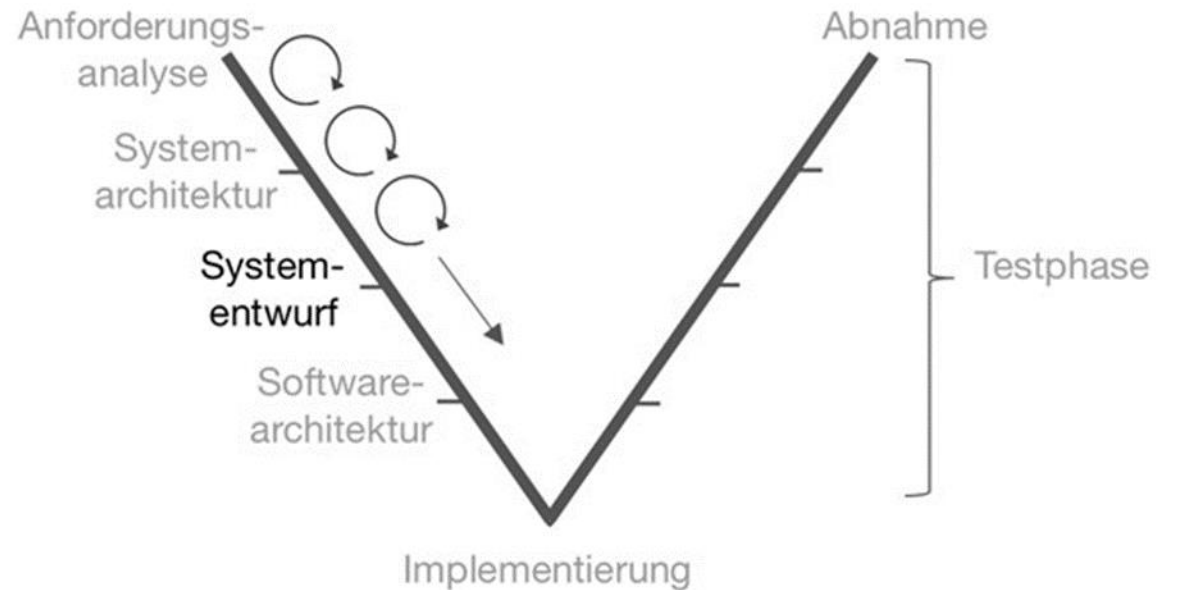
## Aufgaben:

- Umsetzung der Vorkonditionierung
- Ansteuerung der relevanten Aktorik/Sensorik
- Verarbeiten von Aufträgen
- Lernen



- ✓ Fertigstellen der Dokumentation
- ✓ Fertigstellung von Architektur und Schnittstellen
  - Verfeinern von Konzepten
  - Safetyanalyse erweitern
  - Erfüllung der Anforderungen
  - Prototyp erstellen
  - Vorbereitung auf Veranstaltungen
  - Ausblick auf Erweiterungen

- Architektur und Schnittstellen
- Konzepte
  - Klimamodell
  - Künstliche Intelligenz
  - Umgebungserkennung
  - Annäherungserkennung
  - OTA-Updates



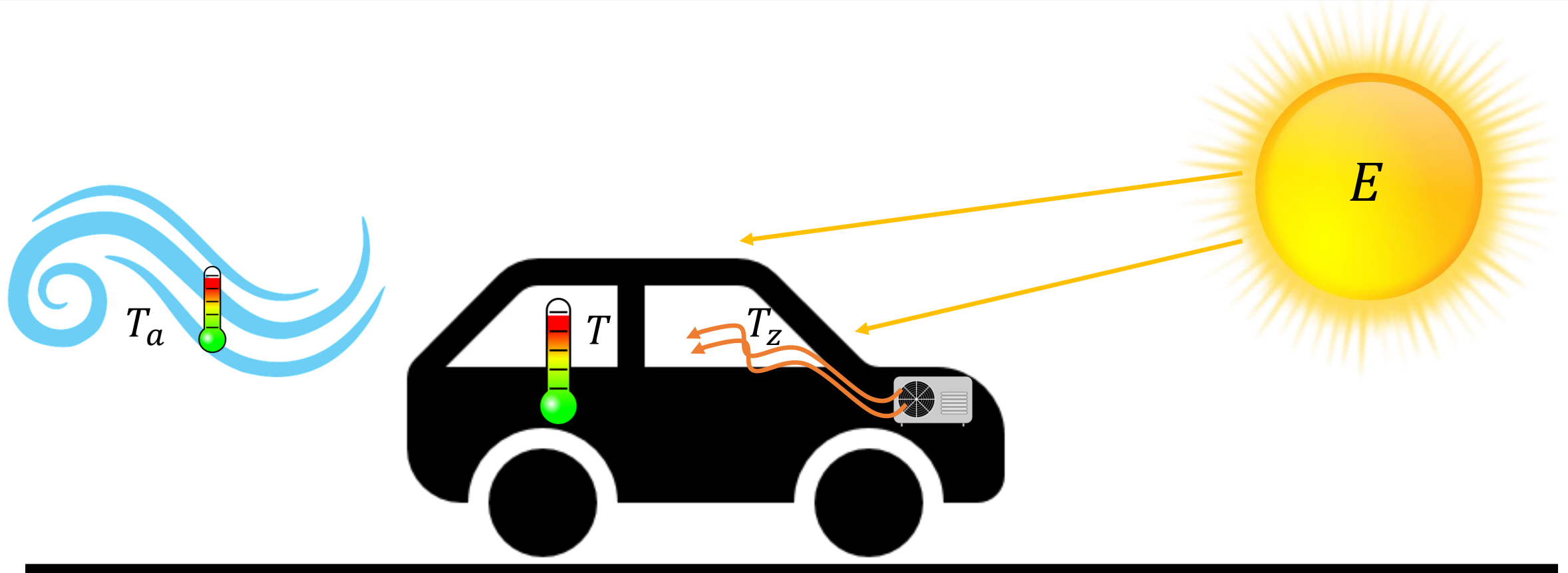


# Klimamodell

Bestimmung notwendiger Klimaeinstellungen für ein gegebenes Ziel



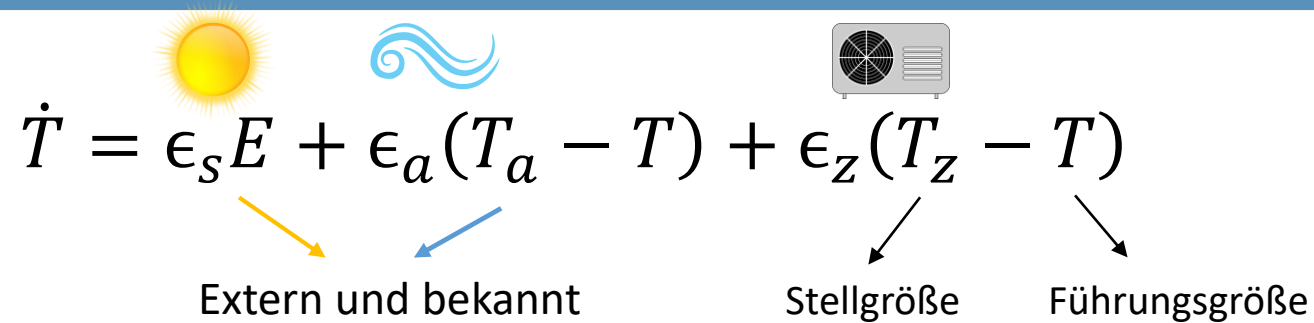
- Alter Ansatz
  - 30 Minuten vor Abfahrt Standklimatisierung auf die gewünschte Temperatur einstellen
  - Keine Anpassung der Temperatur während der Laufzeit
  - Keine äußeren Einflüsse betrachtet
- Neuer Ansatz
  - Zunächst ein Temperaturmodell
  - Vereinfachungen:
    - Luftmasse bleibt gleich
    - Wärmekapazität ist im gesamten Innenraum gleich



$$\dot{T} = \epsilon_s E + \epsilon_a (T_a - T) + \epsilon_z (T_z - T)$$


$$\dot{T} = \epsilon_s E + \epsilon_a (T_a - T) + \epsilon_z (T_z - T)$$

- **Wärmestrahlung:**  $\epsilon_s E$   
 $E$ : einwirkende Wärmestrahlung ( $\text{W/m}^2$ )  
 $\epsilon_s$ : beinhaltet Fläche der betroffenen Einstrahlung
- **Wärmeleitung:**  $\epsilon_a (T_a - T)$  nach Newtons Abkühlungsgesetz  
 $T_a$  anliegende Außentemperatur  
 $\epsilon_a$  beinhaltet Flächeninhalt, Wärmeleitfähigkeit
- **Konvektion:**  $\epsilon_z (T_z - T)$   
 $T_z$  Temperatur der zugeführten Luft  
 $\epsilon_z$  beinhaltet Massenstrom der zugeführten Luftmasse


$$\dot{T} = \epsilon_s E + \epsilon_a (T_a - T) + \epsilon_z (T_z - T)$$

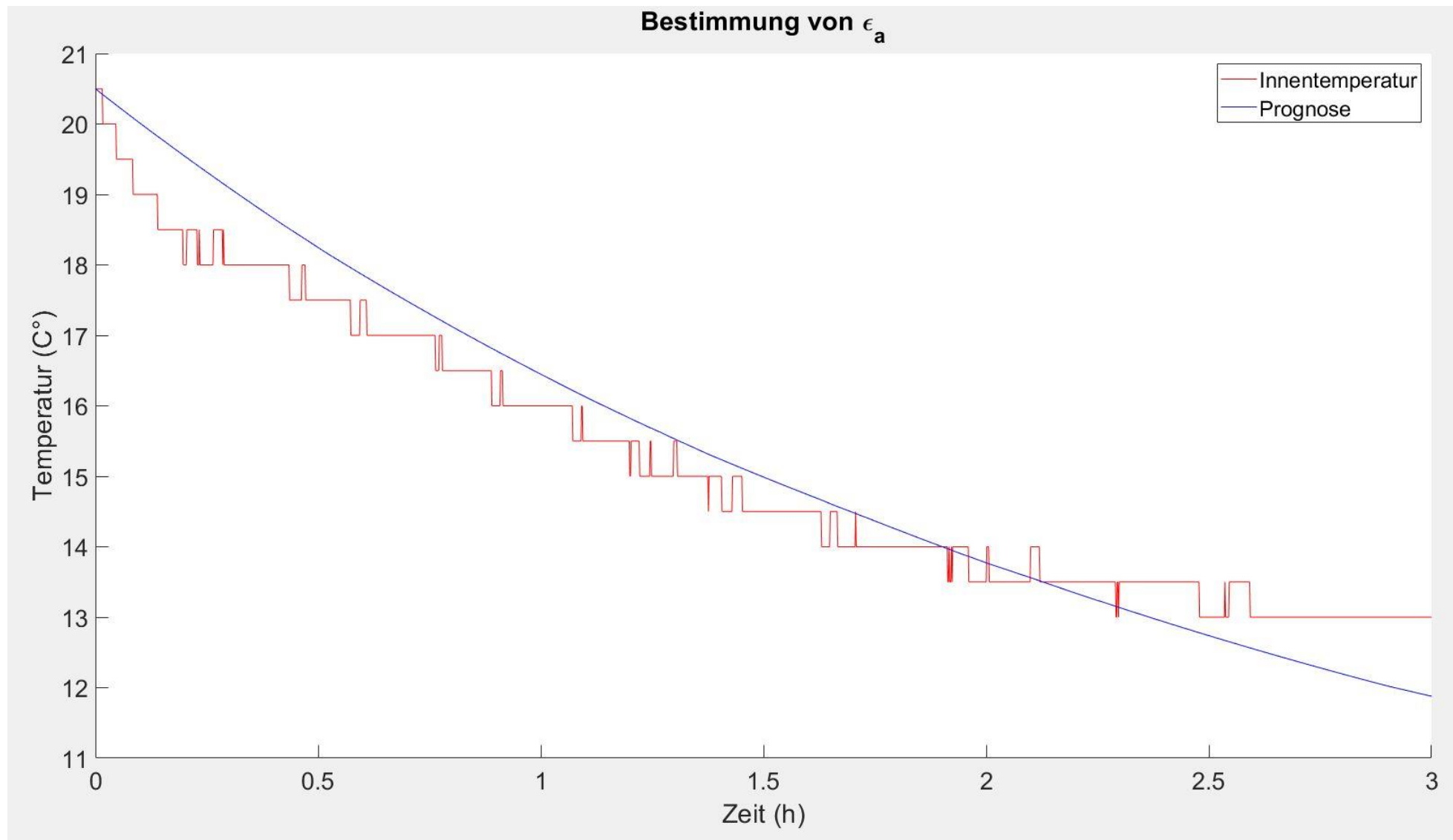
Extern und bekannt      Stellgröße      Führungsgröße

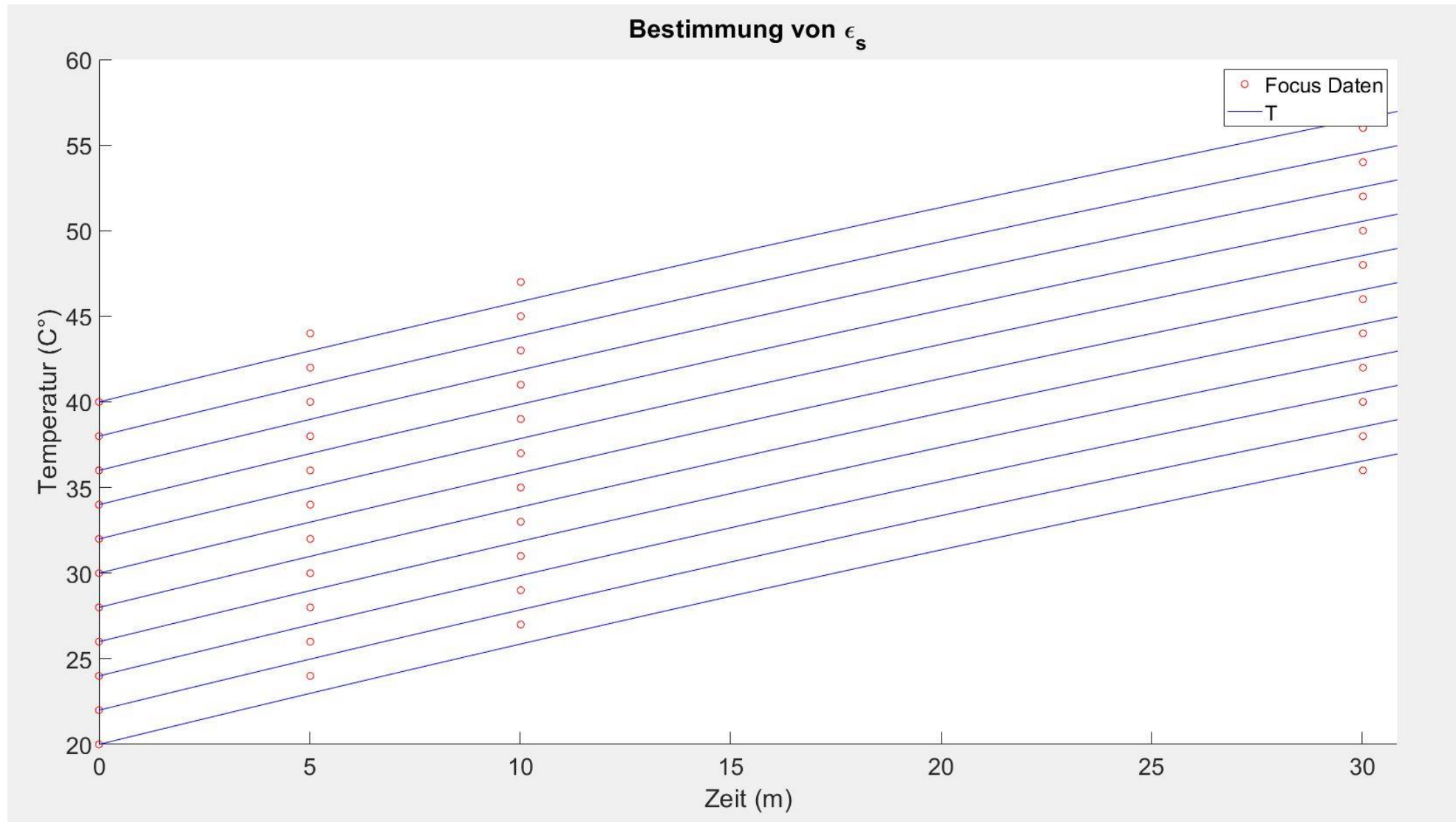
- Für bekannte Entwicklung der Außentemperatur  $T_a(t)$  und Sonneneinstrahlung  $E(t)$  (aus den Wetterdaten) schreiben wir  $\dot{T} = f(T, T_z)$
- Linearisierung über Zeitintervalle  $\delta$ :

$$T_{k+1} = T_k + \delta f(T_k, T_{z_k})$$

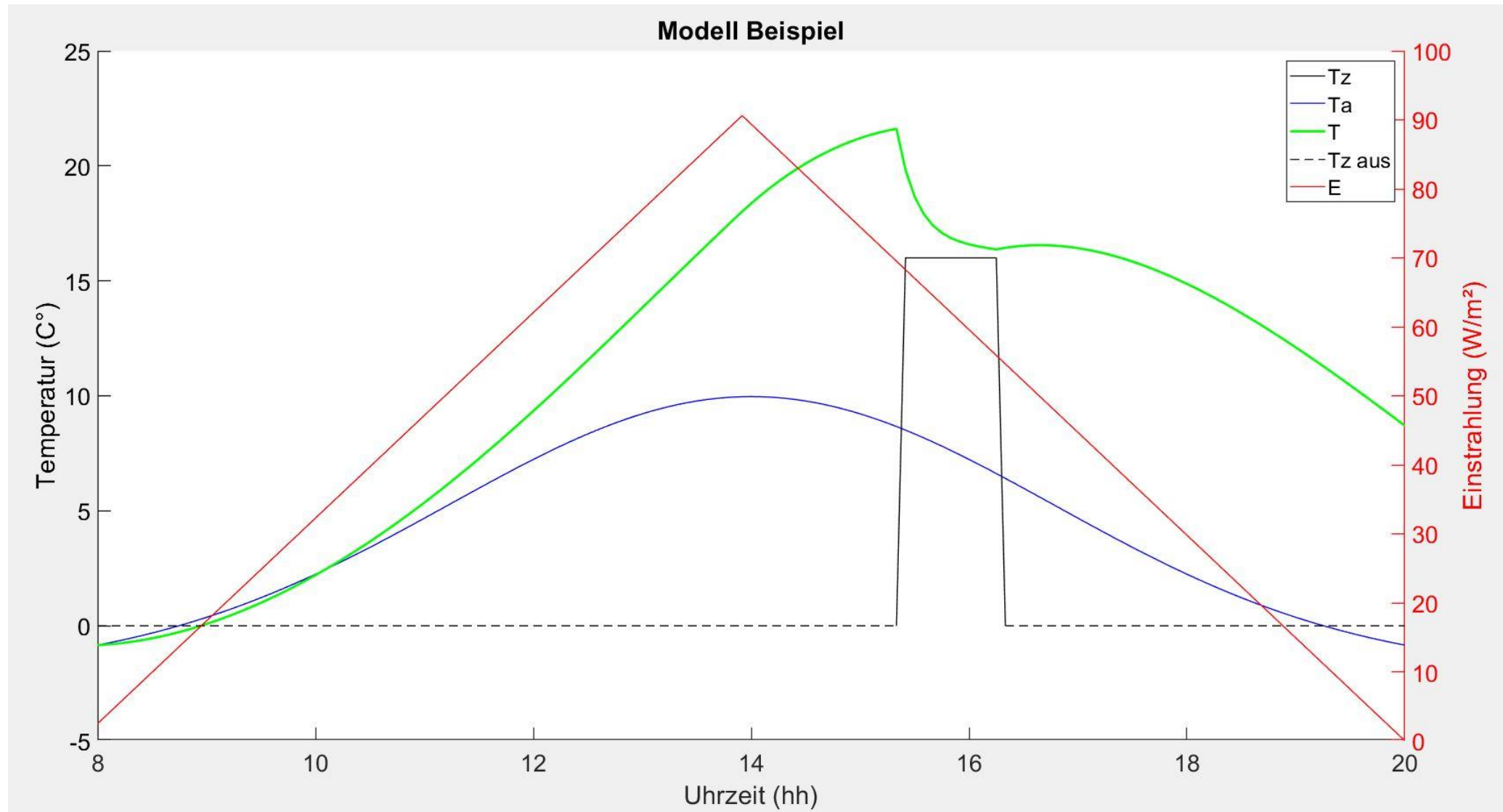
- Mit dieser Modellgleichung kann nun die Entwicklung der Temperatur über die Zeit bestimmt werden

# Einfluss der Isolierung & Außentemperatur





# Temperaturverlauf über Tag





- Ziel: Minimiere

$$\sum_{k=0}^{N-1} (T_{z_k} - T_k)^2$$

unter dem Modell:

$$T_{k+1} = T_k + \delta f(T_k, T_{z_k})$$

mit der Nebenbedingung:

$$T_{z_{min}} \leq T_z \leq T_{z_{max}}$$

und dem Ziel

$T_N$  (Zieltemperatur)

- Keine Informationen über den Aktor zur Temperaturzuführung
  - Selbstständige Regelung
  - Kein Modell zur Verfügung
- Umsetzung des alten Ansatzes
  - 30 Minuten vor Abfahrt die gewünschte Zieltemperatur in der Standklimatisierung
- Wenn Aktor bekannt
  - Können weiterführende Evaluationen folgen
  - Daraufhin kann das Modell vollständig beherrscht werden
  - Die Temperatur im Fahrzeug kann realitätsnah modelliert werden

- Deaktivieren der Standklimatisierung während der Vorkonditionierung
- Heiße Sommertage evaluieren
- Evaluierung der Standklimatisierung
- Modell für Luftfeuchtigkeit
- Weitere Aktoren zur Klima-Beeinflussung betrachten (Rollos, Fenster)
- Weitere Faktoren betrachten: Abwärme von Batterie und CarPC

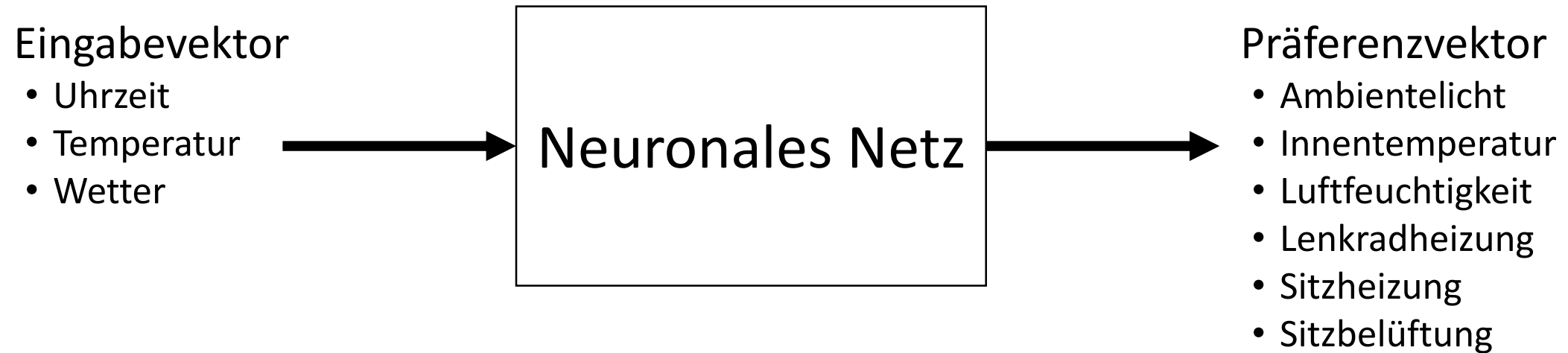
# Zeiten lernen

Extrapolation und Vorhersage von Abfahrtszeiten

- Annahme: Abfahrtszeiten unabhängig von messbaren Parametern
  - Bisheriger Zeitverlauf einzige Informationsquelle
- Autoregressiver Ansatz
  - Vorhersage eines zukünftigen Punktes durch  $p$  vergangene Punkte
  - Modell:  $X_t = \sum_{i=1}^p \phi_i X_{t-i}$
- Problem: Wie bestimmt man die Koeffizienten  $\phi_i$ ?
- Einführung eines Fehlerterms, welcher minimiert wird
- Ansatz: Linear Predictive Coding (LPC)
  - Verfahren aus der Signalverarbeitung
  - Modelliert Systeme durch Beobachtung des zeitlichen Verhaltens

# Präferenzen lernen

Extrapolation und Vorhersage von Präferenzen



**Eingabevektor = [**

**Uhrzeit:** {(Sinustransformation, Kosinustransformation)},

**Außentemperatur:** {-30, -29, ..., 60},

**Wetter:** {sonne, gewitter, regen, sturm, schnee, wind, bewoelkt}

**]**



**Eingabevektor = [**

**Uhrzeit:** {(Sinustransformation, Kosinustransformation)},

**Außentemperatur:** {-30, -29, ..., 60},

**Wetter:** {sonne, **gewitter**, regen, sturm, schnee, wind, bewoelkt}

**]**

- Wetter one-hot kodiert
  - Nur ein Wert kann aktiv sein
  - Nicht geordnet

**Eingabevektor = [**

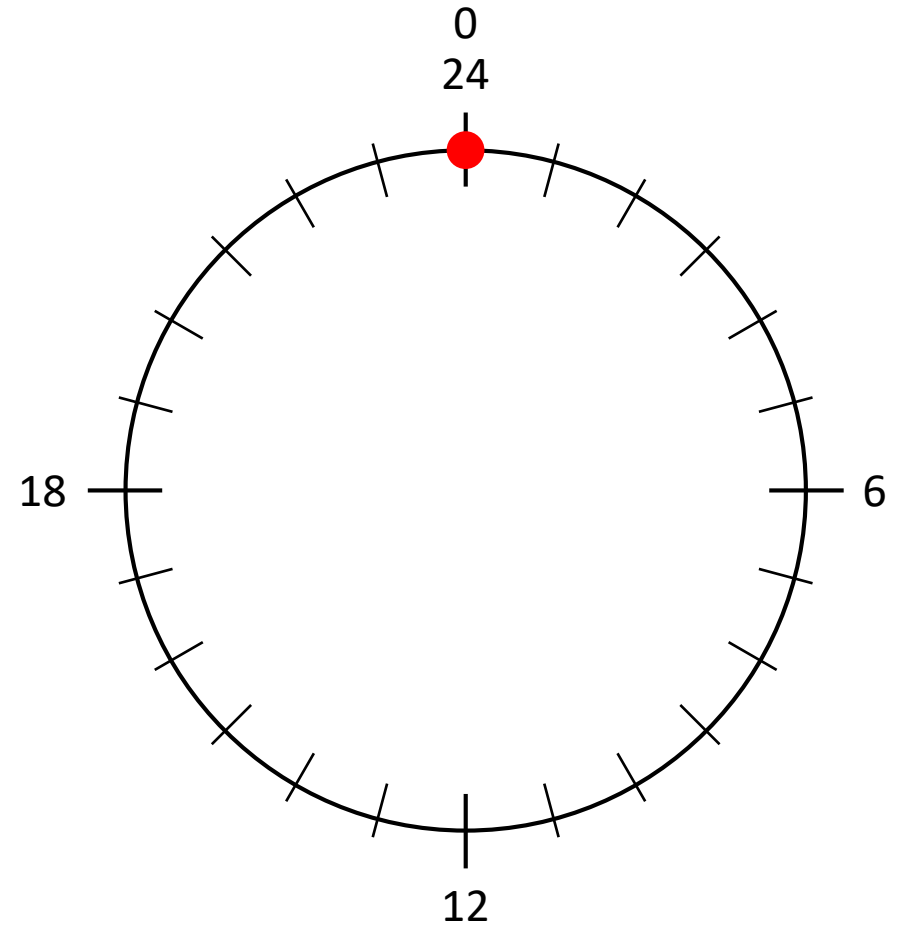
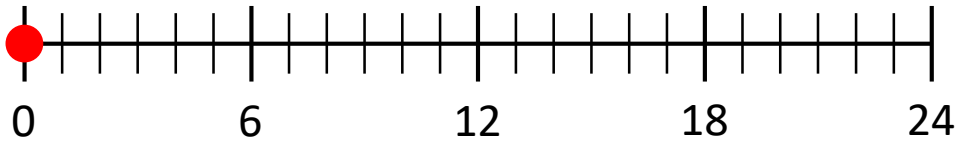
**Uhrzeit:** {(Sinustransformation, Kosinustransformation)},

**Außentemperatur:** {-30, -29, ..., 60},

**Wetter:** {sonne, gewitter, regen, **sturm**, schnee, wind, bewoelkt}

**]**

- Wetter one-hot kodiert
  - Nur ein Wert kann aktiv sein
  - Nicht geordnet



**Outputvektor = [**

**Ambientelicht:** {gelb, rot, blau, violett, weiss, tuerkis, gruen},

**Innentemperatur:** {kalt, kuehl, neutral, warm, heiss},

**Luftfeuchtigkeit:** {stufe0, stufe1, ..., stufe10},

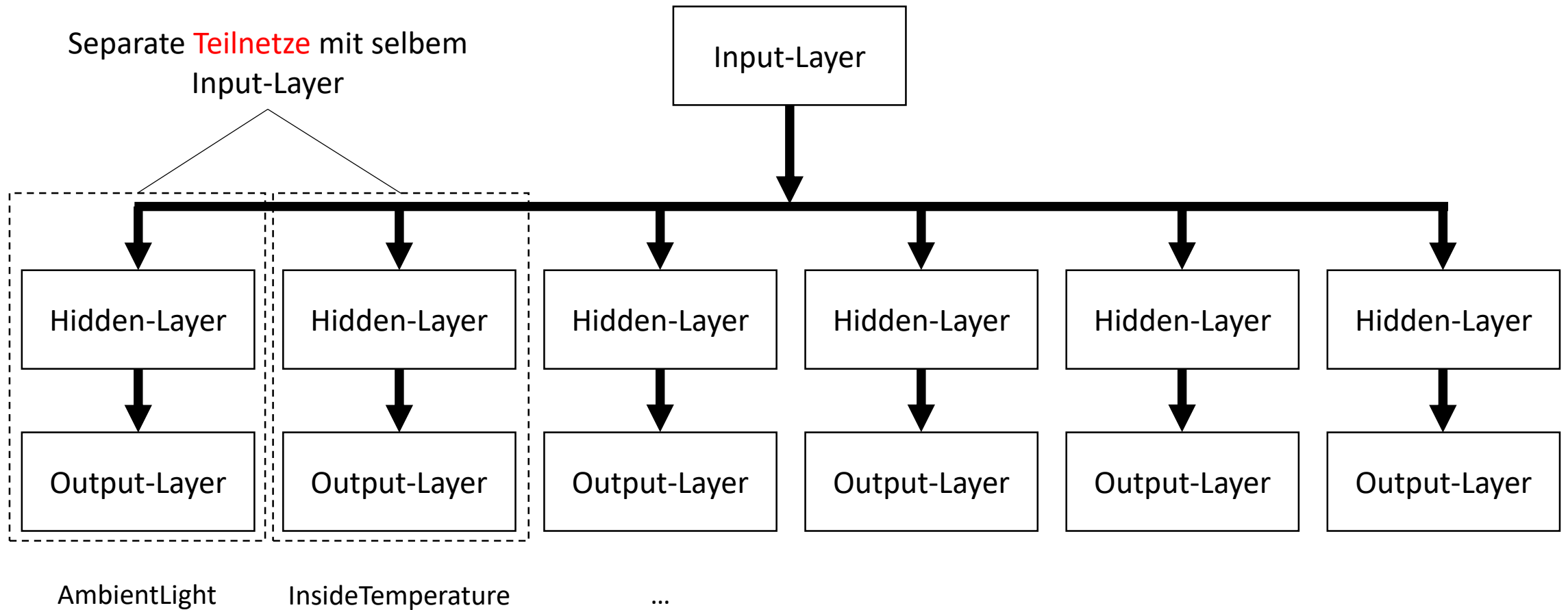
**Lenkradheizung:** {ein, aus},

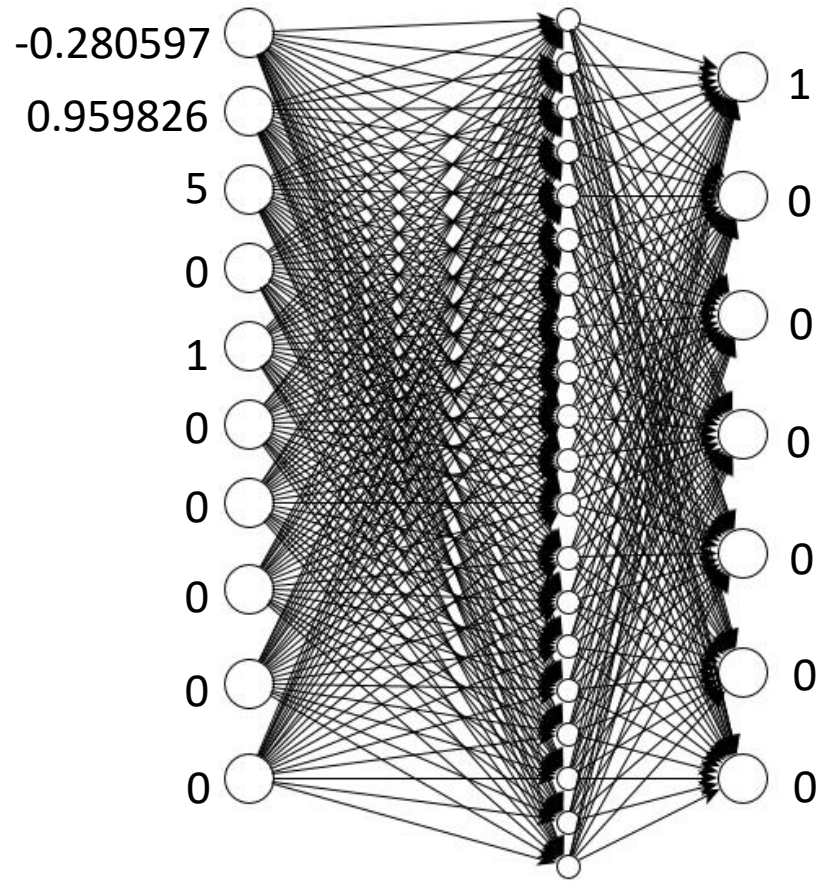
**Sitzheizung:** {stufe0, stufe1, stufe2, stufe3},

**Sitzbelüftung:** {stufe0, stufe1, stufe2, stufe3}

**]**

- Vereinfachung: Abbildung der Temperatur auf Kategorien





**Input-Neuronen:** 10

**Hidden-Neuronen:** 20

**Output-Neuronen:** Hängt von Anzahl der Label ab  
Hier 7 für Ambientelicht

# Umgebungserkennung

## Klassifizierung der Umgebung

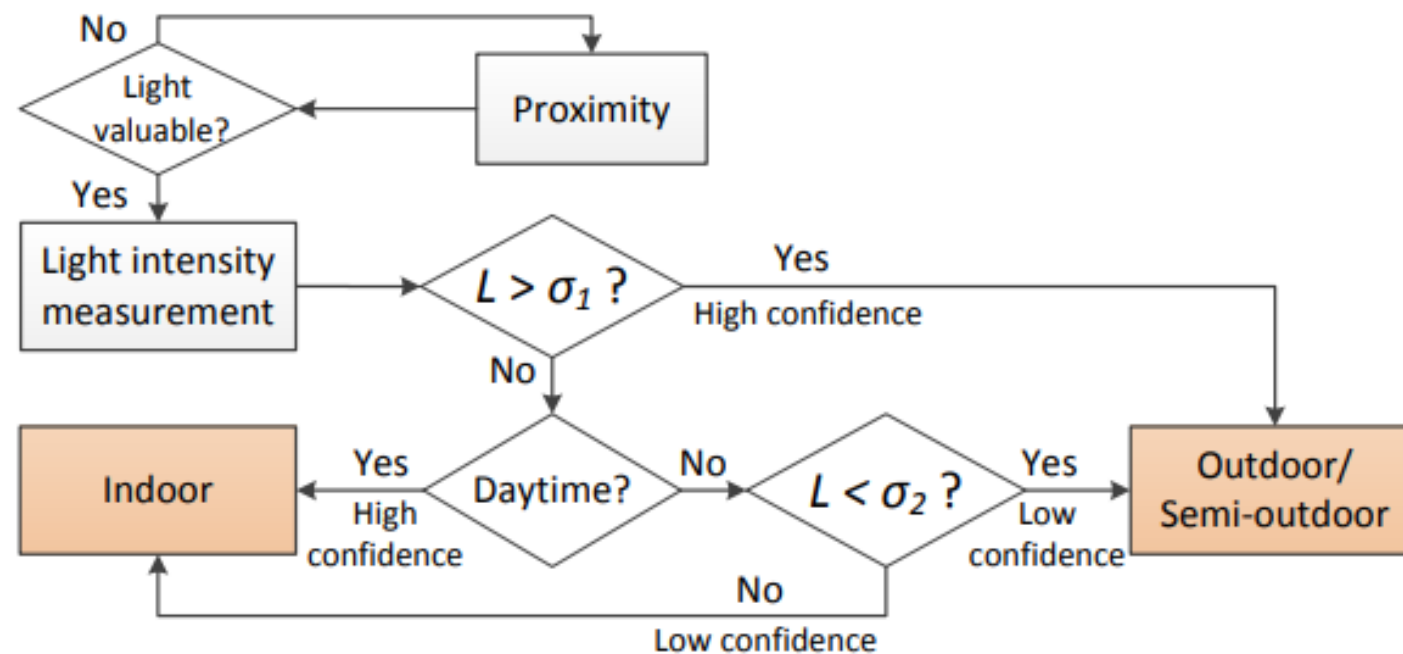
- Unterscheidung zwischen Indoor, Outdoor und Semi-Outdoor
- Ziel → Entscheiden, ob Verbrennungsmotor gestartet werden darf

Umgebung	Outdoor	Semi-Outdoor	Indoor
Definition	Öffentlicher Parkplatz (draußen)	Neben einem Gebäude (überdacht)	Garage (geschlossen)
Beispiel			
Bild			 <a href="http://www.mietgarage-usu.de/Garage-1-6">http://www.mietgarage-usu.de/Garage-1-6</a>



- GPS
  - Empfang: hohe Wahrscheinlichkeit für Outdoor
  - Kein Empfang: hohe Wahrscheinlichkeit für Indoor

- Lichtsensor:



Pengfei Zhou u. a. *IODetector: A Generic Service for Indoor Outdoor Detection*.2012.

- Feldtests mit Mess-Script
  - Auslesen von Sensoren an verschiedenen Standorten
  - Unterschiedliche Tageszeiten und Wetterverhältnisse
  - 26 Messungen an 18 Standorten



Messung Parkhaus Universität



```
[2019-03-07 17:41:25] - Log: outsideTemperature: 12.000000  
[2019-03-07 17:41:25] - Log: globalInsideTemp: 18.500000  
[2019-03-07 17:41:25] - Log: humidity: 41.000000  
[2019-03-07 17:41:25] - Log: windowTemperature: 13.700000  
[2019-03-07 17:41:25] - Log: thermalInput: 0.000000  
[2019-03-07 17:41:25] - Log: visibleLight: 0.000000  
[2019-03-07 17:41:25] - Log: infraredLight: 0.000000  
[2019-03-07 17:41:25] - Log: currentRainIntensity: 0
```

Ausschnitt aus Log des Mess-Scripts

- Nutzbare Sensoren:
  - Infrarotlicht, sichtbares Licht
  - Wärmeeinfluss
  - Regenintensität
- GPS liefert aufbereitete Werte
  - → Nicht für Klassifizierung nutzbar
- Infrarot und sichtbares Licht nachts nicht zur Klassifizierung nutzbar

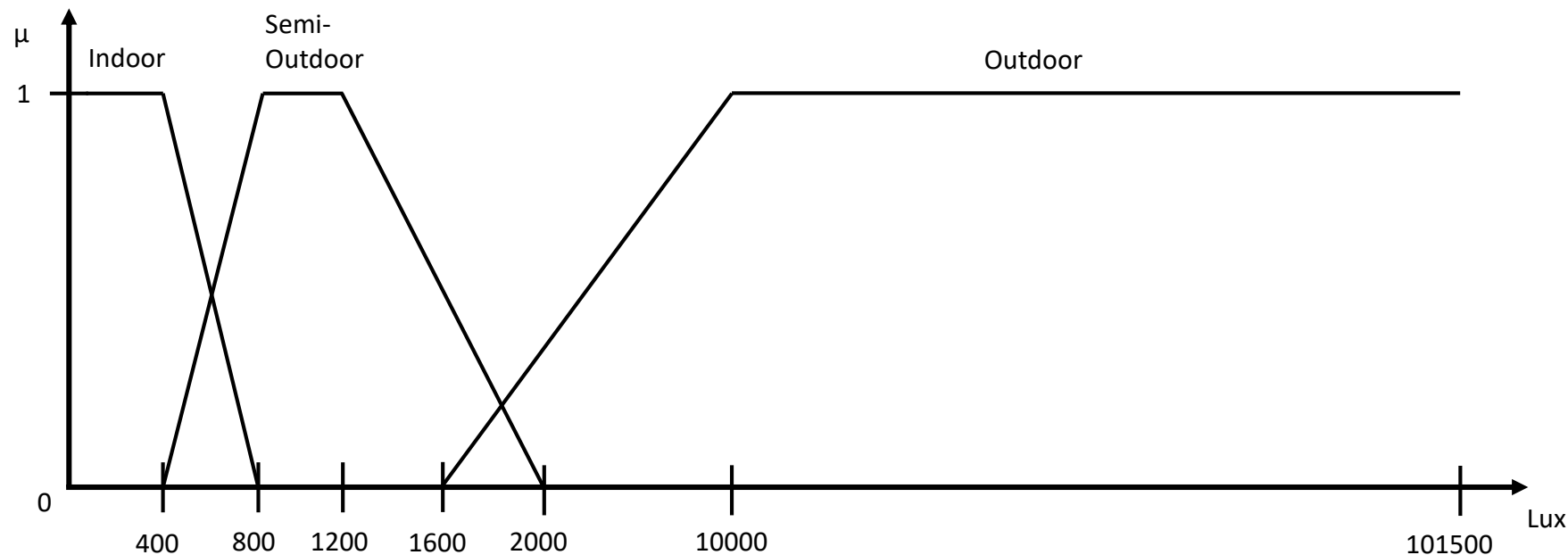


Messung Parkdeck Universität

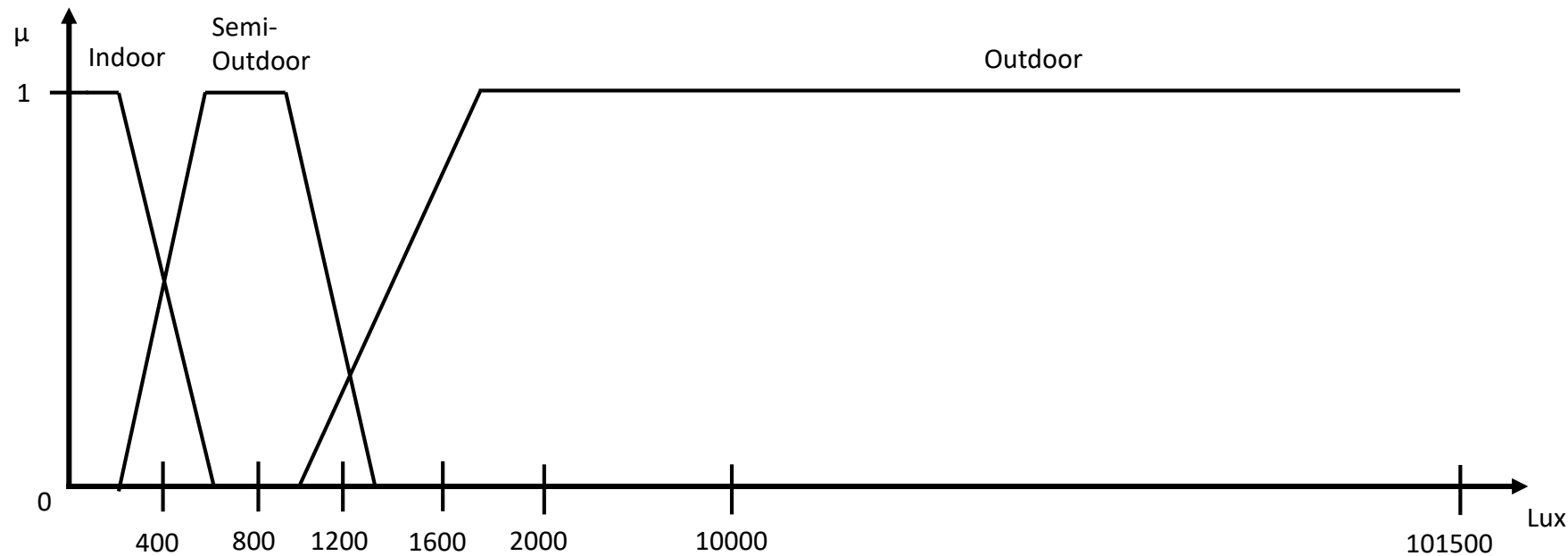


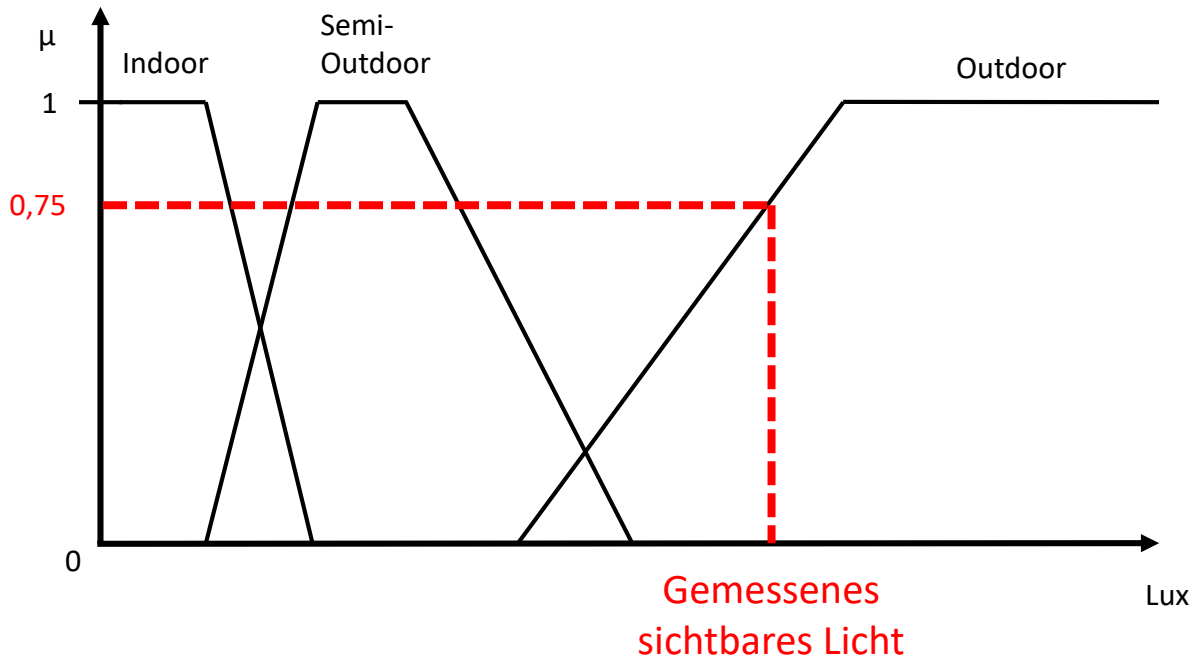
Messung Tiefgarage Universität

- Erstellung von Fuzzy-Mengen aus Logs des Mess-Skripts
  - Eine Fuzzy-Menge für jeden Sensor und Zustand
  - Unterschiedliche Fuzzy-Mengen für jedes Wetterverhältnis
- Beispiel Fuzzy-Mengen bei Sonne:



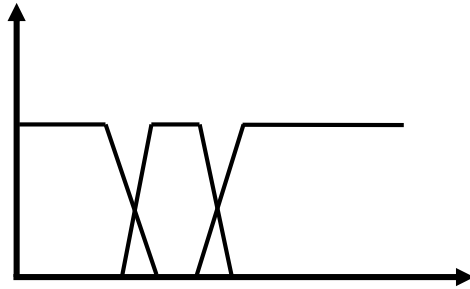
- Erstellung von Fuzzy-Mengen aus Logs des Mess-Skripts
  - Eine Fuzzy-Menge für jeden Sensor und Zustand
  - Unterschiedliche Fuzzy-Mengen für jedes Wetterverhältnis
- Beispiel Fuzzy-Mengen bei Regen:



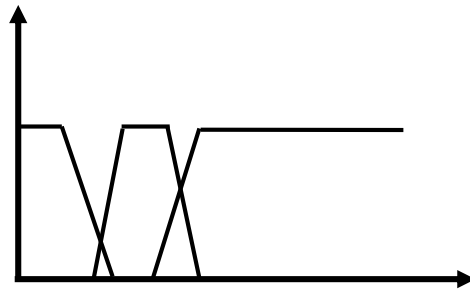


## Zugehörigkeiten

- Outdoor: 0,75
- Semi-Outdoor: 0
- Indoor: 0



- Outdoor: 0,75
- Semi-Outdoor: 0
- Indoor: 0

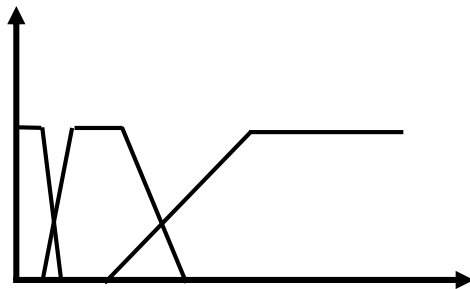


- Outdoor: 0,5
- Semi-Outdoor: 0,5
- Indoor: 0

Gewichtung und  
Akkumulation



- Outdoor: 2,25
- Semi-Outdoor: 1
- Indoor: 0



- Outdoor: 1
- Semi-Outdoor: 0,5
- Indoor: 0

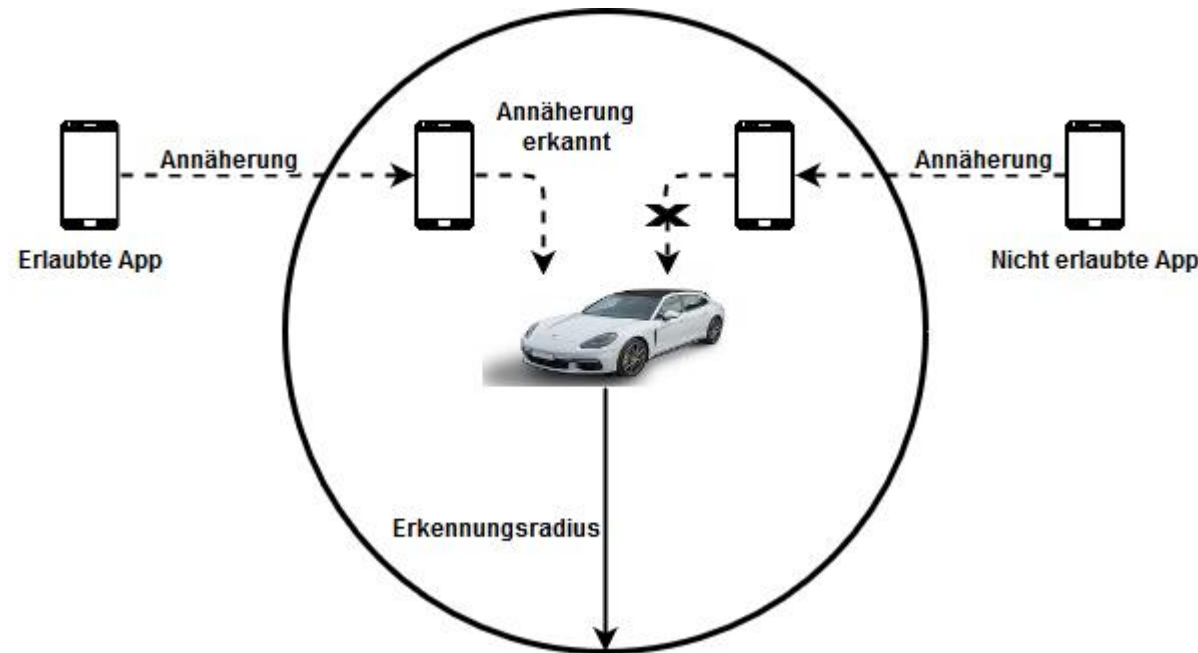
# Annäherungserkennung

Bestimmung, ob sich der Fahrer annähert



- Ziel: Nutzer vor Fahrtantritt einladendes Gefühl geben
- Aktiv während der Vorkonditionierung
- Bei Fahrerannäherung
  - Einschalten des Ambientelichts
  - Entriegeln der Türen
- Bei Fehldetektion Deaktivierung Aktorik
  - Ambientelicht ausschalten
  - Verriegeln der Türen
- Reagiert nur bei Annäherung von erwarteten Apps

- Bereitstellen eines Access Points im Fahrzeug
- Messen der Signalstärke des Access Points mit dem Smart Device
  - Interpretation der Signalstärke als Abstand zum Fahrzeug
- Bei Überschreiten eines definierten Schwellwertes Benachrichtigung des Moduls



# OTA-Updates

Softwareupdates ohne Hardwarezugriff

- Containervirtualisierung
  - Docker
    - Selber Kernel, voneinander abgeschottete Container
    - Ressourcen können Containern zugewiesen werden
    - Programm kann mit Abhängigkeiten gepackt werden
  - Rancher
    - Containerverwaltung
    - Kann auf Hosts Container starten
    - Benötigt Serververbindung zum Container starten
      - Ständige Verbindung kann nicht garantiert werden
      - -> Rancher kann nicht genutzt werden



- Containerverwaltung für IoT-Geräte und verteilte Systeme
- Docker Container werden durch Nebula-Container gestartet
  - Versucht regelmäßig aktualisierte Konfiguration von Server zu beziehen
- Falls Kommunikation mit Server unterbrochen:  
Weiterverwendung der letzten Konfiguration
  - Auch bei Neustart



- ✓ Fertigstellen der Dokumentation
- ✓ Fertigstellung von Architektur und Schnittstellen
- ✓ Verfeinern von Konzepten
  - Safetyanalyse erweitern
  - Erfüllung der Anforderungen
  - Prototyp erstellen
  - Vorbereitung auf Veranstaltungen
  - Ausblick auf Erweiterungen

# Safetyanalyse

- Funktionale Sicherheit mit ISO 26262
- Einschränkung
- Vorgehen
- Gefahren- & Risikoanalyse
- Fehlerbaumanalyse
- Ergebnisse



- Funktionale Sicherheit
  - von E/E-Systemen
  - in Kraftfahrzeugen bis 3,5t
- Minimierung des Risikos gefährlicher Fehlfunktionen
- Vorgaben und Hinweise zu:
  - Safetymanagement
  - Gefährdungsanalyse und Risikoabschätzung
  - Produktentwicklung auf System-, Hardware- und Softwareebene

- Einschränkung des zu betrachtenden Bereiches
- Identifikation von Schadensszenarien
- Einstufung der Schadensszenarien
- Analyse der Schadensszenarien
- Festlegung von Gegenmaßnahmen

- Einschränkung des zu betrachtenden Bereiches
- Identifikation von Schadensszenarien
- Einstufung der Schadensszenarien
- Analyse der Schadensszenarien
- Festlegung von Sicherheitszielen
- Berücksichtigung der Sicherheitsziele

Gefahren- & Risikoanalyse  
und Fehlerbaum-Analyse

- Explizite Betrachtung der uCHILL Software-Komponenten
  - Ausgeschlossen sind:
    - verwendete Hardware
    - Fremdsoftware
- Fahrzeuge die ohne uCHILL sicher sind
- Kein Zugriff auf sicherheitskritische Systeme (Bremsen, Airbag, etc.)
- Fahrzeuge mit:
  - Geschlossener Fahrgastzelle
  - Verbrennungs- und/oder Elektromotor
  - aller benötigter Vorkonditionierungs-Aktorik und –Sensorik
  - Mitteleuropäischer und US-amerikanischer Straßenzulassung

- Von ISO 26262 vorgegeben
- Zu betrachtende Faktoren:
  - Severity (Schweregrad der Auswirkung)
  - Exposure (Häufigkeit)
    - Zeit der Ausgesetztzeit
    - Auftrittswahrscheinlichkeit
  - Controllability (Beherrschbarkeit)
- Aus Faktoren wird das ASIL (Automotive Safety Integrity Level) ermittelt
  - Essentiell für Spezifikation von Sicherheitsanforderungen & Umsetzung von Sicherheitszielen in der Konzeption

# Bestimmung des ASILs



- **S0:** keine Verletzungen
- **S1:** leichte Verletzungen
- **S2:** Schwere bis lebensgefährliche Verletzungen
- **S3:** fatale lebensgefährliche Verletzungen
- **E0:** unwahrscheinliches Auftreten
- **E1:** sehr geringes Auftreten
- **E2:** geringes Auftreten
- **E3:** durchschnittliches Auftreten
- **E4:** wahrscheinliches Auftreten
- **C0:** Generell kontrollierbar
- **C1:** einfach kontrollierbar
- **C2:** recht kontrollierbar
- **C3:** schwer zu kontrollieren

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

- QM
  - Keine weiteren Sicherheitsmaßnahmen der ISO 26262 erforderlich für betrachtete Funktion
  - Erfolgreiches Umsetzen eines Qualitätsmanagement nach ISO 9001 ausreichend
- ASIL A-D
  - Weitere Sicherheitsmaßnahmen der ISO 26262 erforderlich

- Fehlfunktion:  
Fehlerhafte Steuerung von Aktorik während der Fahrt
- Fehlerauswirkung:  
Ablenkung des Fahrers
- Betrachtetes Szenario:  
CHILL-Modul verändert plötzlich das Ambientelicht und stellt die maximale Intensität im Stadtverkehr ein
- Bemerkung zum Schweregrad:  
Die plötzliche Ablenkung begünstigt, dass der Nutzer in eine Kollision bei mittlerer Geschwindigkeit gelangt.
- Severity: S3
- Exposure: E3
- Controllability: C0
- ASIL: QM
- Sicherer Zustand:  
CHILL-Modul ist während der Fahrt inaktiv
- Sicherheitsziel:  
CHILL-Modul steuert keine Aktorik während der Fahrt

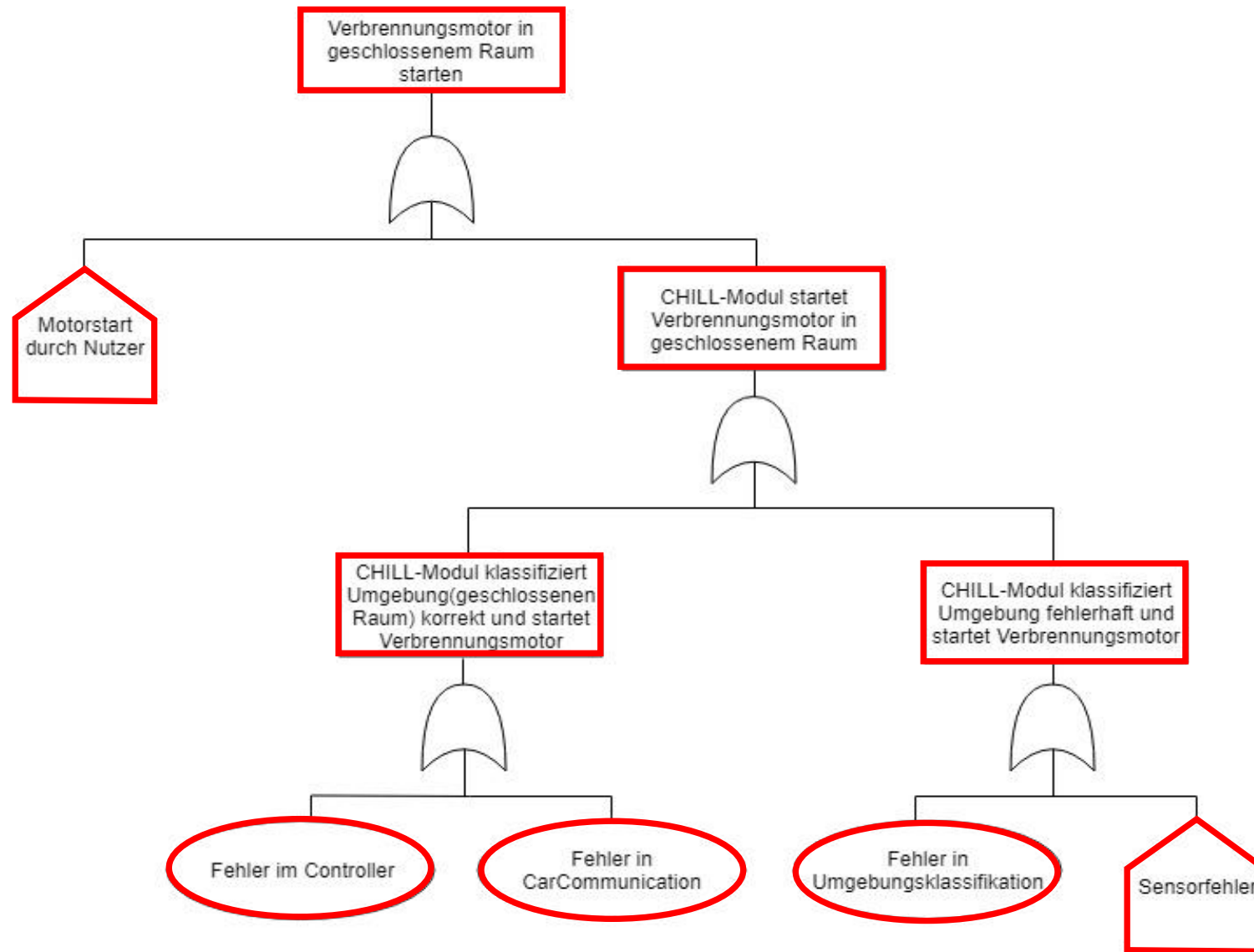


- Fehlfunktion:  
Andauernde fehlerhafte Steuerung von Aktorik während der Fahrt
- Fehlerauswirkung:  
Keine manuelle Einstellung der Klimatisierungs-Aktorik möglich
- Betrachtetes Szenario:  
Nachtfahrt bei hoher Luftfeuchtigkeit, sodass die Scheiben während einer Landstraßen-Fahrt beschlagen. Daraufhin verliert der Fahrer die Sicht
- Bemerkung zum Schweregrad:  
Unfall auf Landstraße bei hoher Geschwindigkeit
- Severity: S3
- Exposure: E4
- Controllability: C1
- ASIL: B
- Sicherer Zustand:  
CHILL-Modul ist während der Fahrt inaktiv
- Sicherheitsziel:  
CHILL-Modul steuert keine Aktorik während der Fahrt

- Fehlfunktion:  
Starten des Motors im geschlossenen Raum
- Fehlerauswirkung:  
Laufender Verbrennungsmotor im geschlossenen Raum
- Betrachtetes Szenario:  
CHILL-Modul startet  
Verbrennungsmotor im geschlossenen Raum, sodass eine Abgasvergiftung begünstigt wird
- Bemerkung zum Schweregrad:  
Mögliche Abgasvergiftung
- Severity: S3
- Exposure: E4
- Controllability: C1
- ASIL: B
- Sicherer Zustand:  
CHILL-Modul ist ausgeschaltet
- Sicherheitsziel:  
uCHILL klassifiziert die Umgebung nicht fälschlicherweise als Outdoor

- 15-betrachtete Szenarien
  - 2 ASIL B Funktionen
  - 13 QM Funktionen
- Andauernde fehlerhafte Steuerung der Klimatisierungs-Aktorik (ASIL B)
  - Ist in Sicherheitsanforderungen als auch in der Implementierung berücksichtigt
- Start des Verbrennungsmotors im geschlossenen Raum (ASIL B)
  - Ist in Sicherheitsanforderungen als auch in der Implementierung berücksichtigt

- Systemanalyse zur Betrachtung der:
  - Zuverlässigkeit des Systems
  - Ausfall-Wahrscheinlichkeit eines Sub- und des Gesamtsystems
- Dies geschieht über:
  - Logische Verknüpfungen von Teilsystemen
  - Betrachtung der kritischen Pfade
  - Folglich derer, die zu einem Ausfall des Gesamtsystems führen

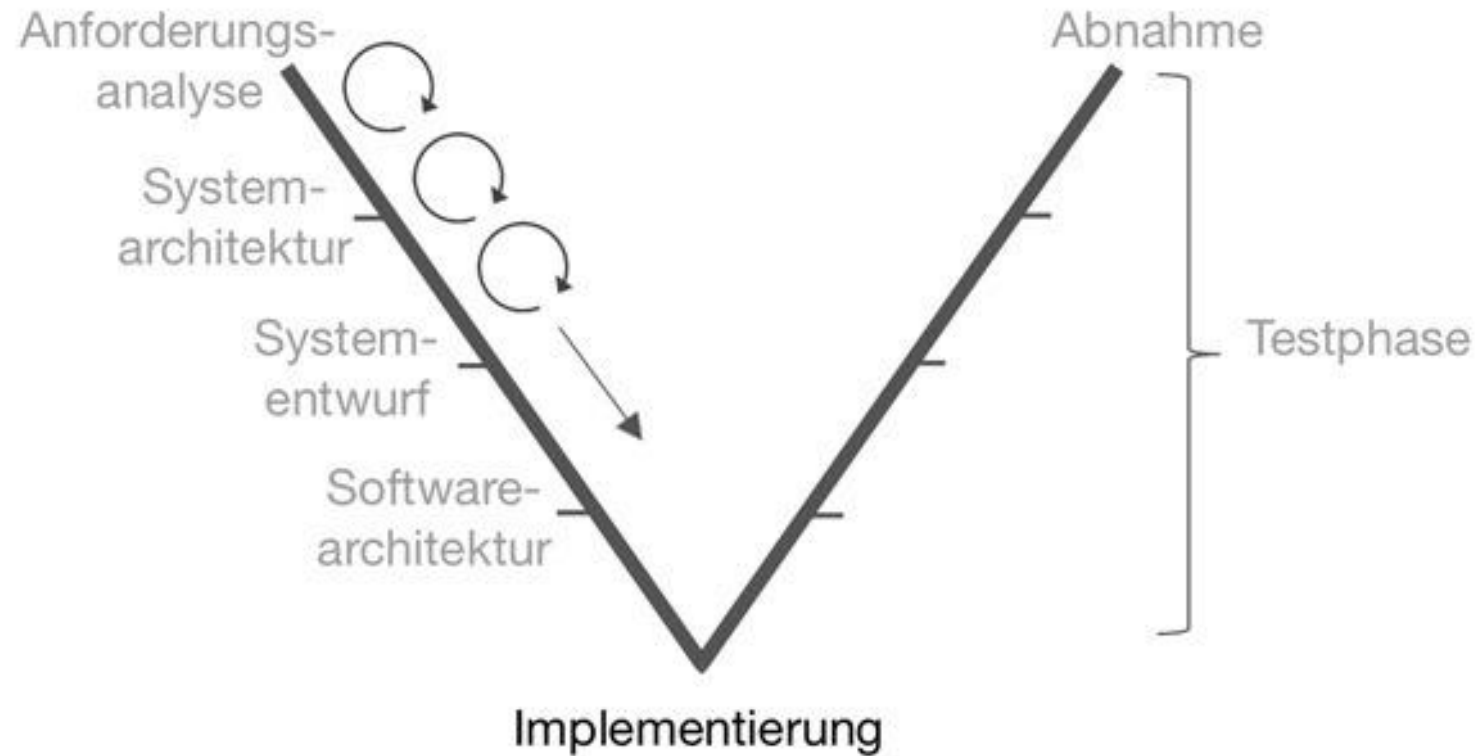


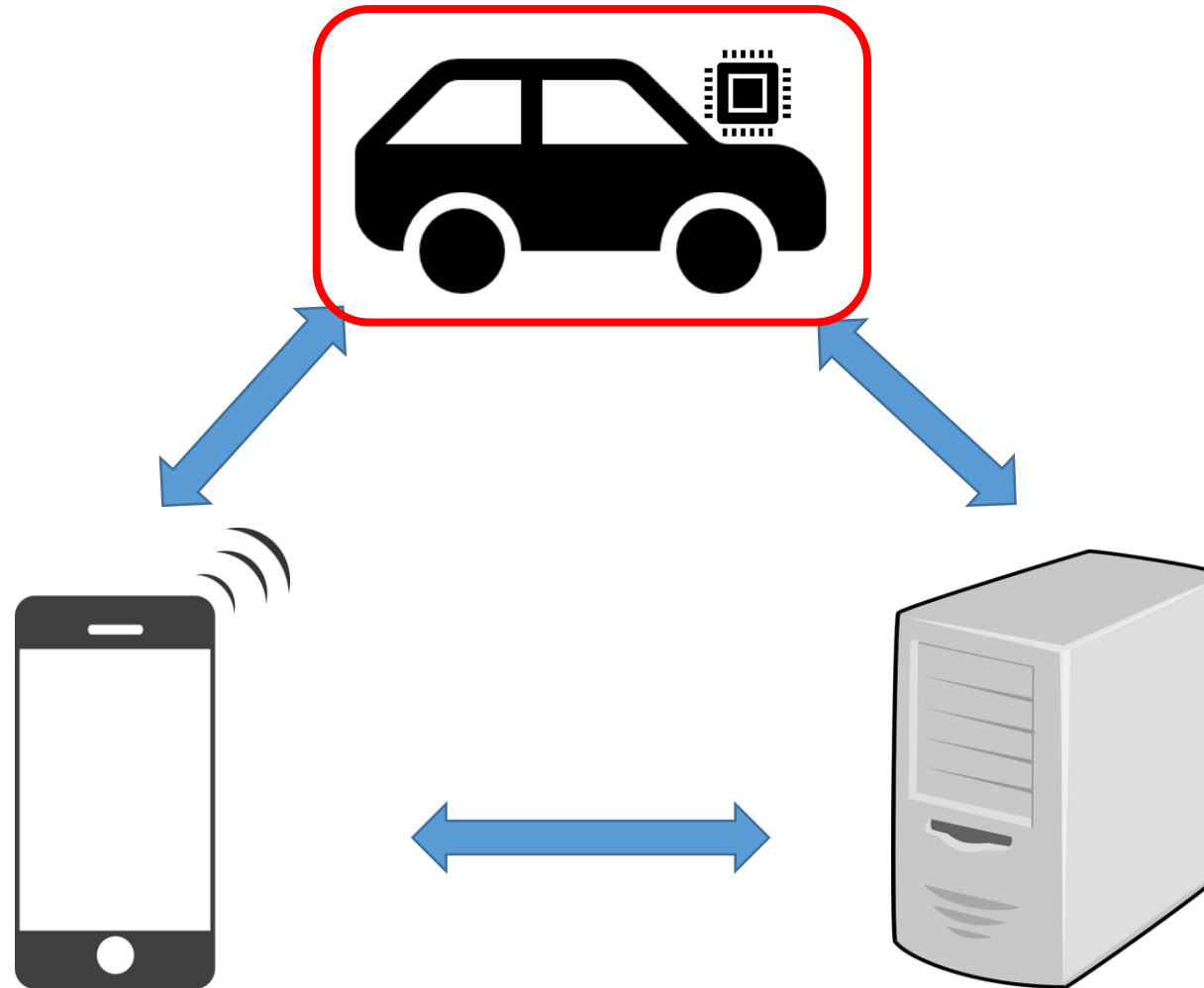
- Für viele Funktionen reicht die Umsetzung von einem Qualitätsmanagement
- Zwei ASIL B Funktionen betrachtet
- Kritische Pfade von ASIL B Funktionen mit Fehlerbaumanalyse erarbeitet
- ASIL B Funktionen werden in Konzeption behandelt
- Fehlerhafte Anwendung von uCHILL:  
Es gibt keine Möglichkeit für den Nutzer in eine gefährliche Situation zu gelangen, falls das System mutwillig falsch verwendet wird.
- Da nicht nach ISO 26262 entwickelt wurde, folgt im weiteren Verlauf die Erläuterungen zum Abfangen der kritischen Pfade

- ✓ Fertigstellen der Dokumentation
- ✓ Fertigstellen von Architektur und Schnittstellen
- ✓ Verfeinern von Konzepten
- ✓ Safetyanalyse erweitern
  - Erfüllung der Anforderungen
  - Prototyp erstellen
  - Vorbereitung auf Veranstaltungen
  - Ausblick auf Erweiterungen

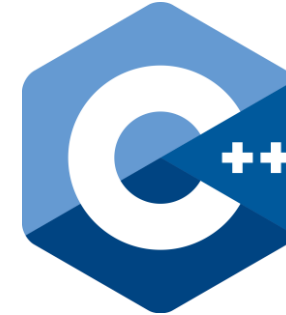
# Implementierung







- Programmiersprache



- Build-Tool

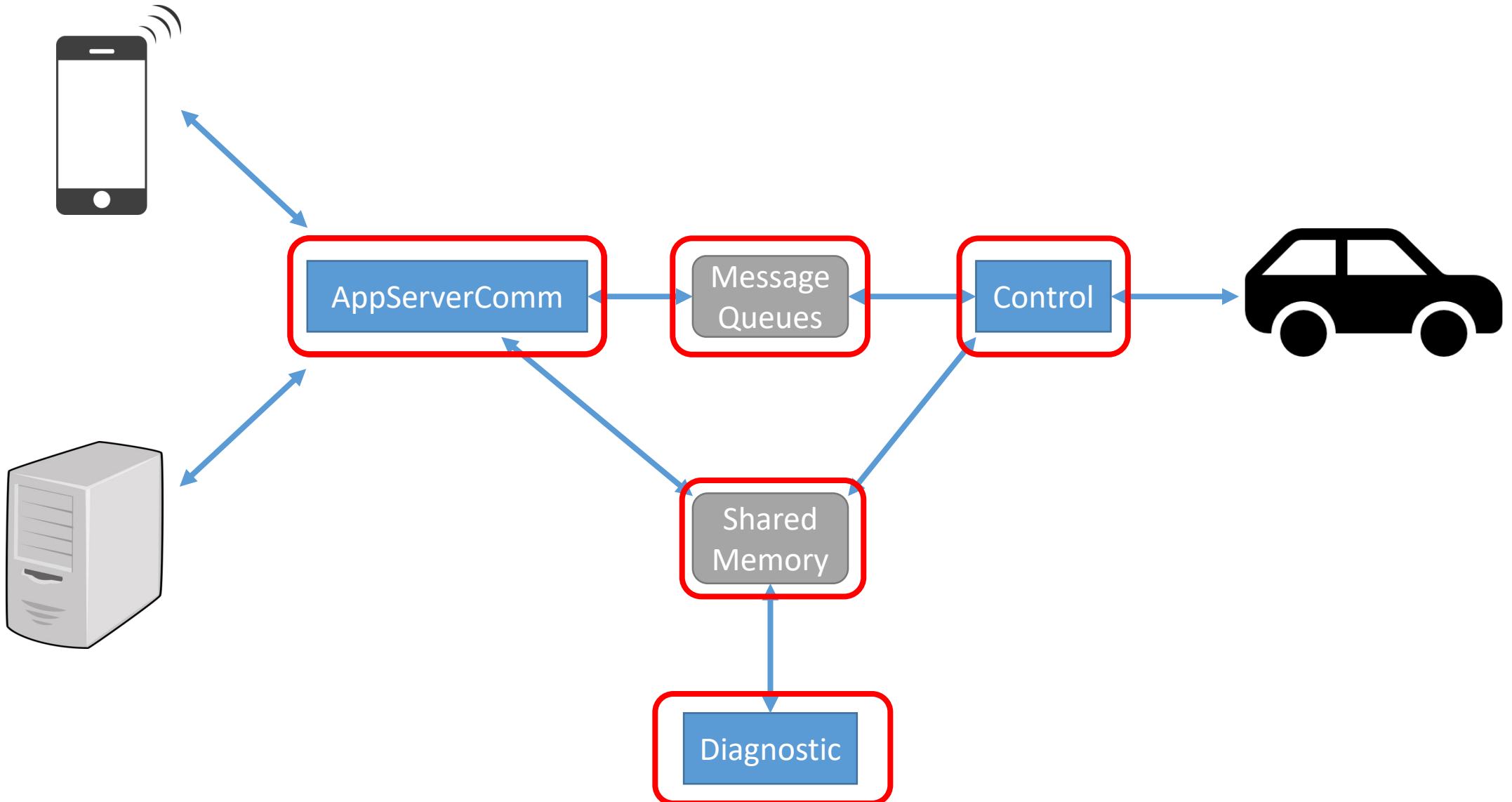
- Plattformunabhängig



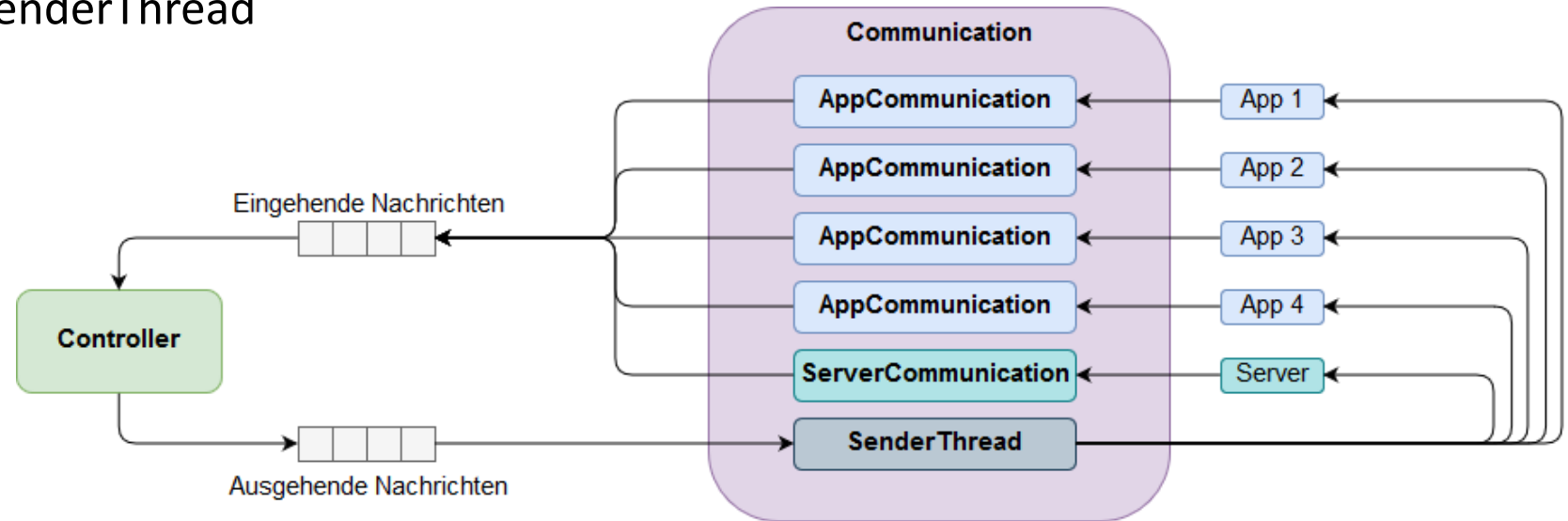
- Libraries

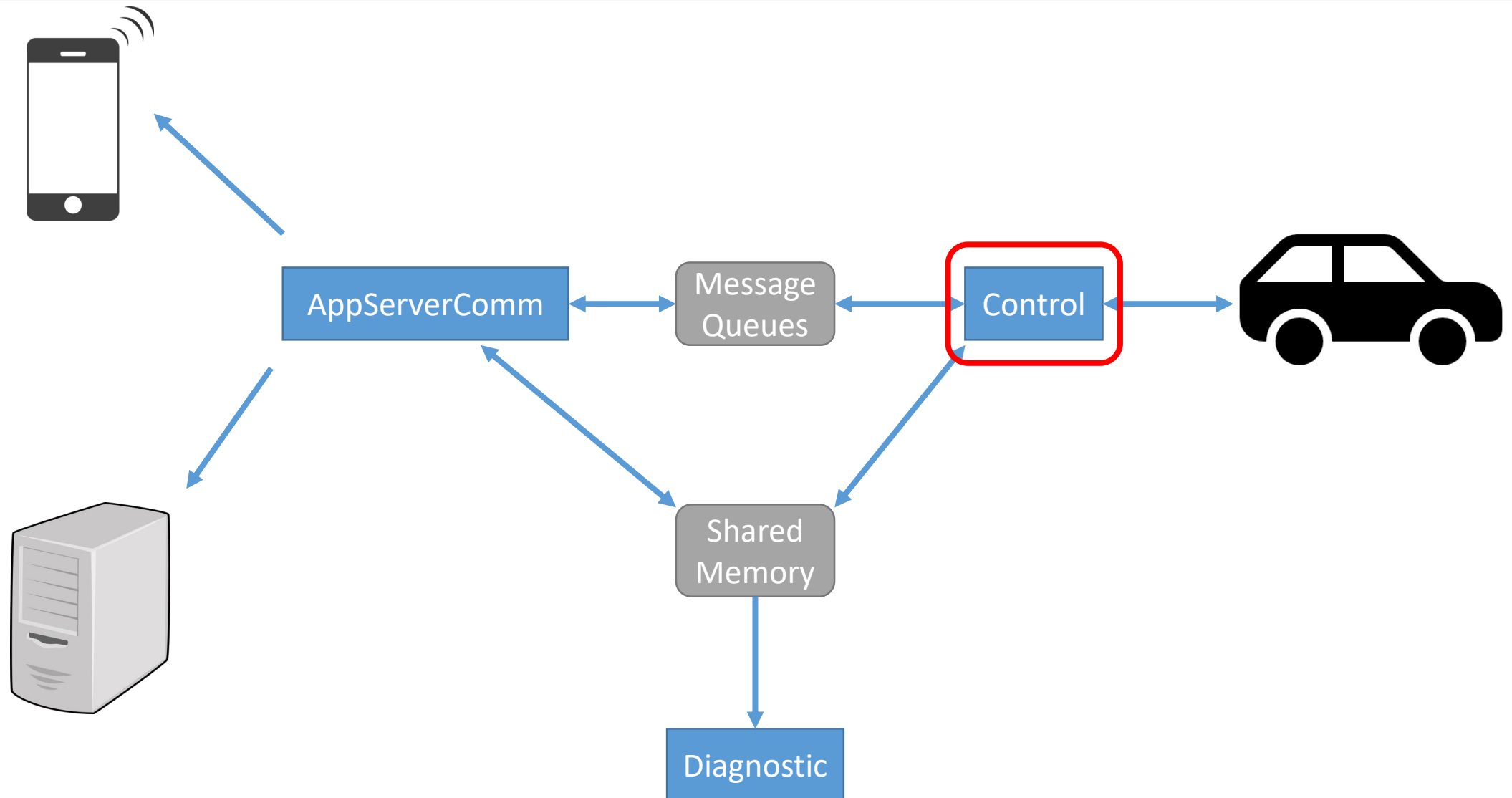
- Shared Memory
- Asio (TCP)
- Beast (HTTP)
- Ptree (JSON)

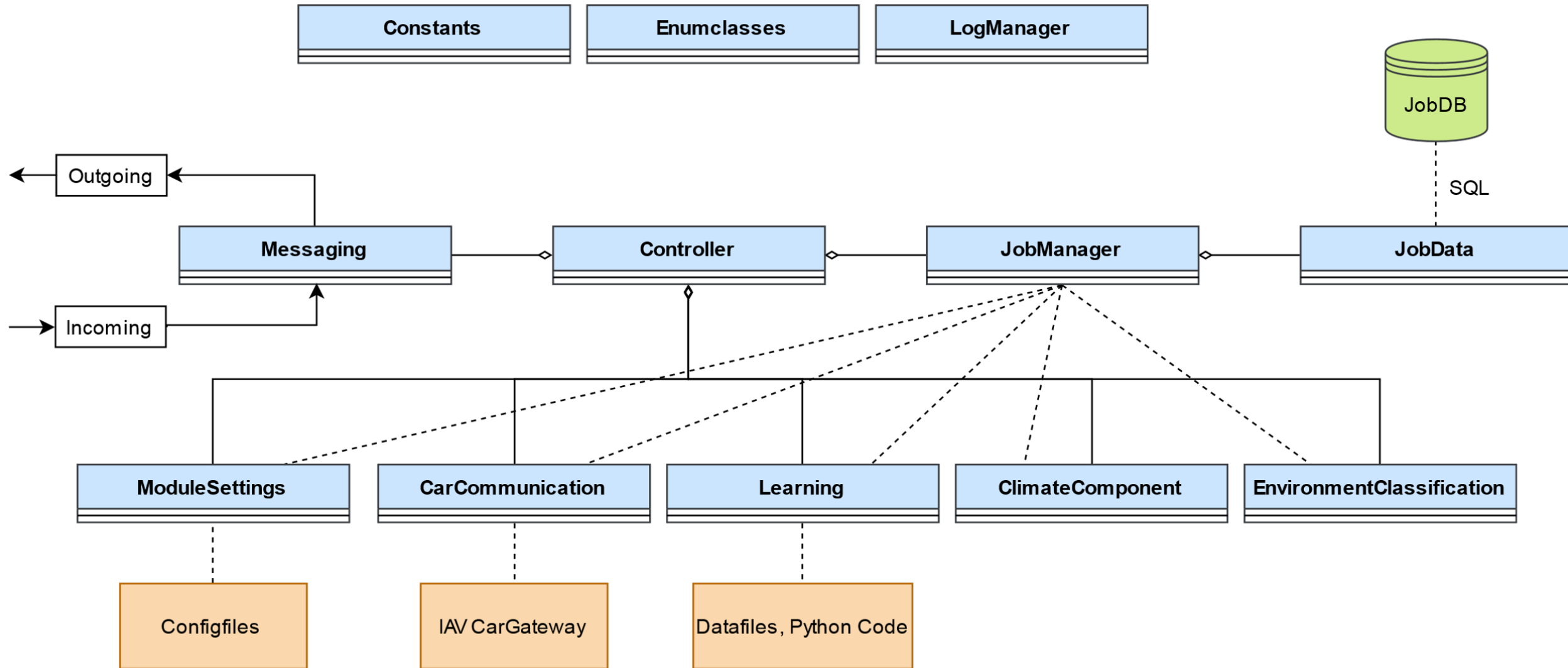


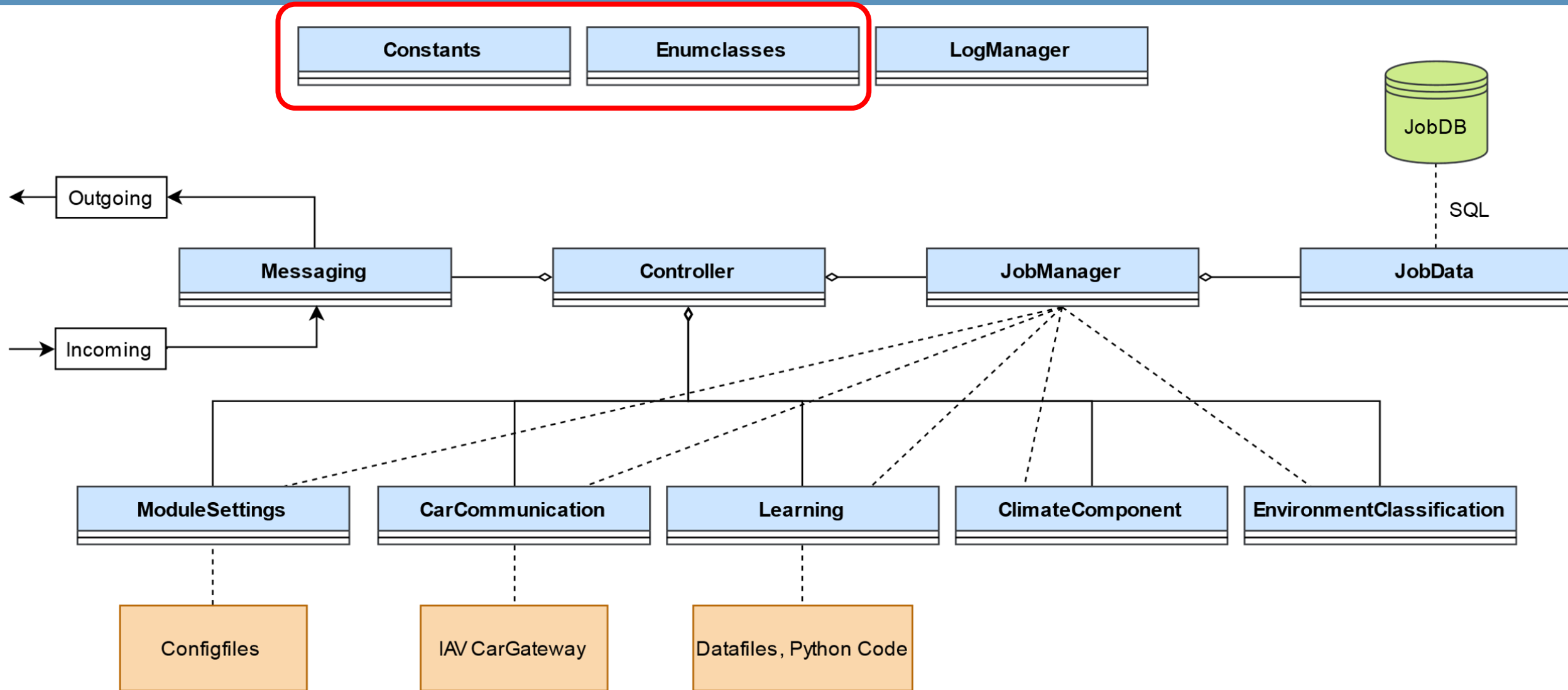


- Interprozesskommunikation via Boost interprocess Message\_Queues
- Asynchrones Empfangen von Nachrichten mittels Boost Asio Sockets
- Senden von Nachrichten im SenderThread mittels Boost Asio Sockets

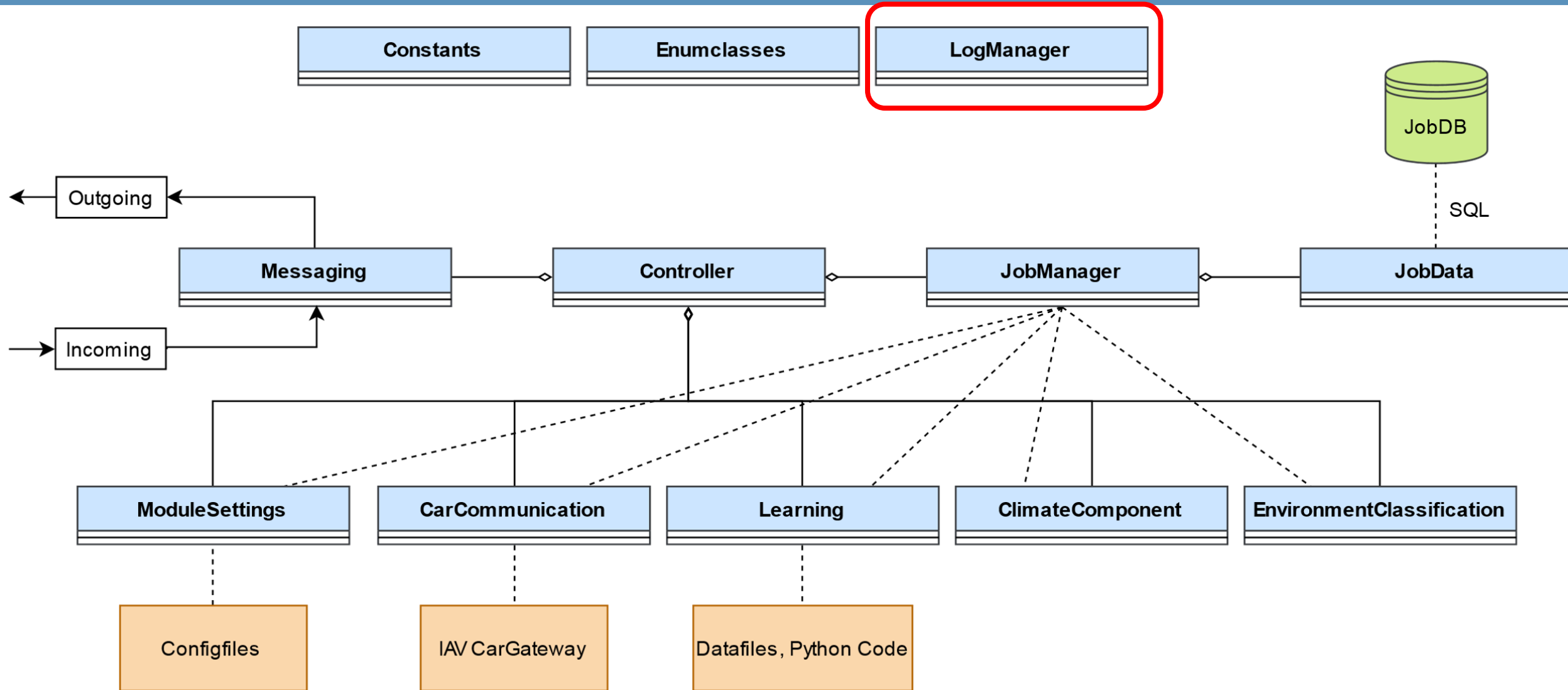








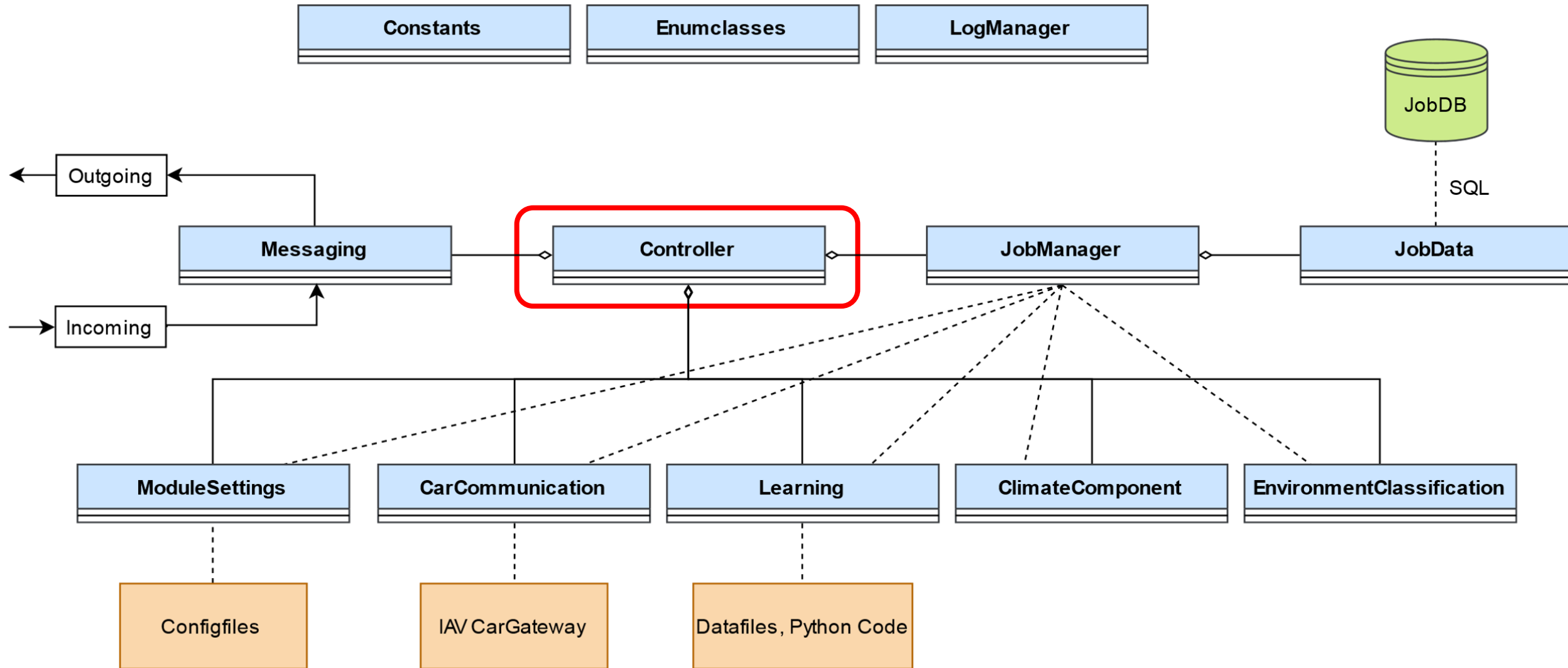




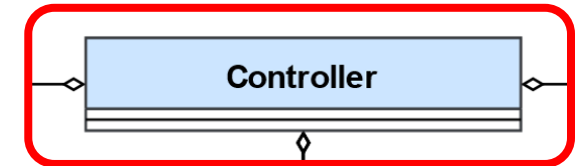
- Erstellung von Lognachrichten
- Zugriff über statische Methoden, Singleton
- Wrapper um Boost Log
- Einstellung minimaler Loglevels, Dateigrößen, Ordnergrößen



Methode / Level	Nutzung
LOG_DEBUG	Debugnachrichten
LOG_INFO	Informationen über interne Einstellungen
LOG_WARNING	Warnungen
LOG_ERROR	Fehler (Daten fehlerhaft etc.)
LOG_FATAL	Fehler mit Beendigung des Prozesses
LOG_CONDITIONING	Einstellungen am Fahrzeug

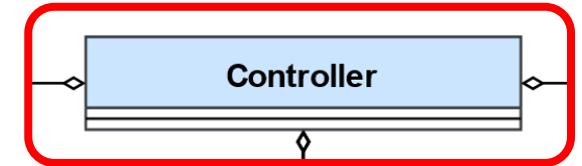


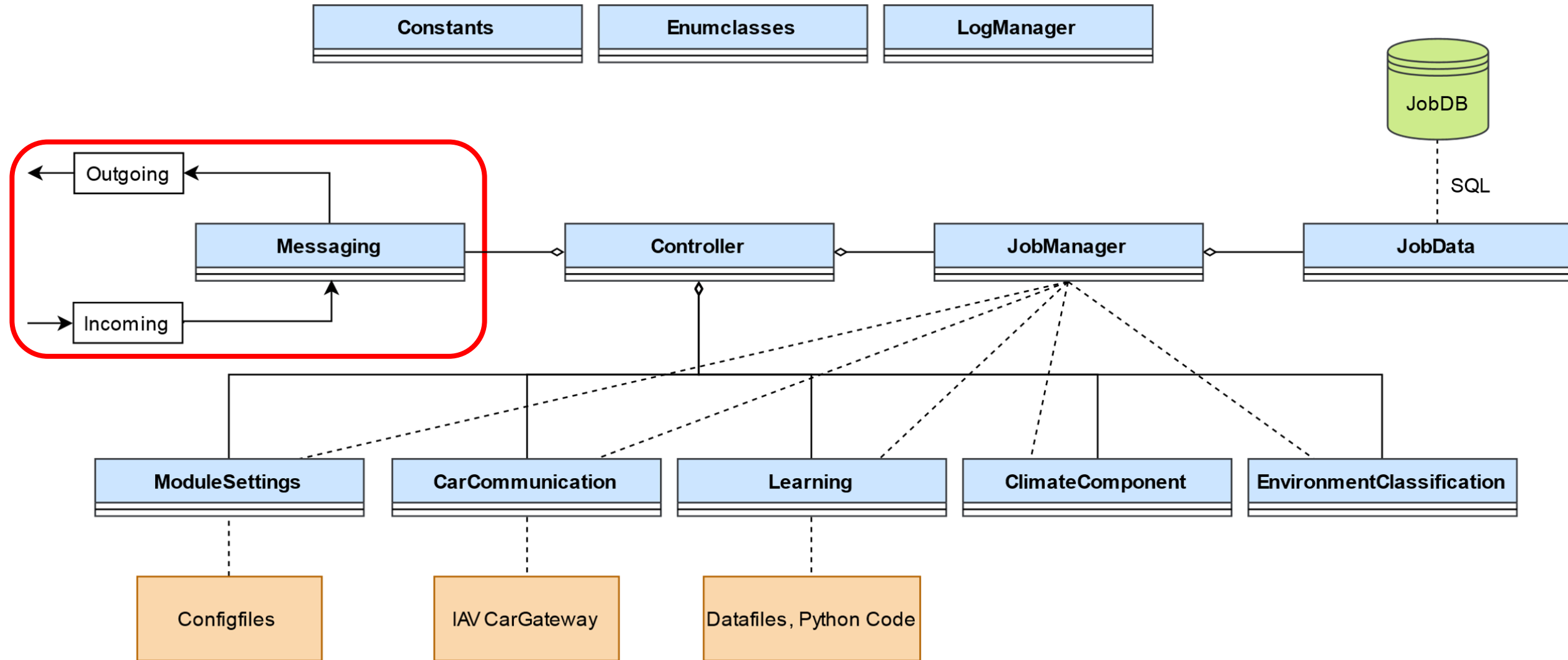
- Zentrale Steuerung des CHILL-Moduls
- Verwaltung von Einstellungen und Betriebsmodi
- Initialisierung und Steuerung von Unterkomponenten

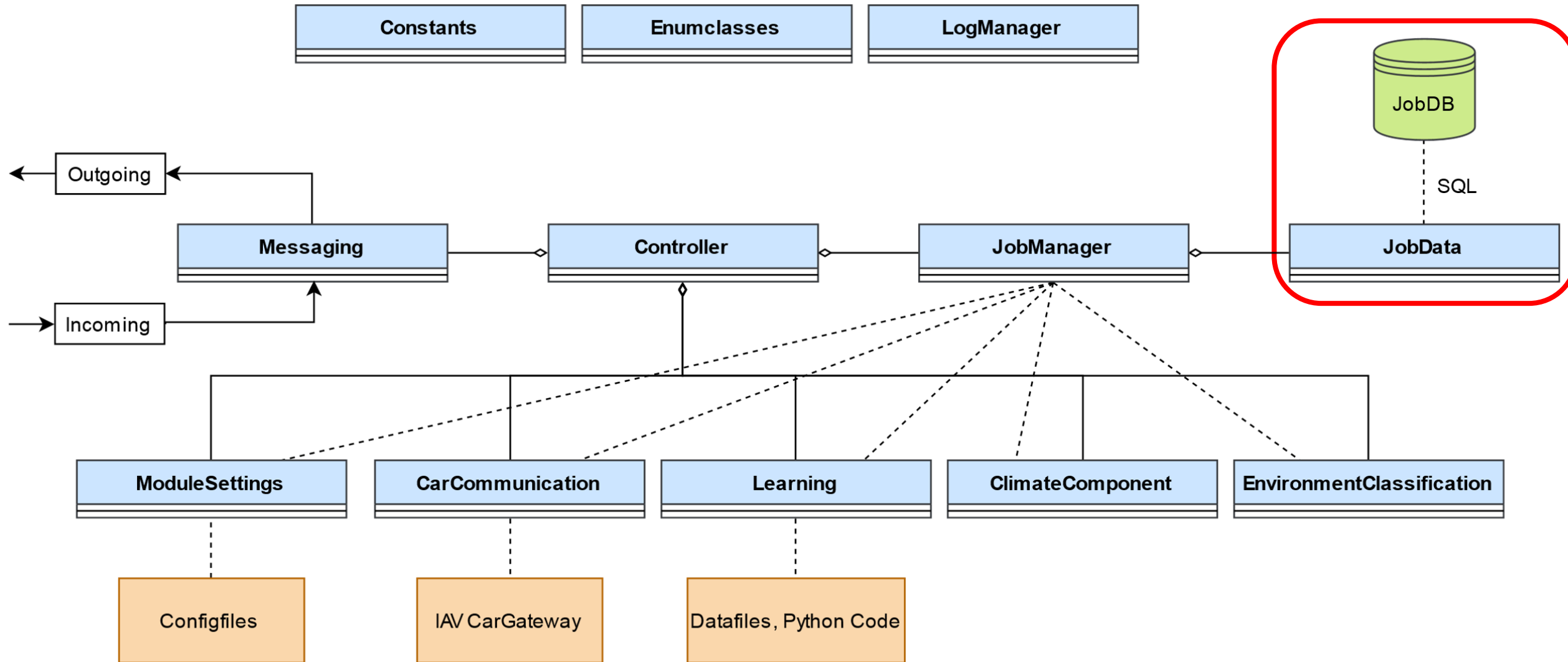


Modus	Funktion
ACTIVE_UNKNOWN	Initialzustand
NOT_DRIVING	Fahrzeug wird aktuell nicht benutzt, Konditionierung möglich
DRIVING	Fahrzeug fährt, nur Lesen von Nutzerverhalten
VACATION	Urlaubsmodus, System inaktiv

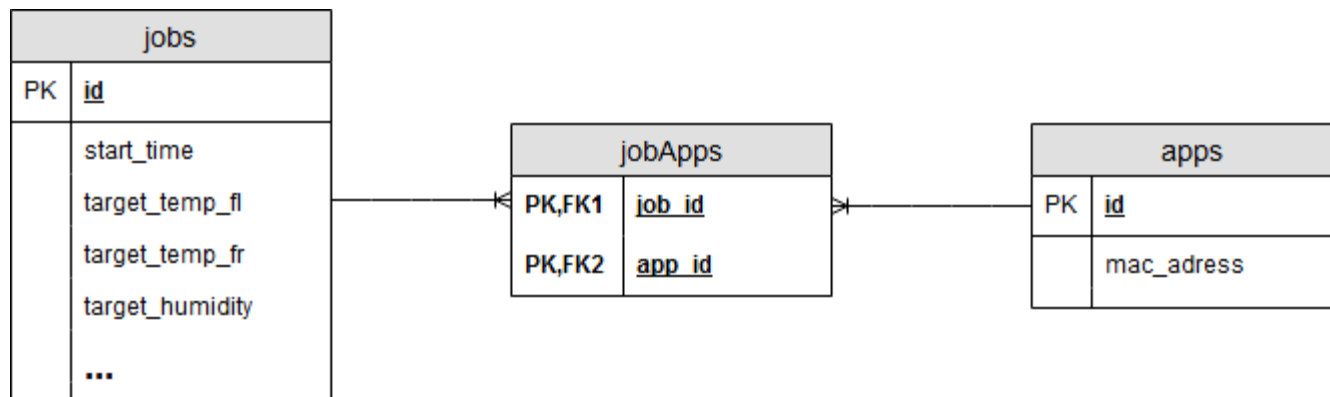
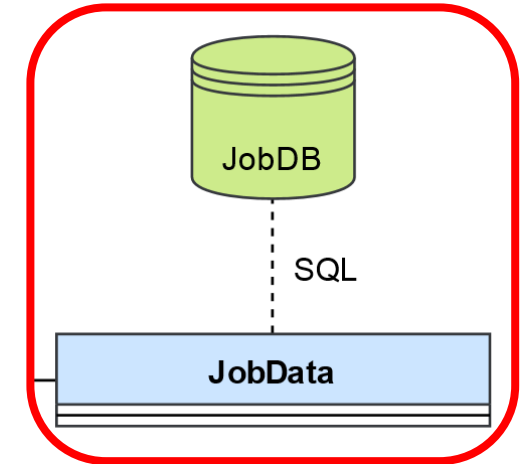
- Run Loop:
  - Prüfung eingehender Nachrichten
  - Update des Modus
  - Modeabhängiges Vorgehen
    - ACTIVE\_UNKNOWN und VACATION: sleep
    - DRIVING: Sammeln von Sensorwerten zum Lernen
    - NOT\_DRIVING: Ermittlung und Ausführung von Konditionierungen (via JobManager)
- Erkennung des Fahrtzustandes
  - Aktuelle Geschwindigkeit
  - Aktuelle Motordrehzahl
  - Änderung der GPS-Position
  - Änderung zu NOT\_DRIVING nicht sofort



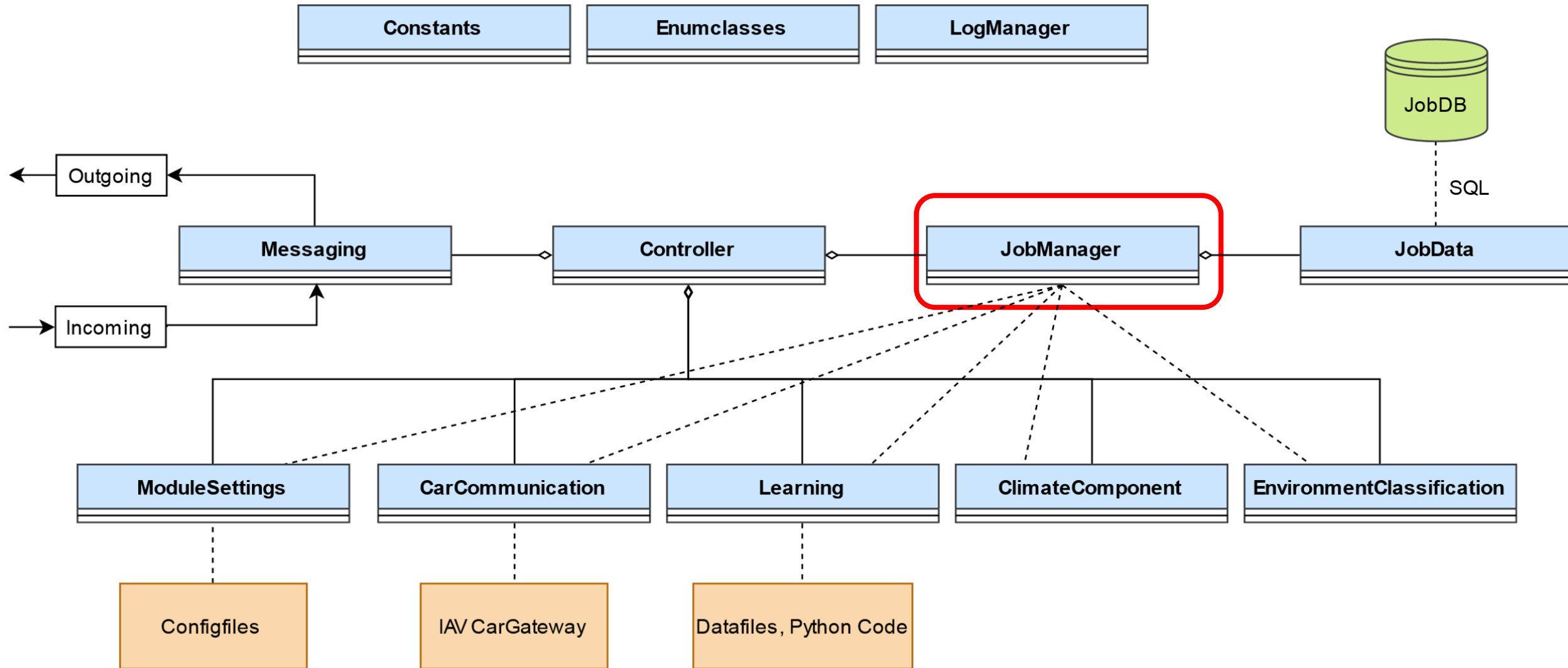




- Abstraktion der Auftragsdaten für JobManager
- SQLite mit SQLiteCPP
- Speichern und Laden von Aufträgen und Apps
- Vergabe von Job-Ids
- Prüfung, ob notwendige Felder gefüllt sind







- Verwaltung von Konditionierungsaufträgen
- Nutzung von JobData Funktionen
- Bereitstellung von Funktionen für
  - Erkennung ob ein Auftrag in Kürze ansteht
  - Vorbereitung neuer eingehender Aufträge
  - Verschiedene Statusinformationen zum Ablauf
  - Durchführung eines Auftrages



- runJob() Ablauf

- Prüfung, ob bereits ein Auftrag aktiv ist
- Sammeln der notwendigen Auftragsdaten
  - Jobdatenbank
  - Lernkomponente
- Bestimmung der erlaubten Aktorik
  - Umgebungserkennung
  - Wetterbestimmung
  - Einstellungen
- Klimatisierungseinstellungen aus Klimakomponente
- Start der Konditionierungsschleife



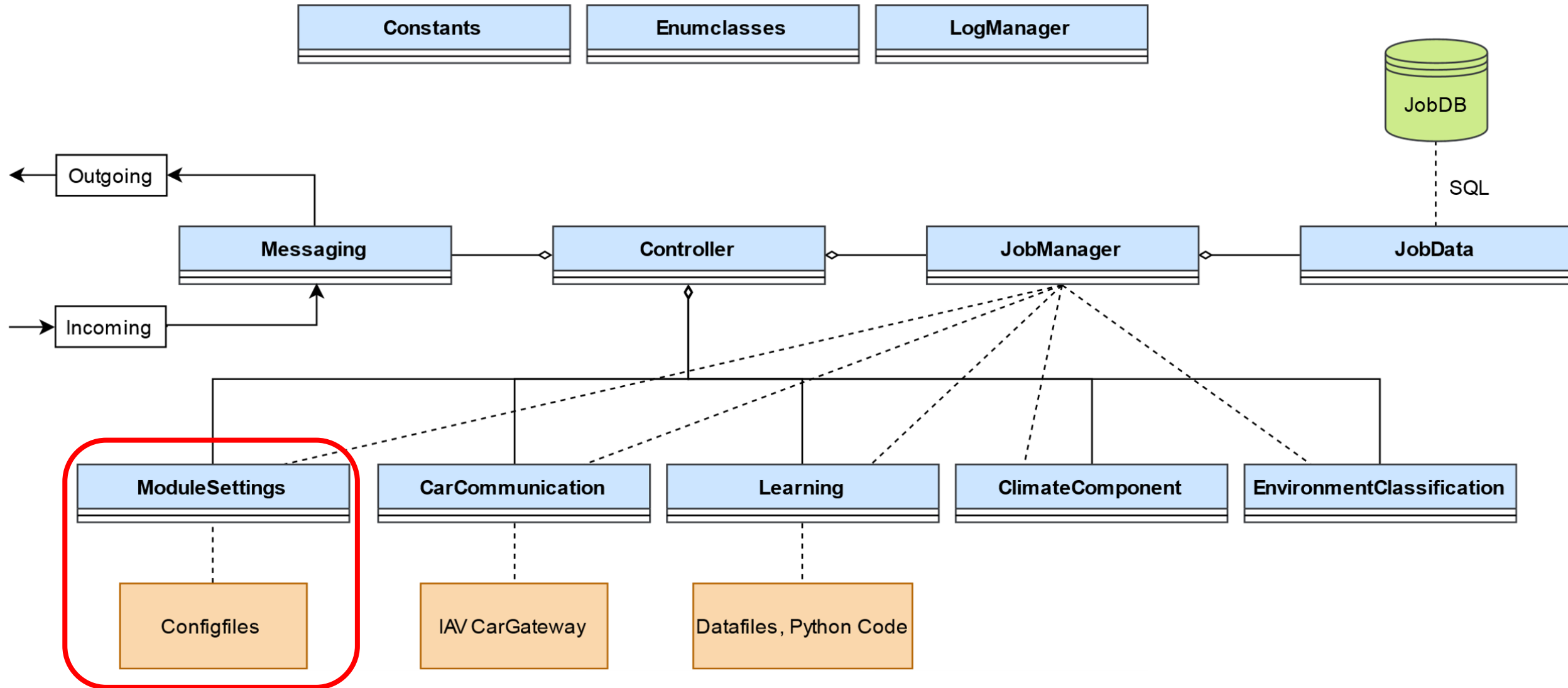
- Konditionierungsschleife

- Prüfung und Einstellen des nächsten Klimawertes
- Start der Annäherungserkennung, wenn Zeitfenster um Zielzeitpunkt erreicht ist
- Einstellung der finalen Aktorik, wenn Annäherung erkannt wird
- Beenden des Auftrages, wenn finale Aktorik gesetzt ist und Nutzer die Tür öffnet

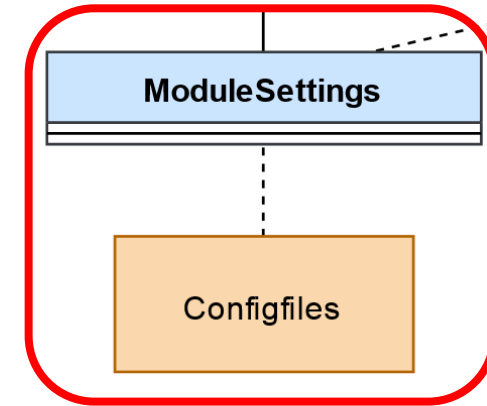


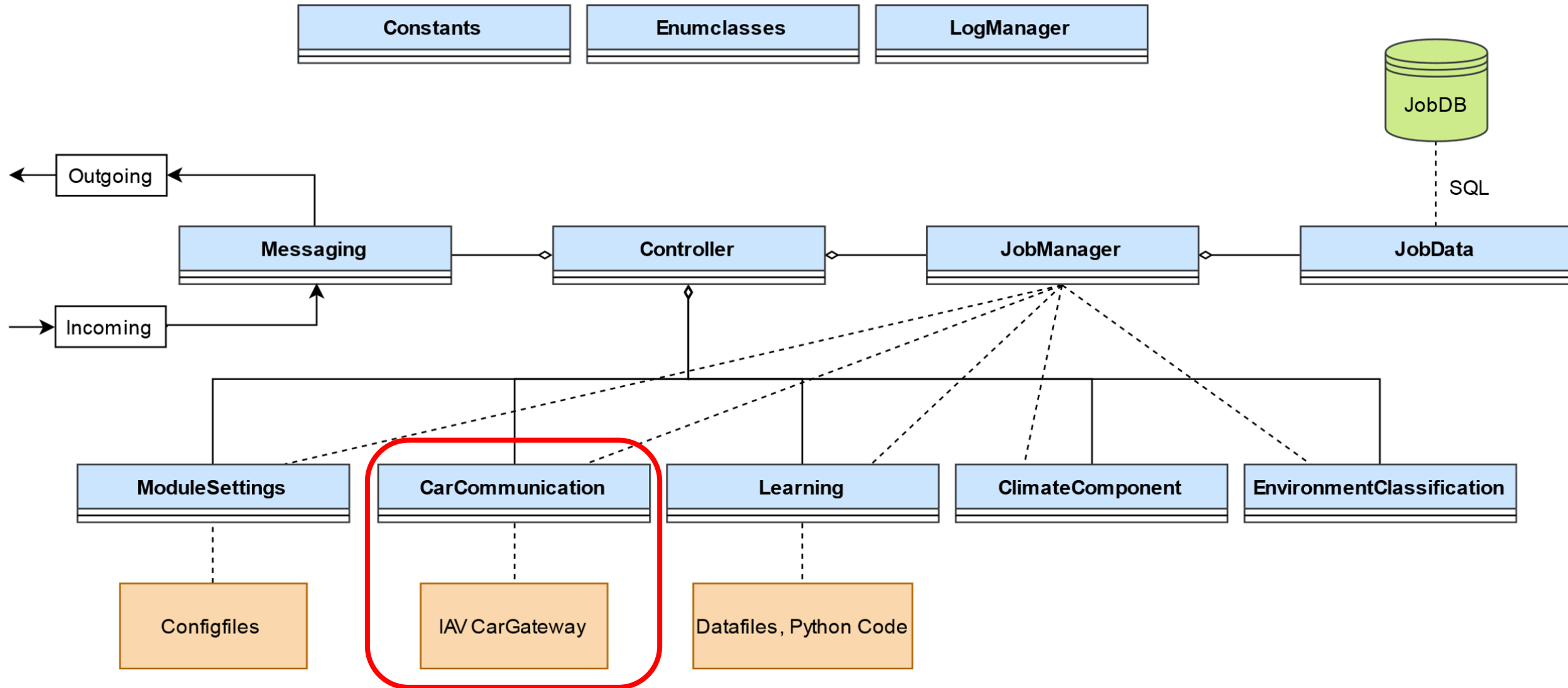
- Bei Fehlfunktionen oder unerwartetem Verhalten

- Einstellung eines sicheren Zustandes im Fahrzeug
- Beenden des Auftrages

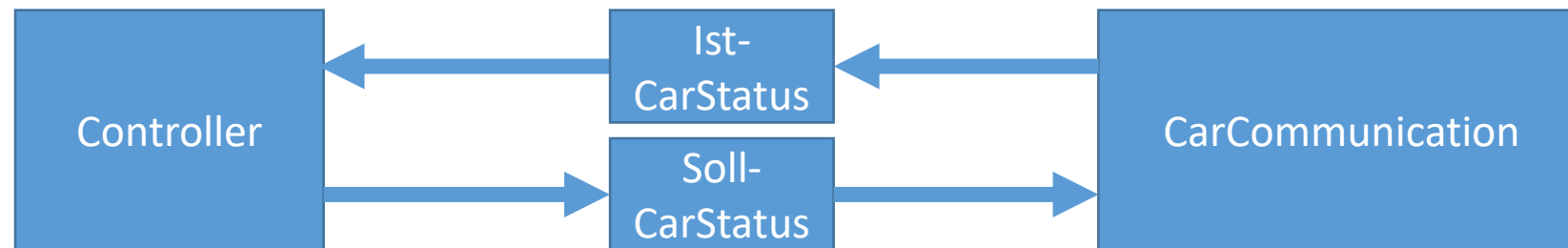
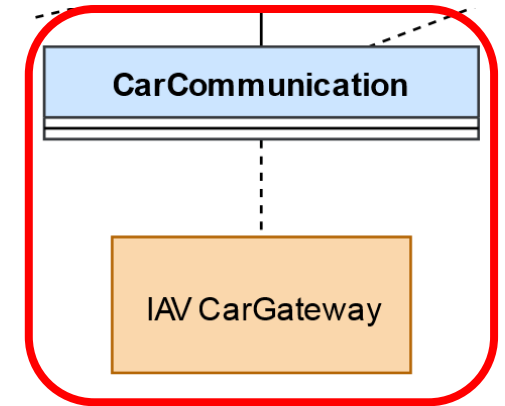


- Verwalten von Konfigurationsdateien
  - Stellt `getValue()` und `setValue()` bereit
- Konfigurationsdateien in JSON
  - Laden und Speichern durch Boost Property Trees
- Unterscheidung änderbaren und festen Einstellungen
  - Module-ID fest
  - Urlaubsmodus änderbar
- Definition von sicheren Default-Werten





- Kommunikation mit IAV CarGateway
- Basiert auf Boost Asio (TCP) und Boost Beast (HTTP)
- Schnittstelle von IAV ist in JS implementiert
  - Einarbeitungszeit nötig
- Informationsaustausch per ist- und soll-CarStatus-Objekte
- CarStatus: Werte aller relevanten Sensoren und Aktoren





- Anfrage an das Gateway:

/rpc/com.iav.iot.test.getWindowGroup {}

HTTP-Request-Target

HTTP-Request-Body

- Antwort:

Fensterheber Beifahrerseite

Fehlernummer

{"result":{"UCU\_FHFS":3,"UCU\_FHBFS":3, ... "UCU\_Instance":2,"UCU\_Error":0}}

Fensterheber Fahrerseite

Instanznummer

- Anfrage an das Gateway:

/rpc/com.iav.iot.test.setWindowGroup {"args":{"windowFrontLeft":3,...}}

HTTP-Request-Target

HTTP-Request-Body

- Antwort:

Fensterheber Beifahrerseite

{"data":{"windowFrontLeft":3,"windowFrontRight":3,..."error":0}}

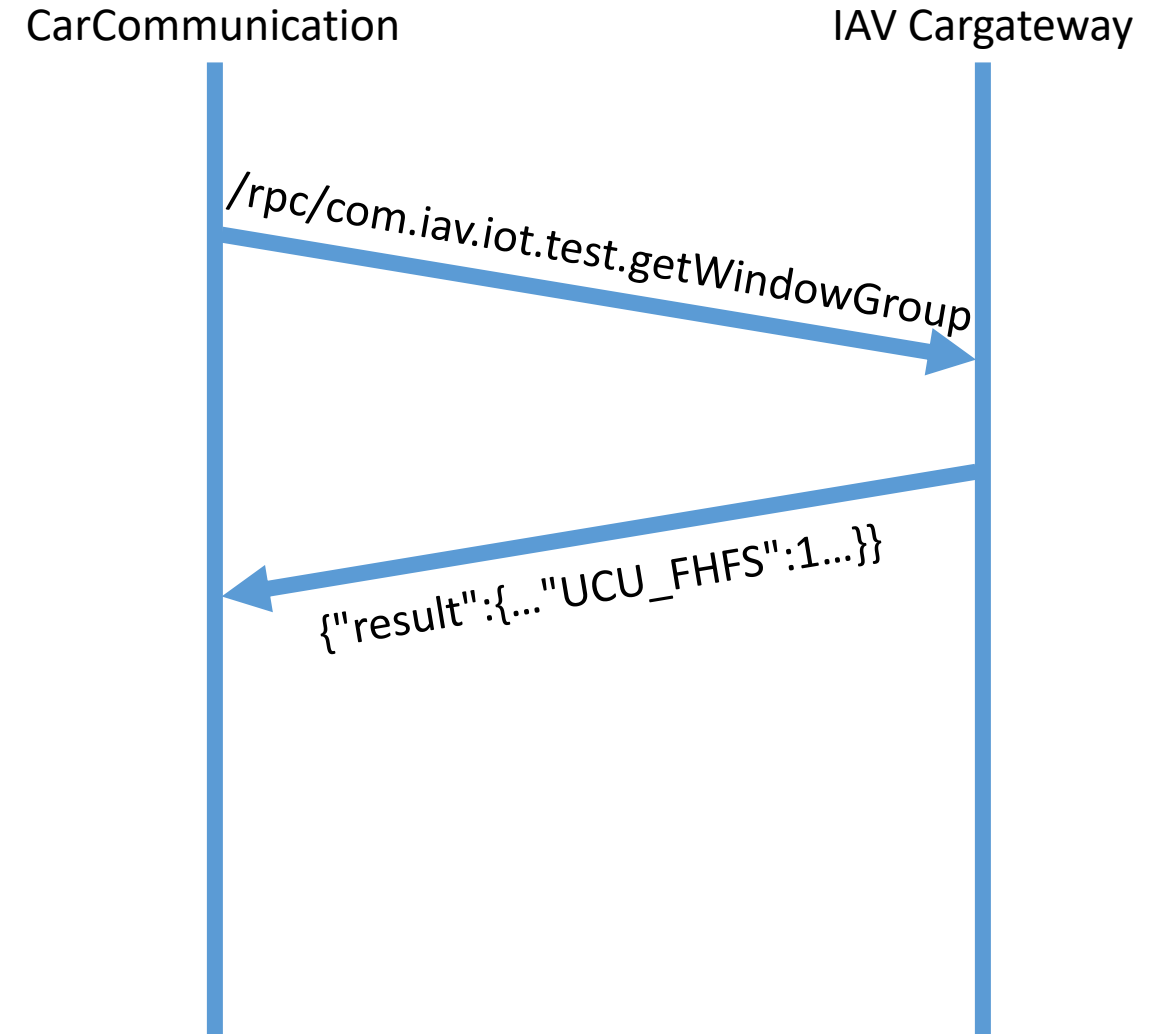
Fensterheber Fahrerseite

Fehlernummer

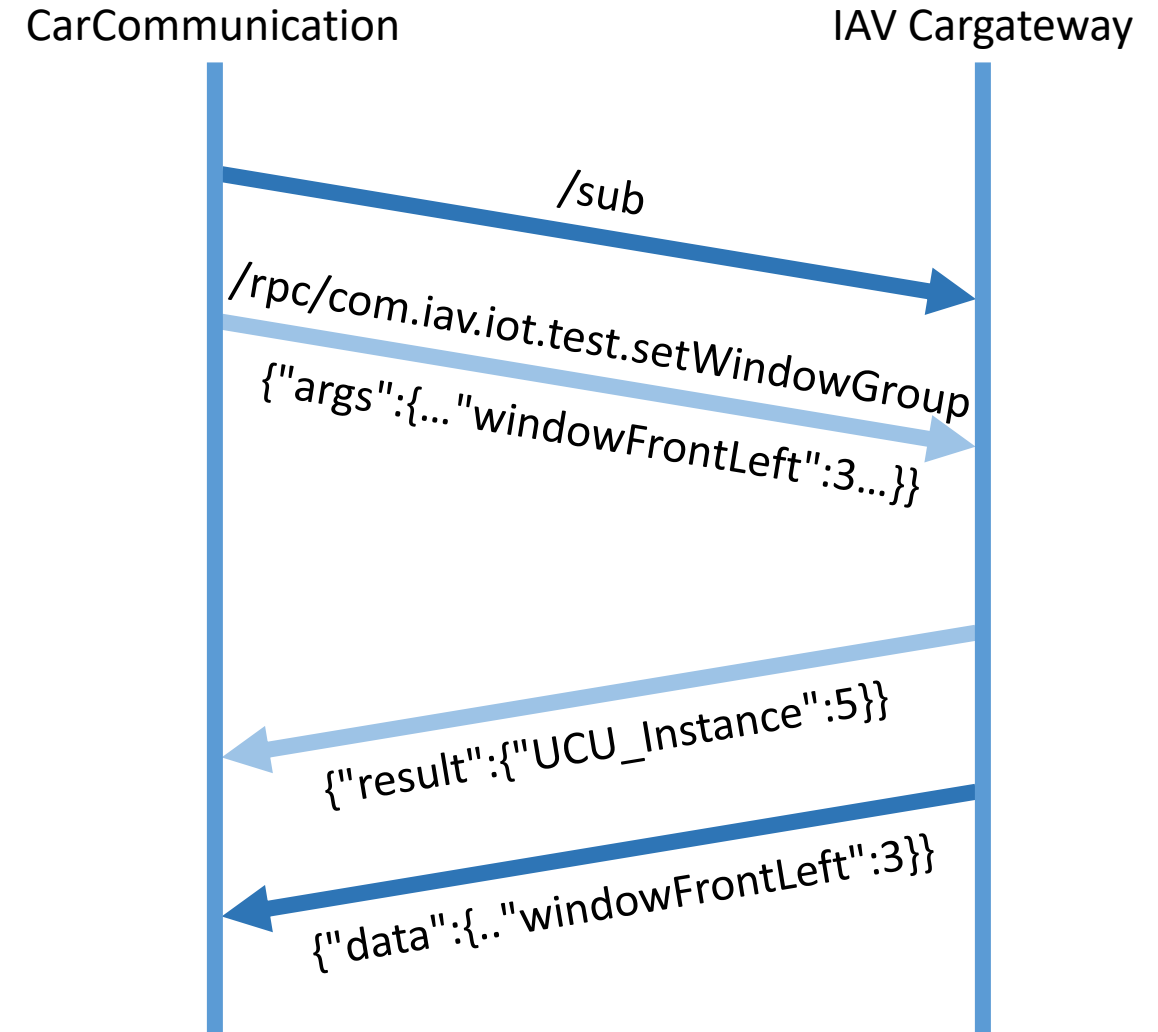
- RPC schicken
- Targets in Gruppen organisiert
- Return parsen mit Boost Property Tree

## Beispiel getWindowGroup:

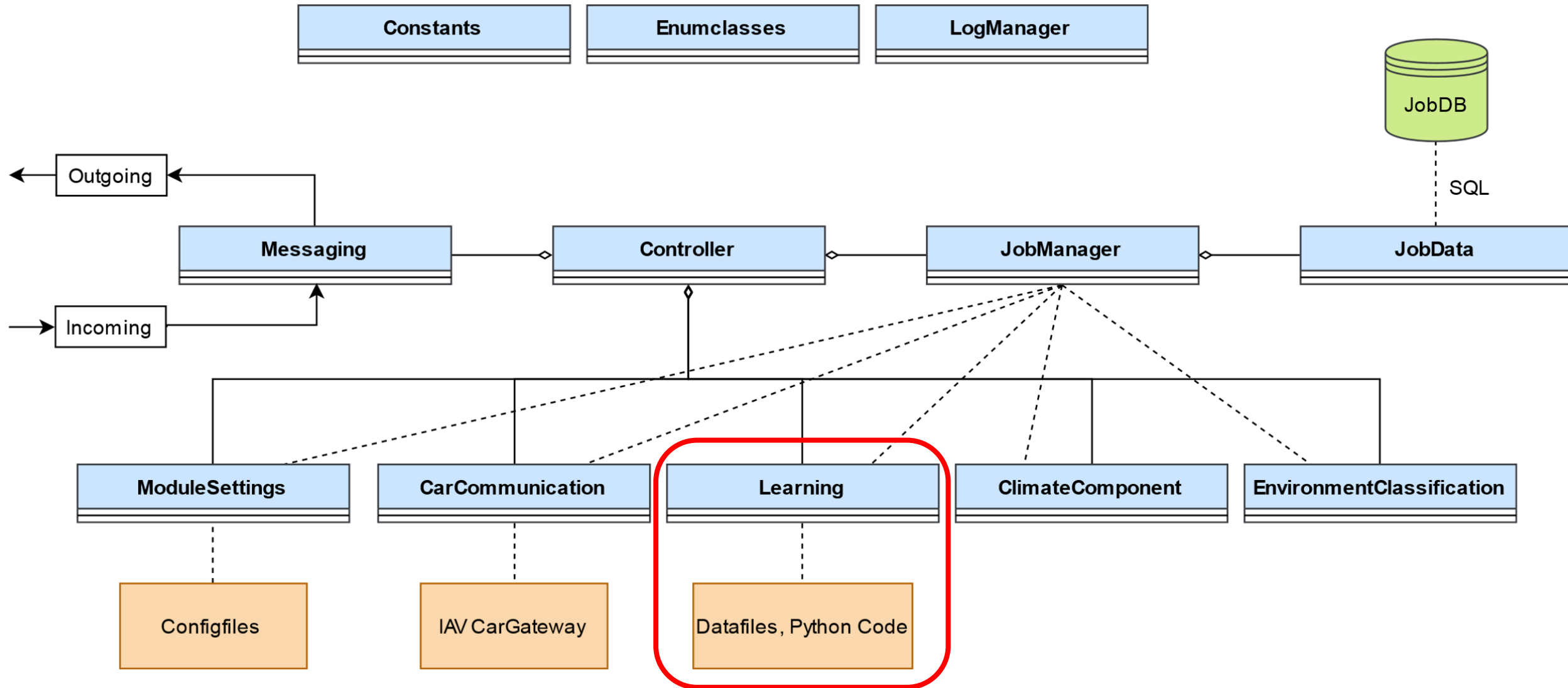
- Zustand Fenster
- Panoramadach
- Zustand Rollos



- Vergleich Ist- mit Soll-Zustand
- /sub schicken
- RPC schicken
- Instanznummer erhalten
- Return parsen mit Boost Property Tree

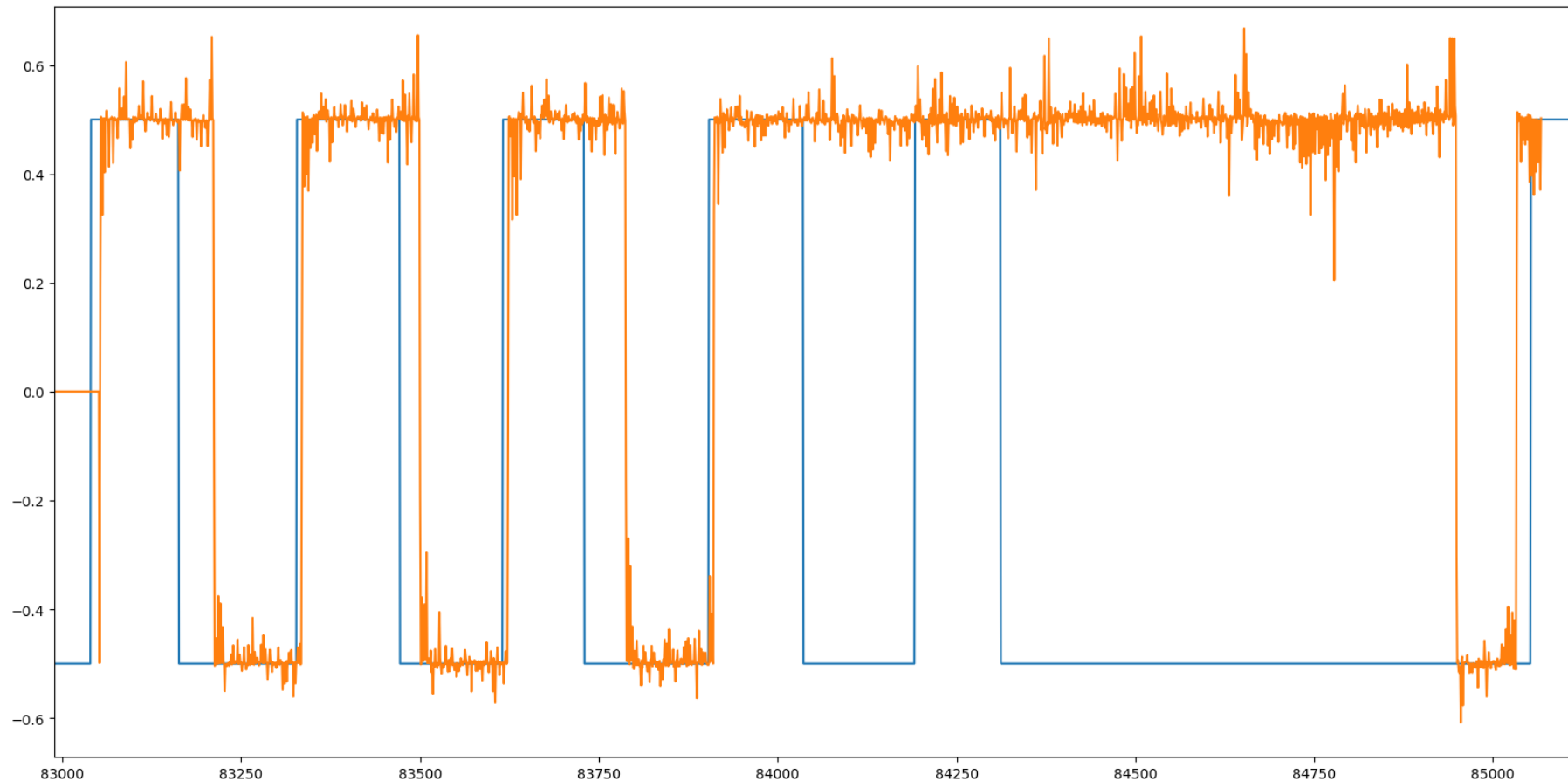


- Zyklisches Aktualisieren des CarStatus
  - Geplant: Umsetzung durch UCU und IAV Car Gateway
  - Entfernt durch Schnittstellenänderung Ende 2018
  - Architektur bereits darauf angepasst
- Emulierung durch zyklisches Senden von get
  - Läuft in separatem Thread



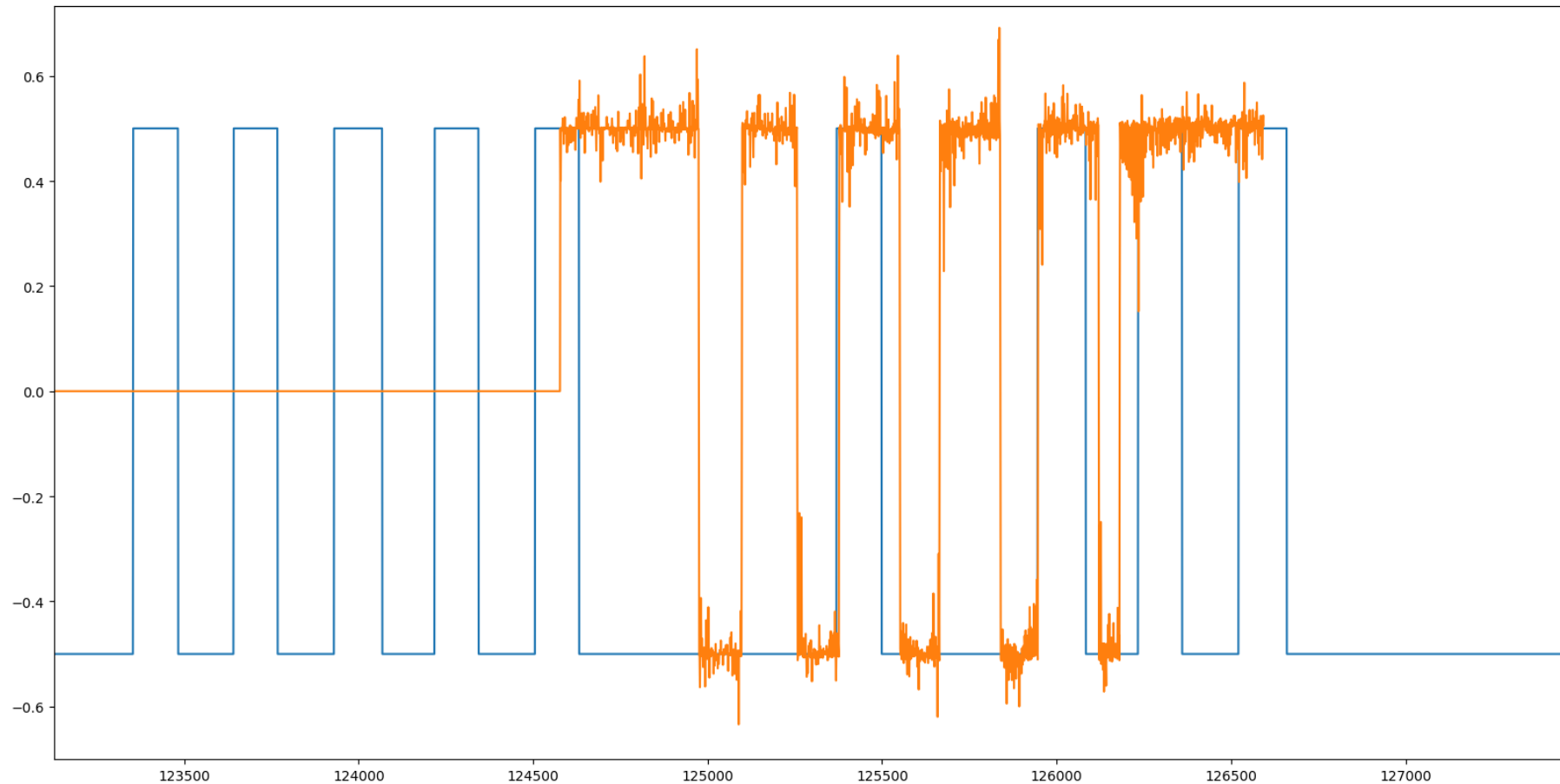
- Informationen vom Controller
  - Ob gefahren wird
  - Wer fährt
- Wird pro Nutzer alle 5 Minuten in einer .csv-Datei abgelegt
- Koeffizientenbestimmung, wenn keine Fahrt zu erwarten ist
  - Anpassung einer bestehenden Bibliothek
- Extrapolation bei Bedarf
  - Bis zu eine Stunde in die Zukunft
  - Wenn Fahrtbeginn erkannt wird, wird dieser zurückgegeben

- Arbeitszeit-Daten einer Person über 1,5 Jahre





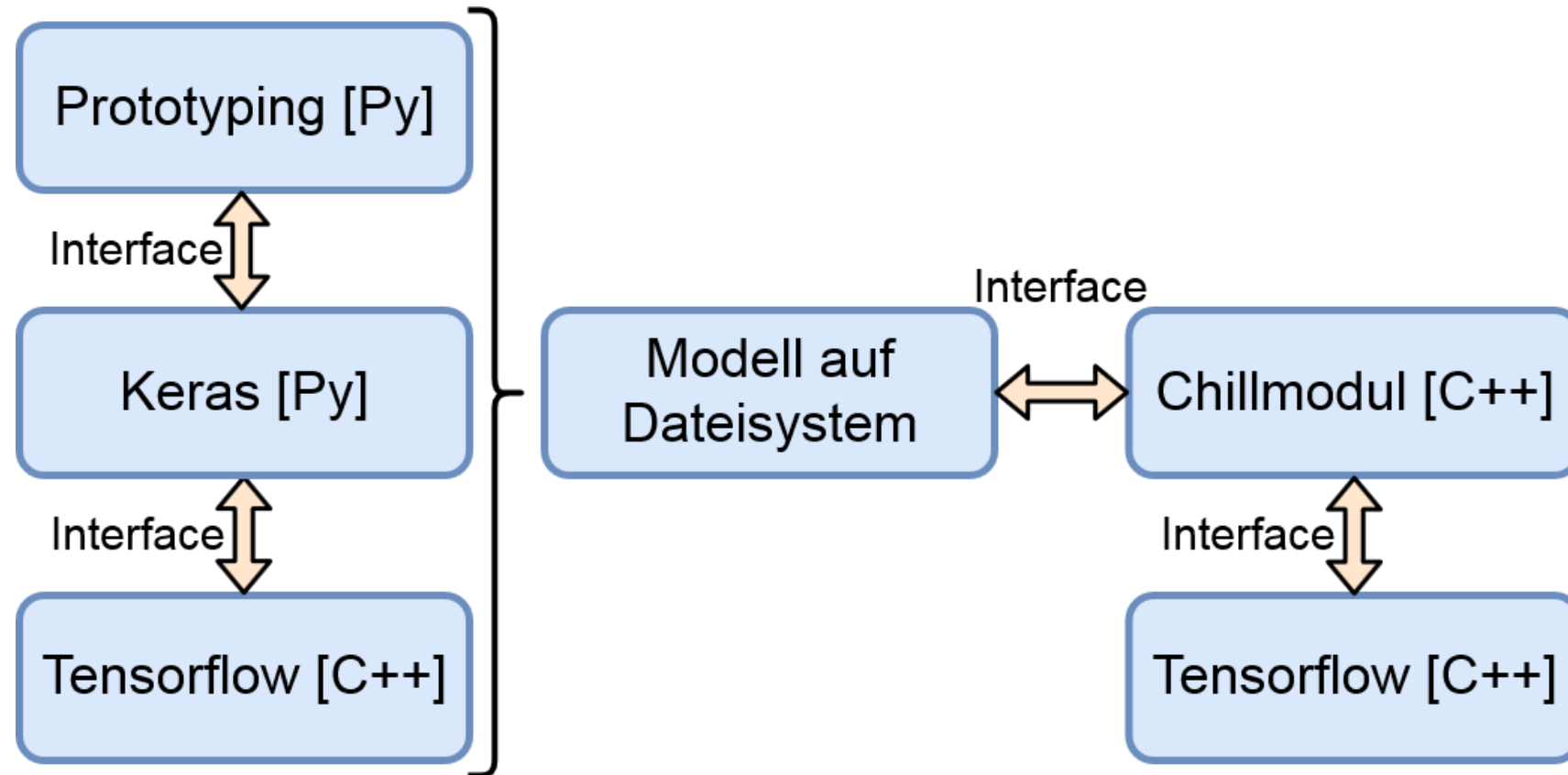
- Arbeitszeit-Daten einer Person über 1,5 Jahre



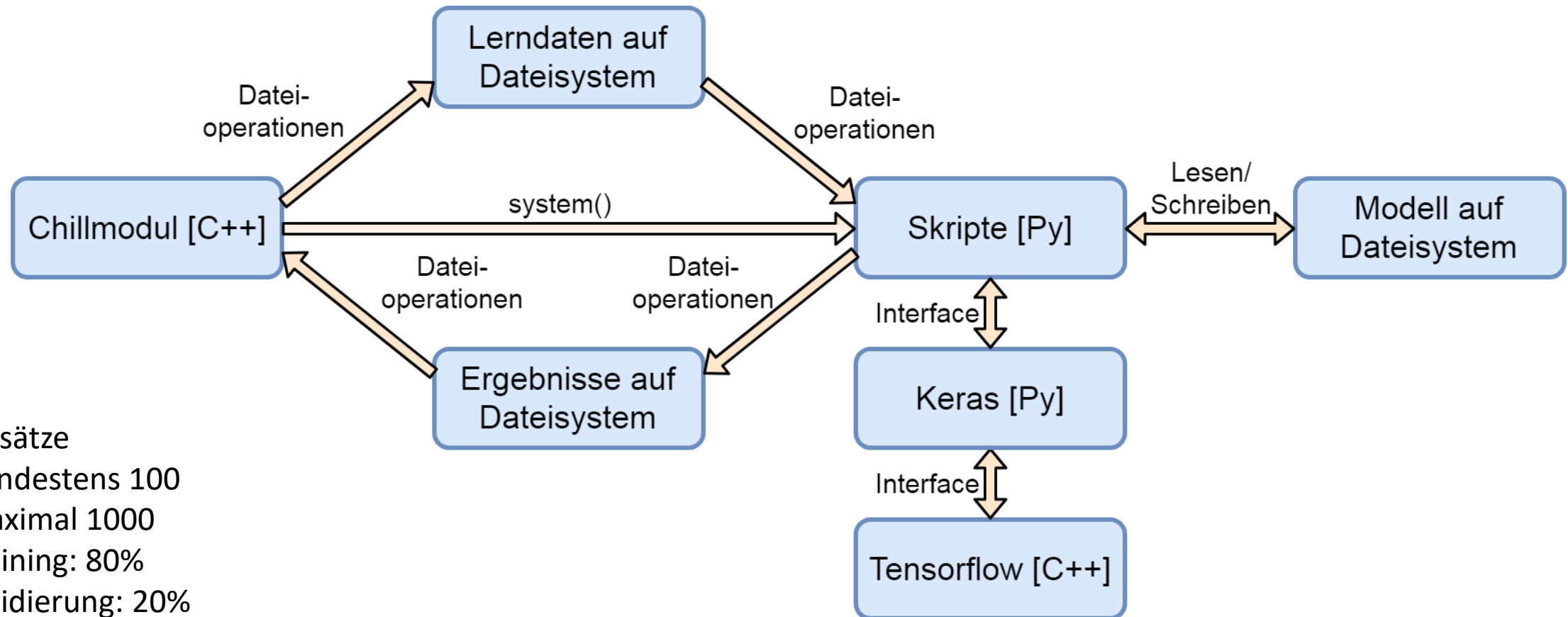
- Bisher nur Daten von einer Person vorhanden
- Ausgiebigere Evaluation für abschließende Bewertung nötig
- Ausbau des Konzeptes wäre möglich und vorteilhaft

- Prototyping in Python mit Keras und Tensorflow
  - Weit verbreitet
  - High-Level -> Schnelle Iterationszyklen
- Spätere Umsetzung mit C++-Interface von Tensorflow geplant
  - Export des Modells aus Python
  - Import, Training und Extrapolation durch C++-gesteuertes Tensorflow
- Verwendung von Tensorflow mit C++ schwieriger als gedacht
  - Keine Binaries verfügbar
  - Import und Training von Modellen hätte hohe Einarbeitungszeit erfordert

## Ursprünglicher Plan



## Tatsächliche Umsetzung



### Datensätze

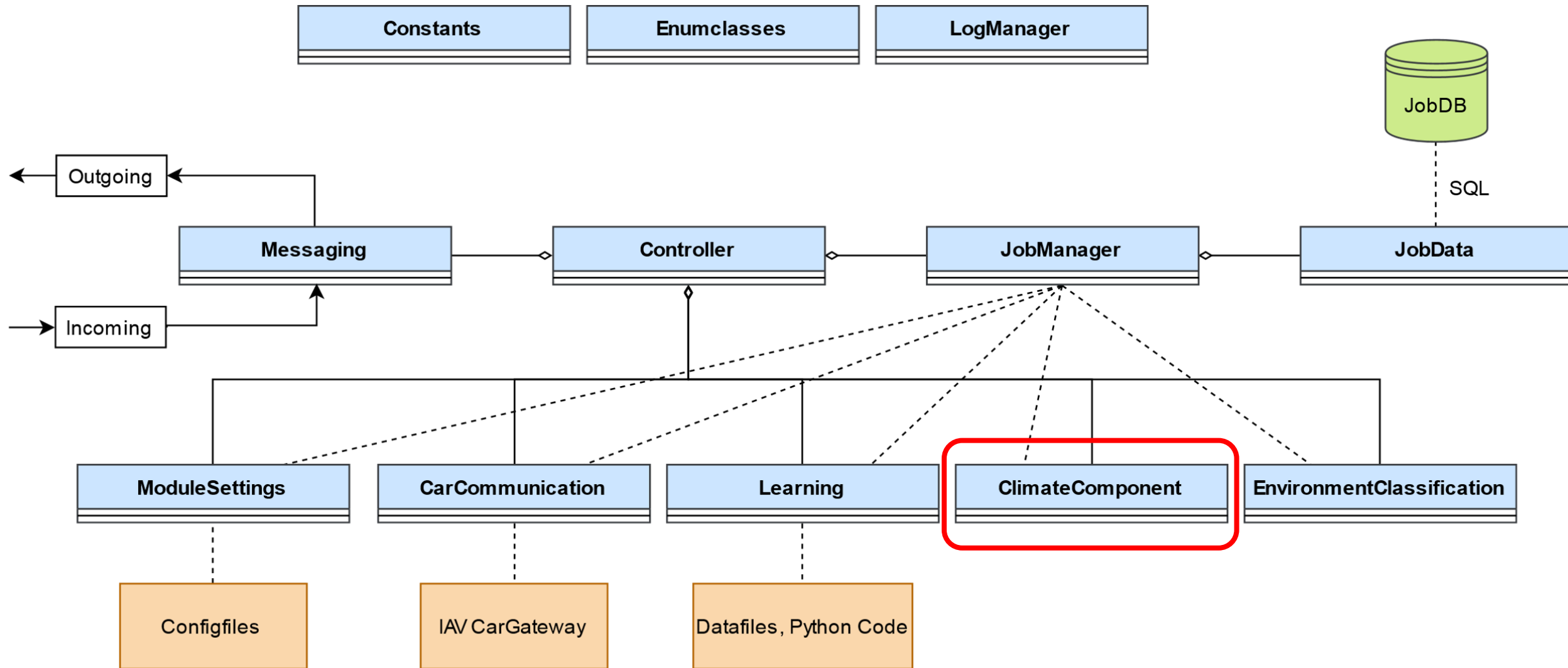
- Mindestens 100
- Maximal 1000
- Training: 80%
- Validierung: 20%

# Präferenzen Beispieldatensatz

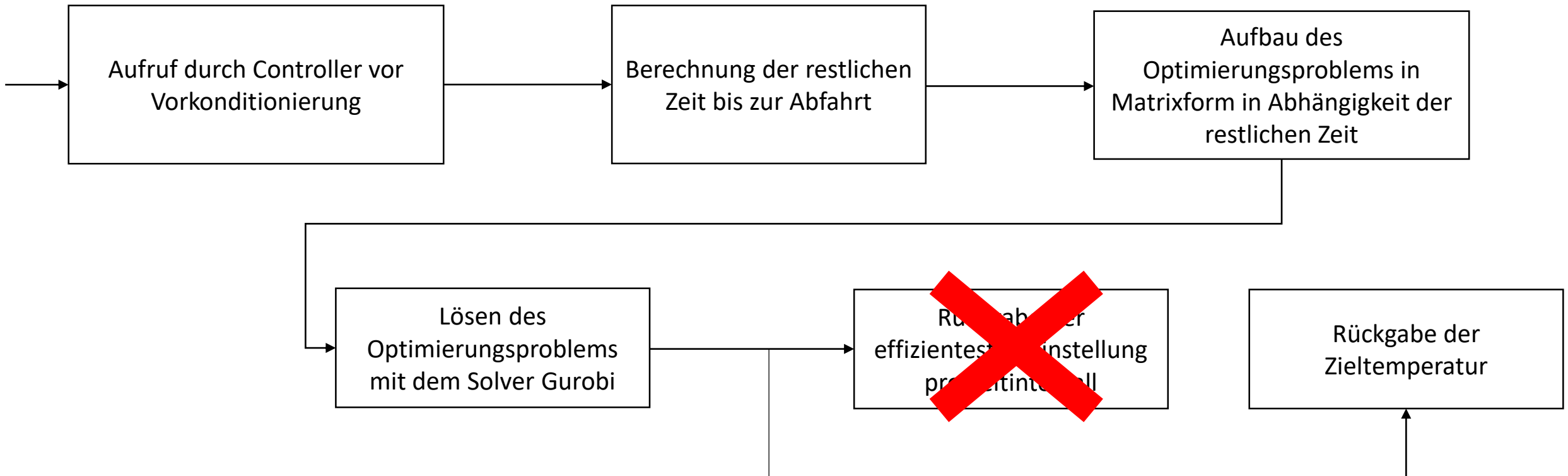


Uhrzeit		Temperatur	Wetter	Ambient	Innen-temp.	Luft-feucht.	Lenkrad-heiz.	Sitzheiz.	Sitzbel.
Sinus	Kosinus								
-0.9732	-0.2295	-5	bewoelkt	gruen	heiss	stufe4	ein	stufe3	stufe0
-0.8989	-0.4381	24	regen	blau	kuehl	stufe4	aus	stufe0	stufe2
-0.4545	-0.8907	22	regen	blau	kuehl	stufe4	aus	stufe0	stufe2
Eingabevektor				Ausgabevektor (Label)					

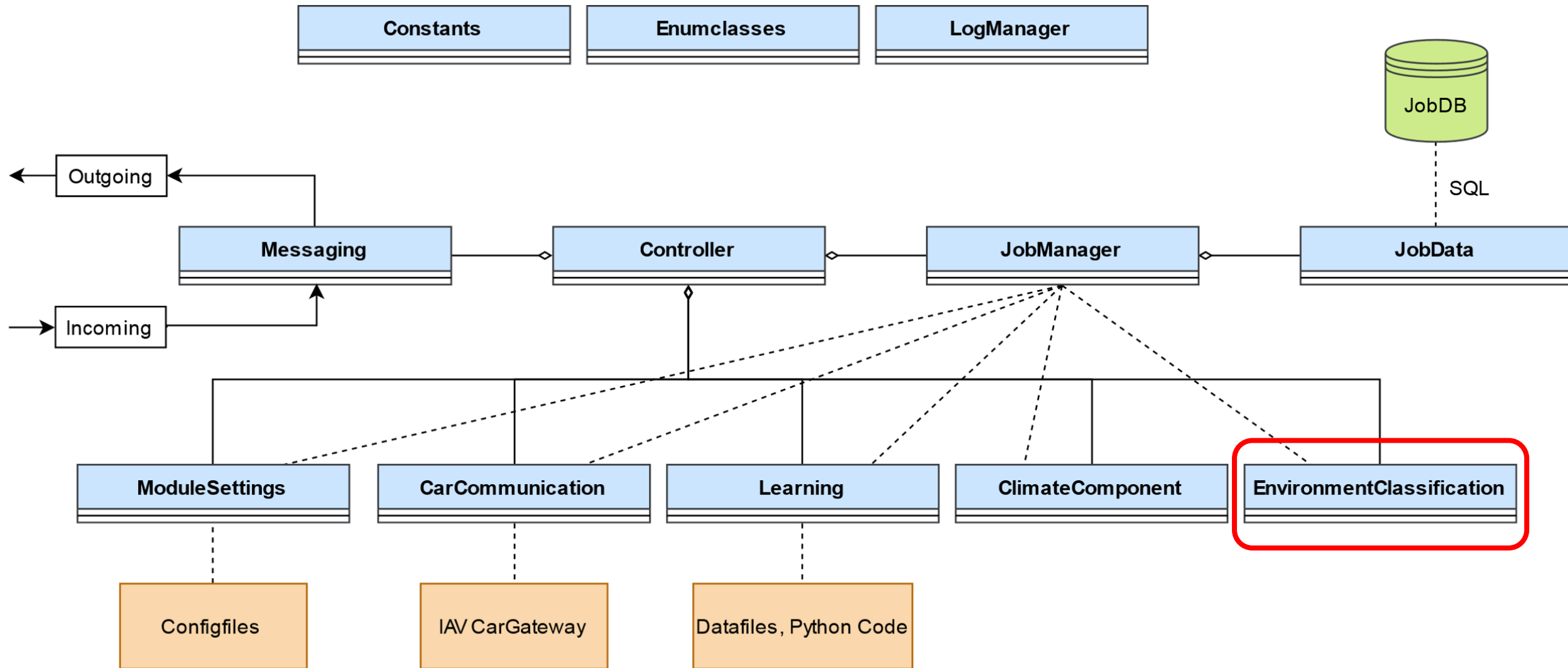
- Evaluieren mit Realdaten
- Erweitern des Netzes um weitere Aktorik
- Weitere Spezialisierung der Teilnetze möglich



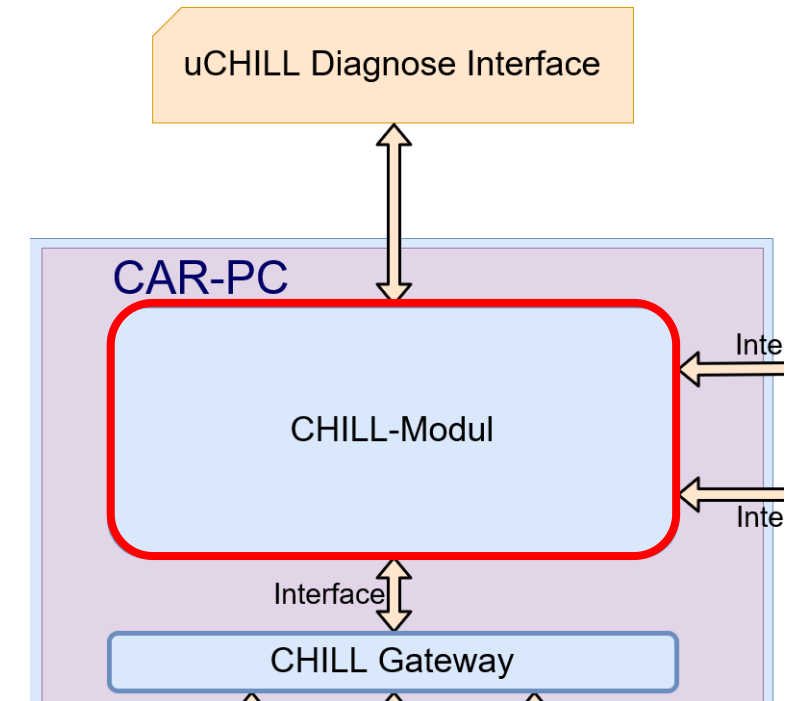




- Implementierung des Konzeptes in C++
- Floating Lizenz aufgrund von Docker Container

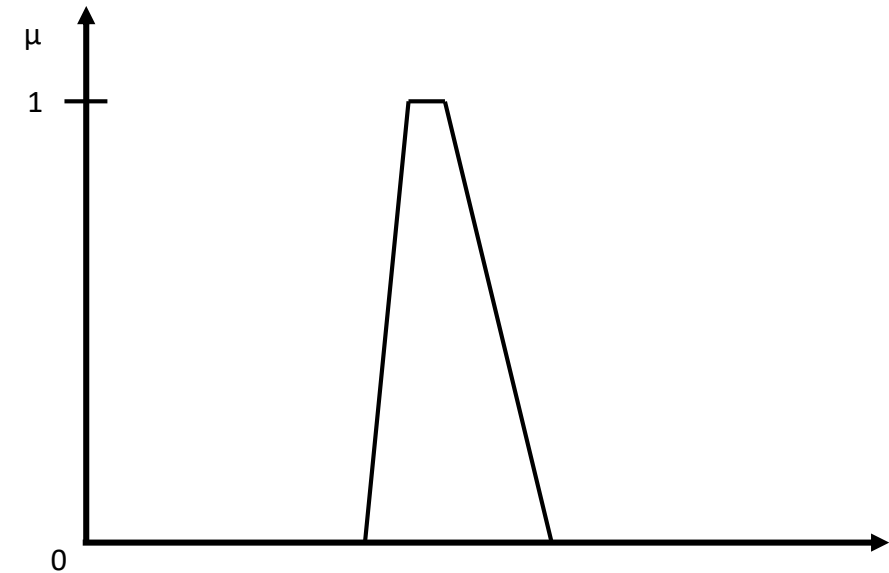


- Wird vom Controller als Objekt instanziiert
- Erstellt Fuzzy-Mengen und berechnet Zugehörigkeitswerte
- Stellt Funktion zur Klassifizierung bereit



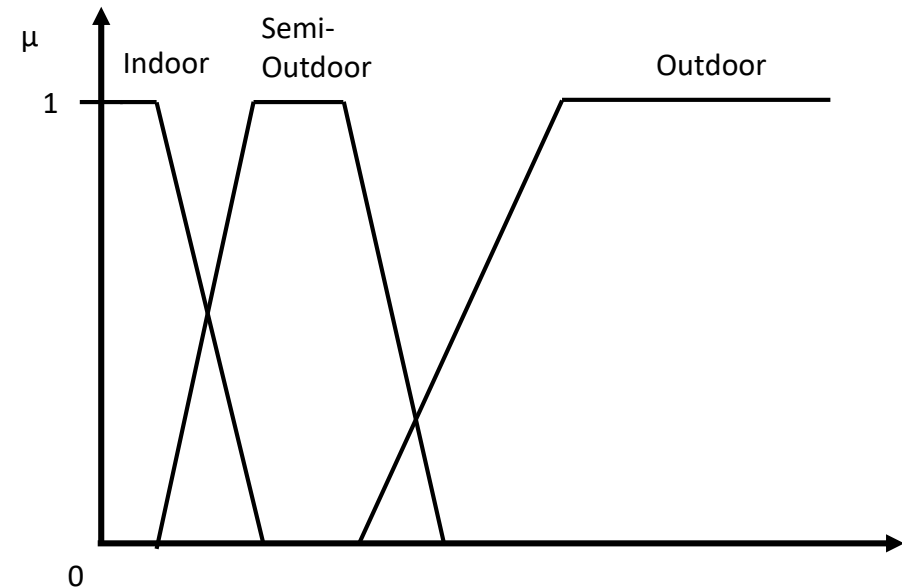
- Erstellen der Fuzzy-Mengen

```
struct FuzzySet  
{  
    double leftBase; // a  
    double leftHigh; // b  
    double rightHigh; // c  
    double rightBase; // d  
};
```



- Erstellen der Fuzzy-Gruppen

```
struct FuzzyEnvironmentClass  
{  
    FuzzySet indoor;  
    FuzzySet semiOutdoor;  
    FuzzySet outdoor;  
};
```



- Aufrufen von `classify()` vom Controller mit Parametern
  - Regenintensität
  - sichtbares Licht, Infrarotlicht
  - Wärmeeinwirkung auf das Fahrzeug
- Bei uneindeutigem Ergebnis liefert die Methode `INCONCLUSIVE` zurück

- Feldtest mit Mess-Skript erweitert um Ergebnis der Umgebungserkennung
- Messungen an 12 Standorten
  - 4 Semi-Outdoor
  - 5 Outdoor
  - 3 Indoor

- Indoor
  - An jedem Standort korrekt klassifiziert
  - Wärmeeinfluß und Lichtwerte minimal
- Akkumulierte Werte des Vektors
  - Indoor: 2,5
  - Semi-Outdoor: 0,5
  - Outdoor: 0



Messung Tiefgarage Universität



- Semi-Outdoor
  - 1/4 der Standorte korrekt klassifiziert
  - Schmäler Unterscheidungsgrad
- Akkumulierte Werte des Vektors
  - Indoor: 0,875
  - Semi-Outdoor: 2,15
  - Outdoor: 0



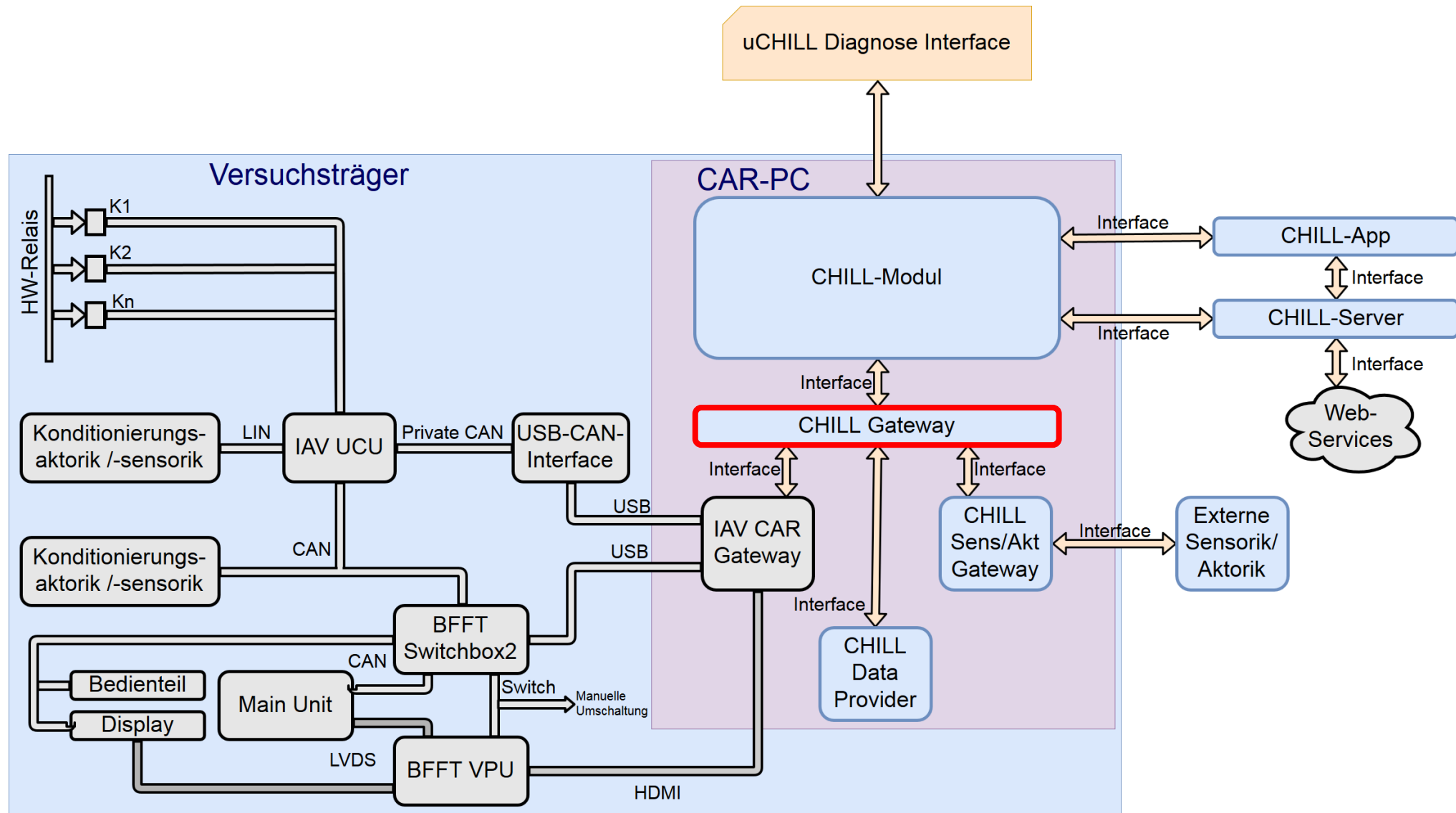
Messung Parkhaus Universität

- Outdoor
  - An jedem Standort korrekt klassifiziert
  - Wärmeeinfluß und Lichtwerte hoch
  - Nichts fälschlicherweise als Outdoor klassifiziert
- Akkumulierte Werte des Vektors
  - Indoor: 0
  - Semi-Outdoor: 0
  - Outdoor: 2,5

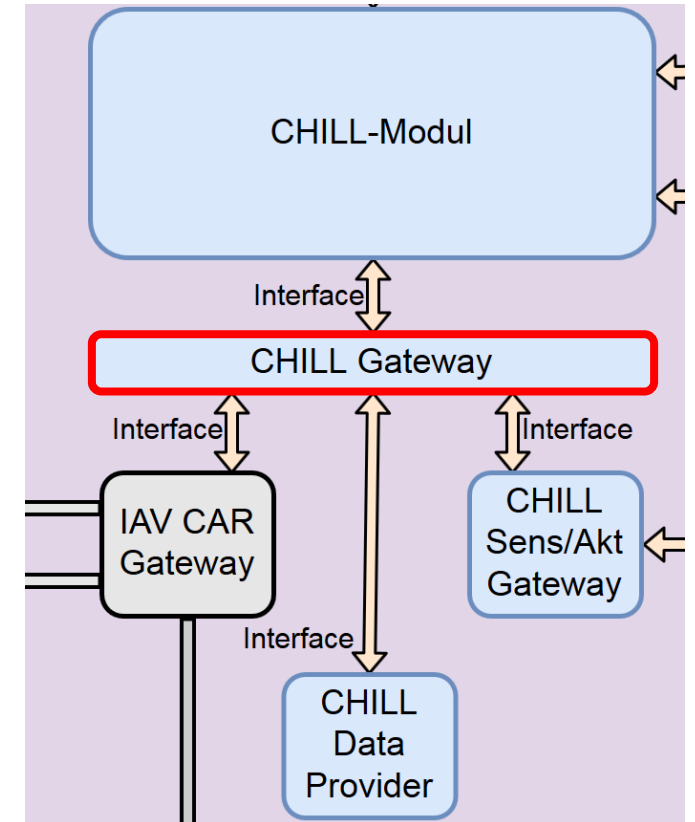


Messung Parkdeck Universität

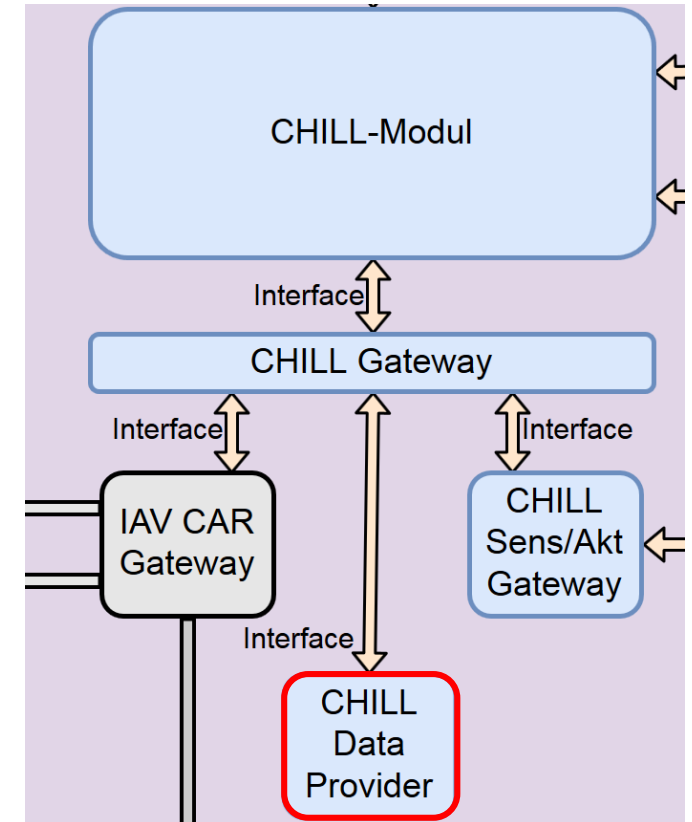
- Hinzufügen von Hardware/Sensoren
  - GSM-Modul des CarPC
  - Externe Hardware für GPS-Signal
  - Kameras
- Mehr Messdaten zum Verbessern der Fuzzy-Mengen
- Schrittgröße der Lichtsensoren verfeinern
  - Infrarot 400er Schritte in Lux
  - Sichtbares Licht 6er Schritte in Lux



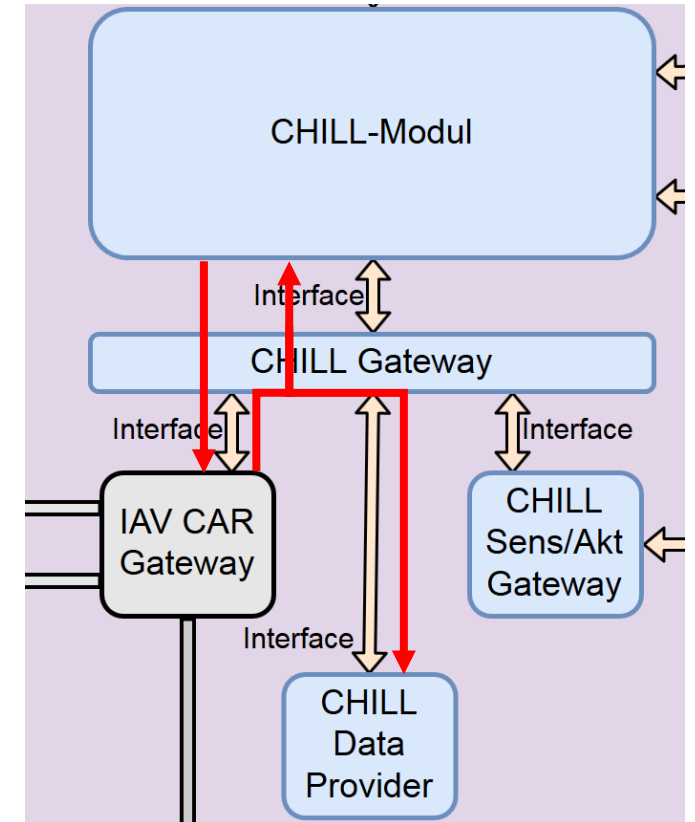
- Weiterleitung von Kommunikation Chill-Modul
- Implementiert mit Boost Beast und Boost Asio
- Drei Möglichkeiten:
  - Zum IAV Cargateway
  - Zum Chill Data Provider
  - Zum Chill Sensorik-Aktorik Gateway
- Sens/Akt Gateway für zusätzliche Sensorik/Aktorik
  - Unterscheidung automatisch anhand HTTP-Targets
- Data Provider mit manueller Einstellung



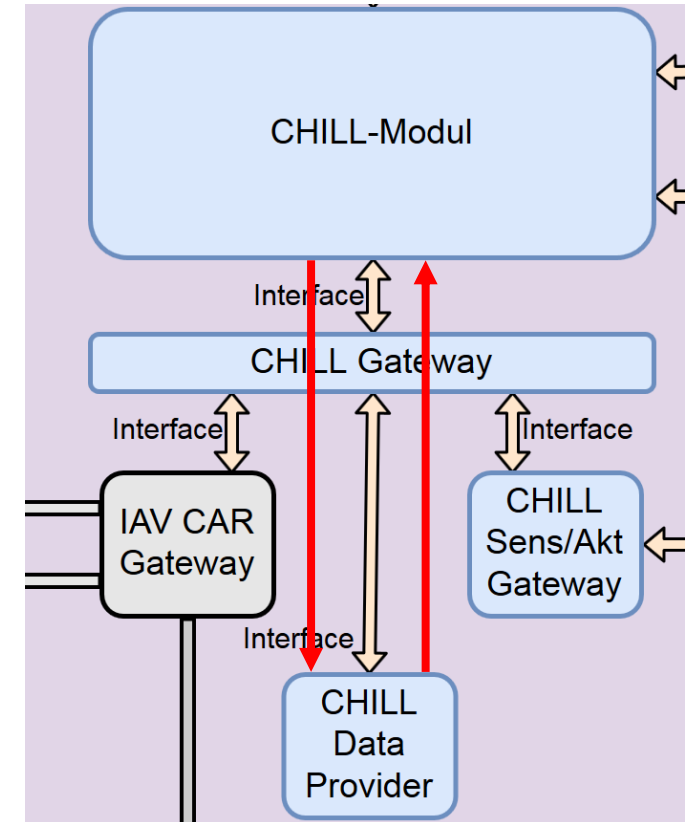
- Aufnahme und Einspielen von realen Fahrzeugdaten
- Transparent für das CHILL-Modul
- Gateway startet/beendet Aufnahme und Einspielen von Daten



- Aufnahme von Daten
  - Nachricht von Modul an CAR Gateway
  - Antwort wird vom CHILL-Gateway an DataProvider geleitet und in einer Datenbank gespeichert

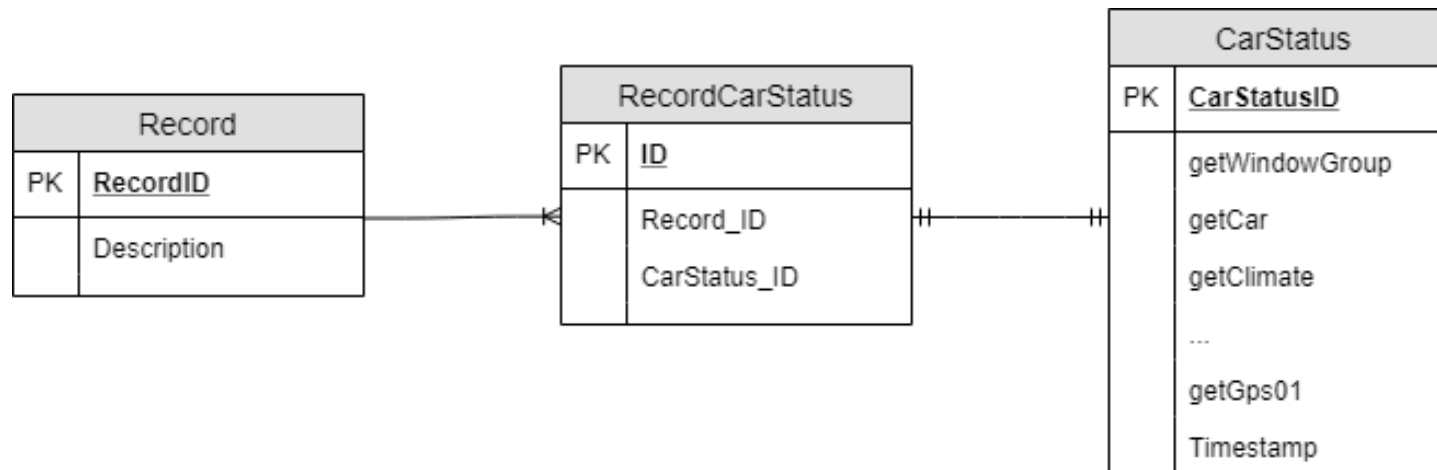


- Einspielen von Daten
  - Anfragen des CHILL-Moduls werden vom CHILL-Gateway an den DataProvider geleitet
  - DataProvider liest Eintrag aus der Datenbank und antwortet mit vorher aufgenommenen Daten

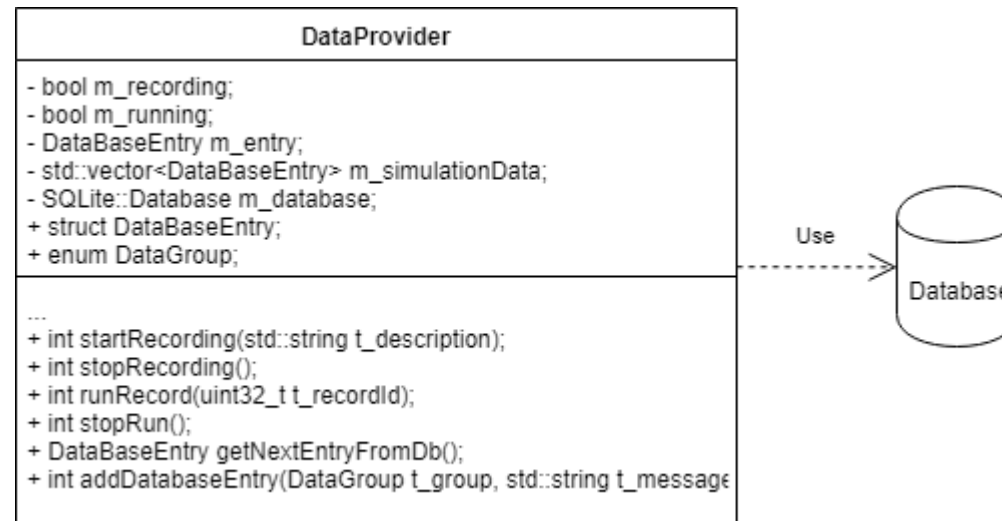




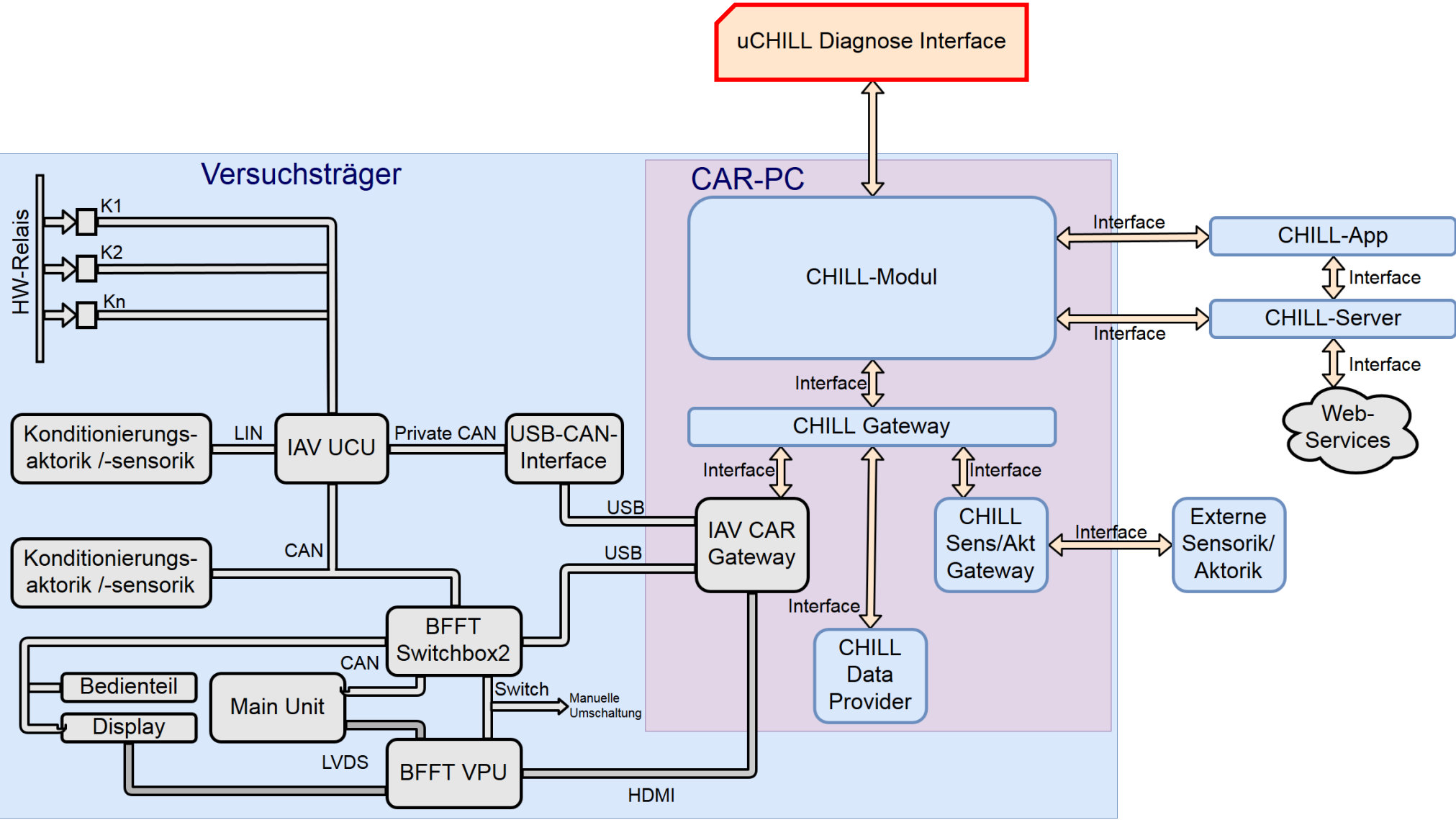
- Jede Aufnahme erzeugt einen neuen Datensatz
- Pakete des Fahrzeugs werden in CarStatus gespeichert
  - Spaltennamen stimmen mit den Paketen der UCU überein
  - 27 Spalten, erweiterbar sobald neue Pakete verfügbar



- Funktionen zum Starten und Beenden von Einspielen und Aufnahme
- Interne Struktur DataBaseEntry enthält alle möglichen Pakete
- Vektor von DataBaseEntries für das Einspielen
- SQLite Datenbank mit C++ Wrapper

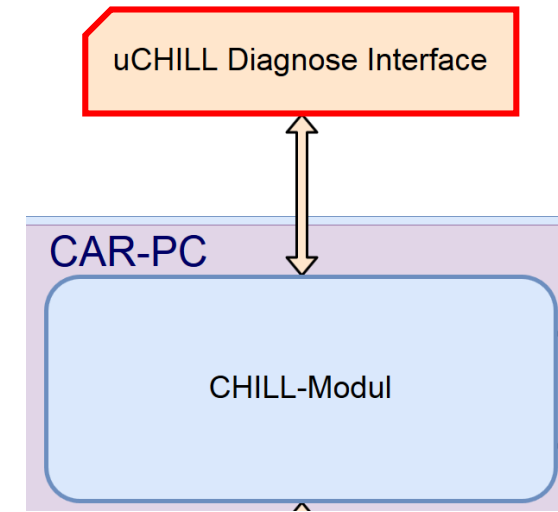


- Hinzufügen von set-Befehlen
  - Benötigt hinterliegendes Modell
- Berücksichtigung von Zeitverhalten
- Weiterleiten von Daten an das SensorAktor-Gateway





- Qt5-basierte GUI
- Übergabe von Werten aus CHILL-Modul per Boost Shared Memory
- Aktivierung/Deaktivierung per Option in ModuleSettings
  - Aktivierung aus GUI
  - Anlegen und füllen von Shared Memory durch Controller



# Diagnose Interface



Zustand Diagnosemodus Ein            Sich bewogende Zahlen: 4

Aktueller Controller Mode  
NOT\_DRIVING

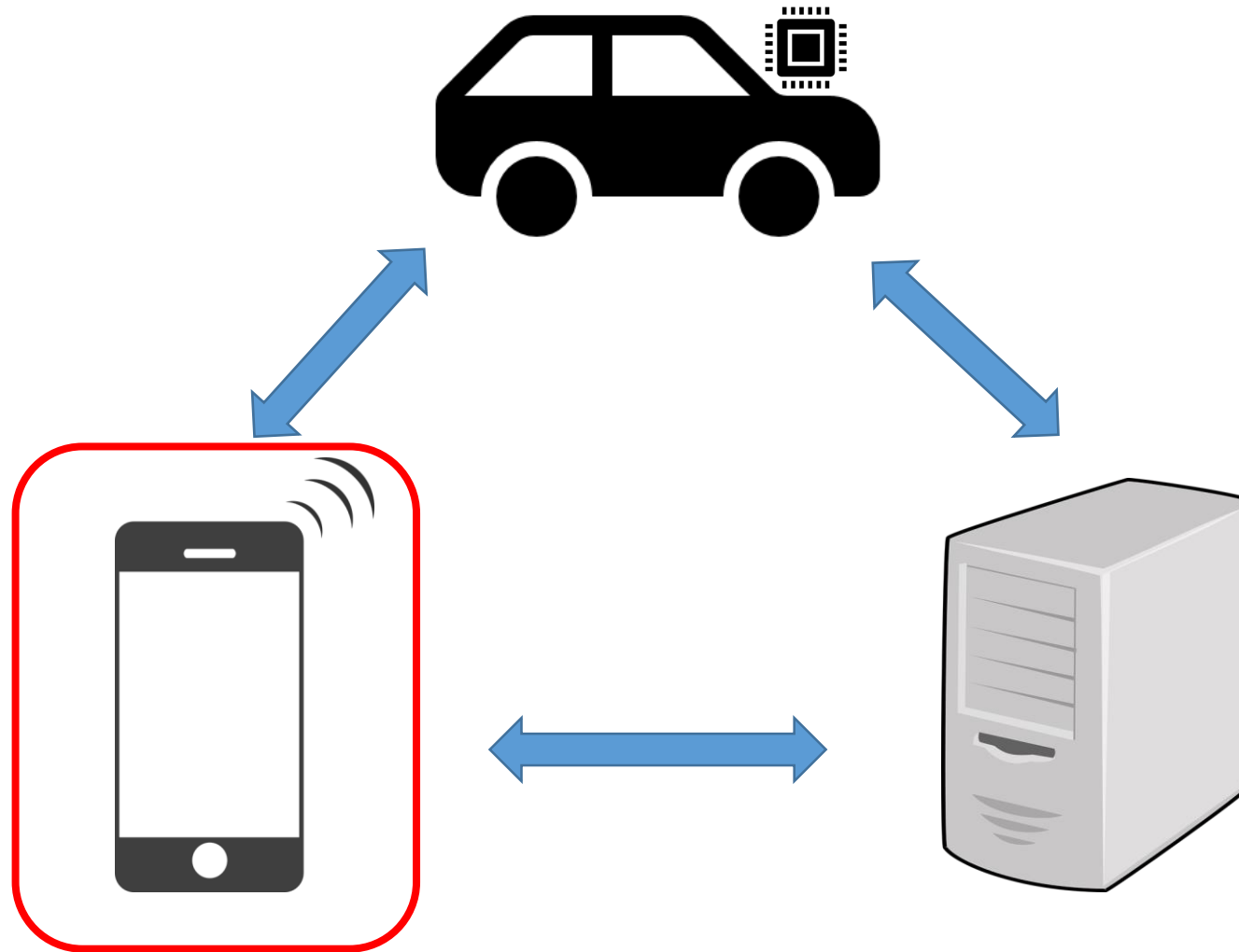
Server verbunden  
Nein

jobRunning Flag  
Job aktiv

CurrentJobData

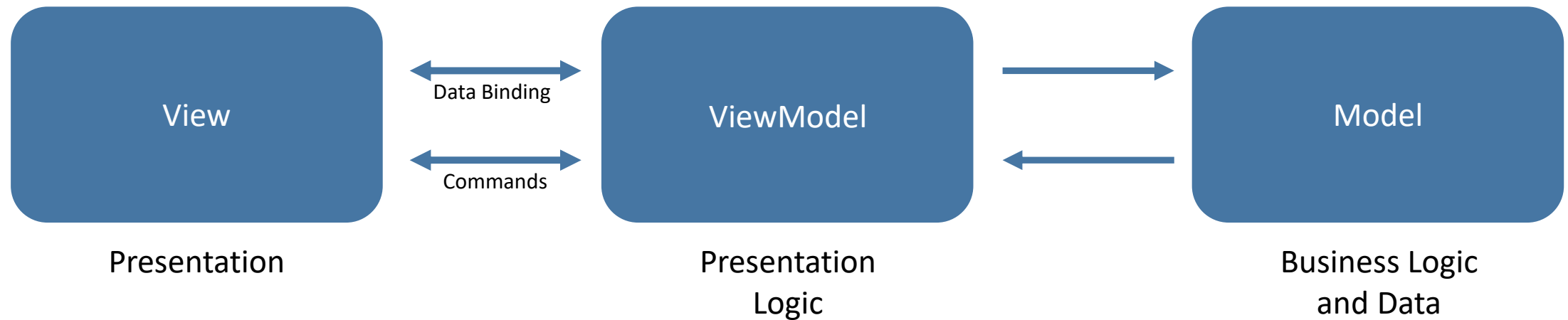
```
{
  "id": "1",
  "target_temp_fl": "25",
  "target_temp_back": "25",
  "target_humidity": "50",
  "start_time": "2019-04-01
17:22:00",
  "AppList": {
    "AppID0": "1",
    "MAC0": "d4:63:c6:2e:a0:00",
    "AppID1": "3",
    "MAC1": "abc"
  },
  "target_temp_fr": "25",
  "ambient_r": "0",
```

# Implementierung App

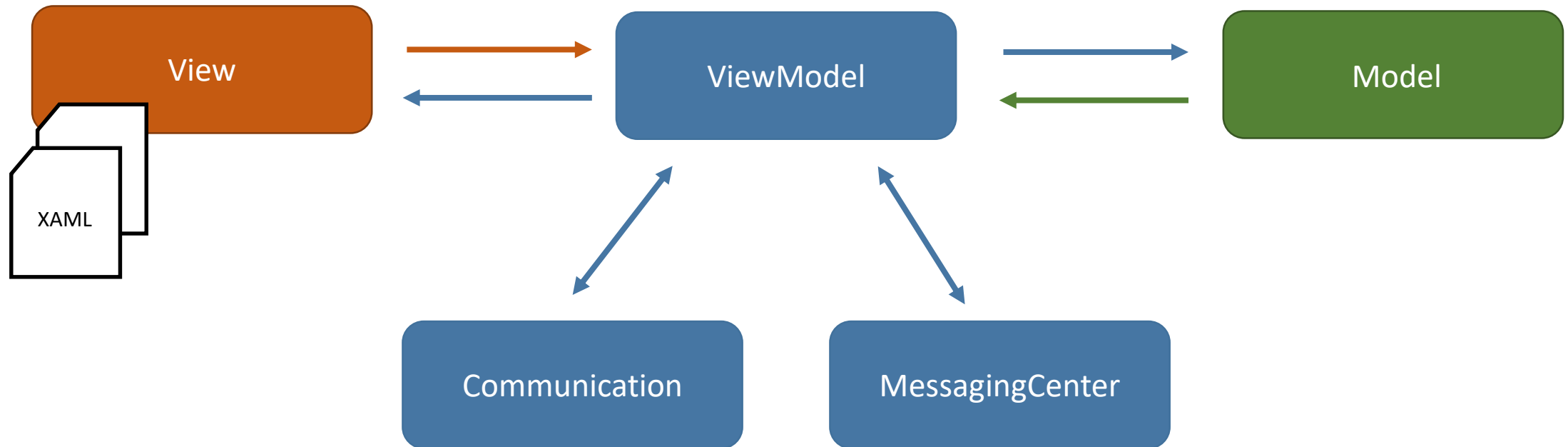


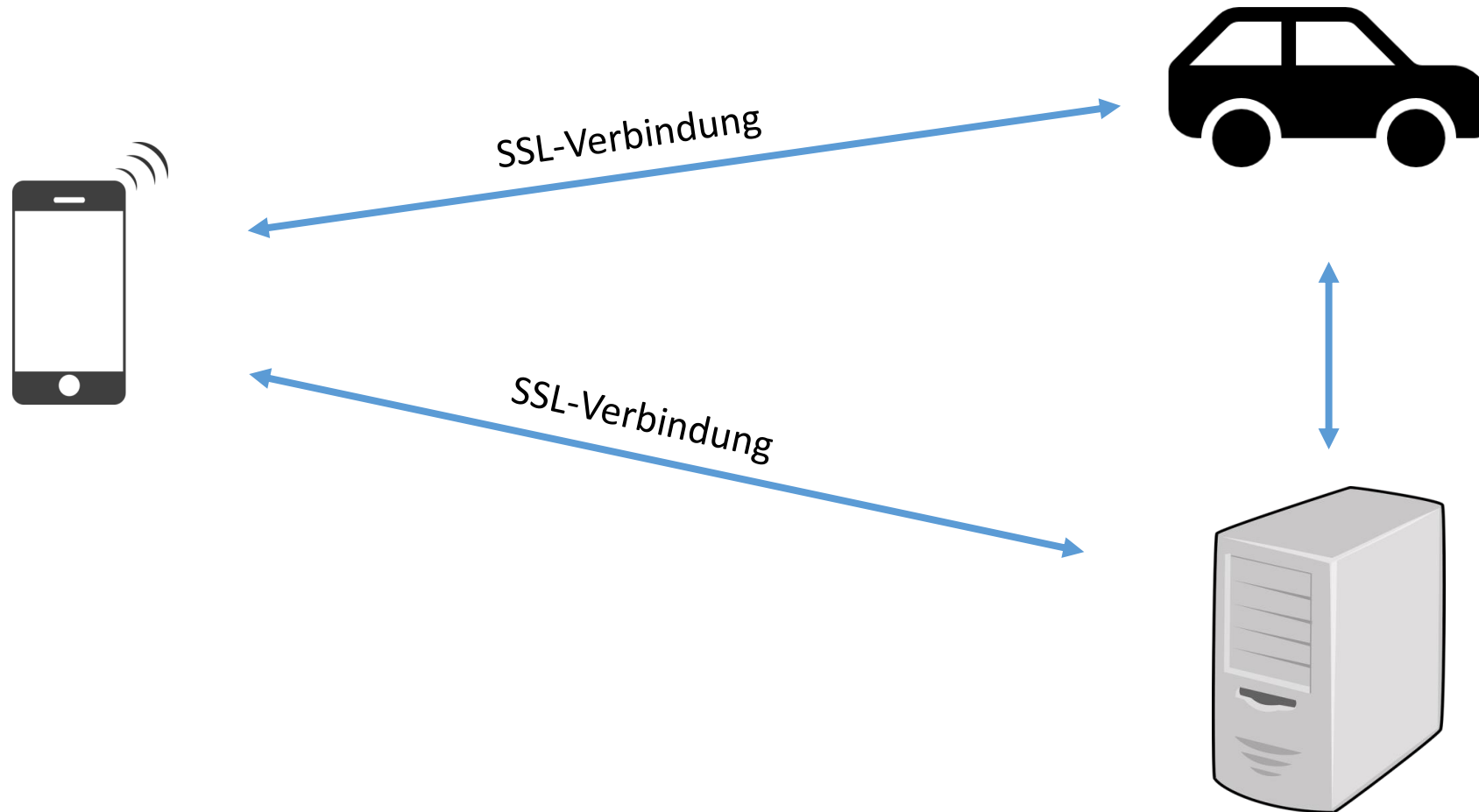
- Native App
  - Bessere Usability und Performance
- Entwicklung in C# mit Xamarin
  - Integriert Visual Studio
  - Plattformunabhängige Entwicklung von Apps
- Fokus auf Android



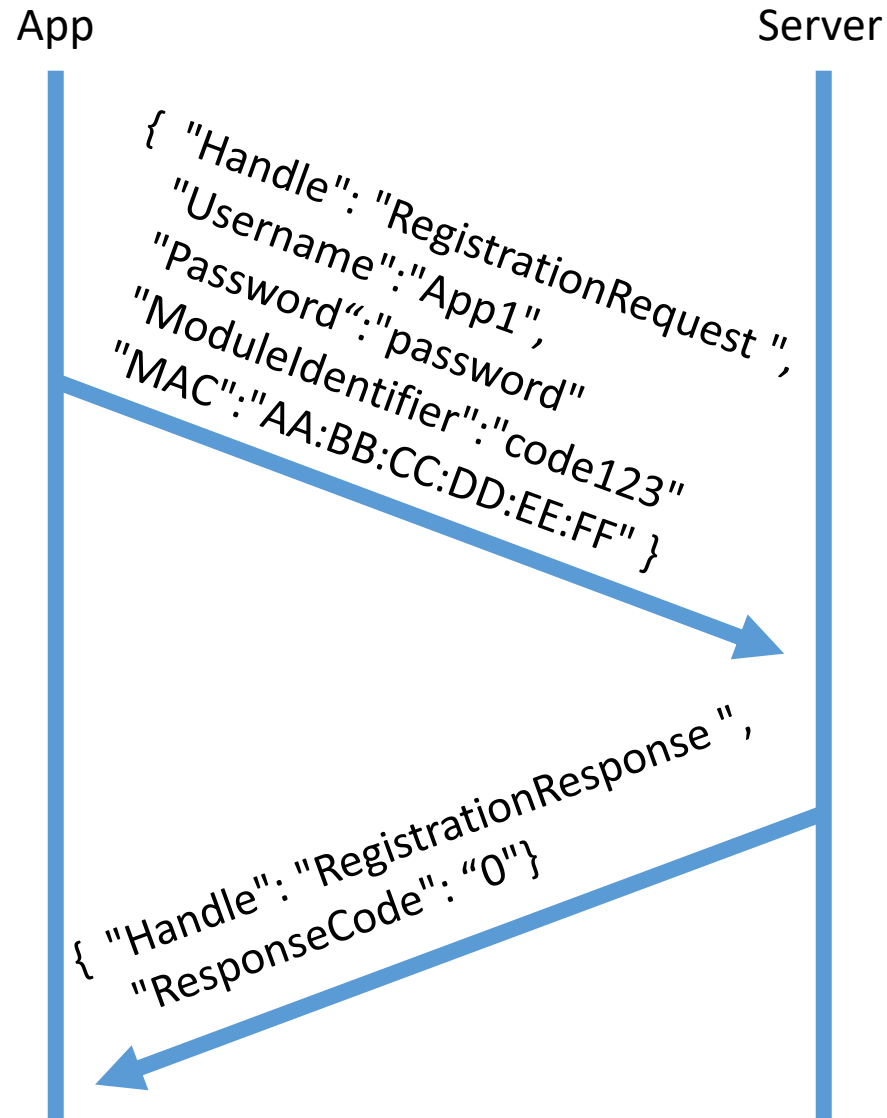


- **Vorteile:**
  - Leichte Testbarkeit durch Entkopplung der View
  - Austauschbarkeit





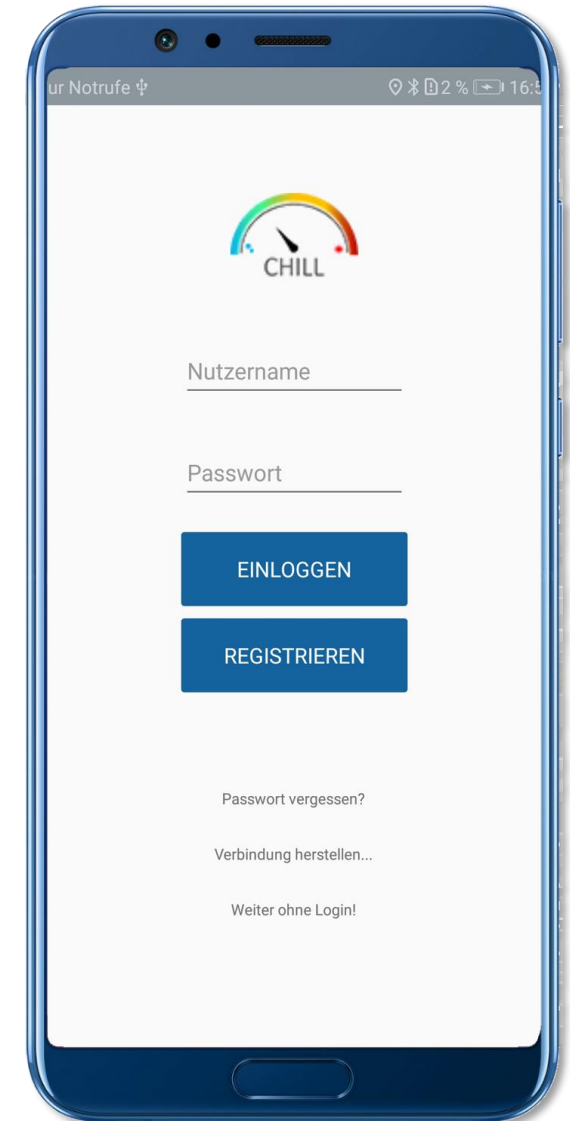
# Beispiel - Registrierung



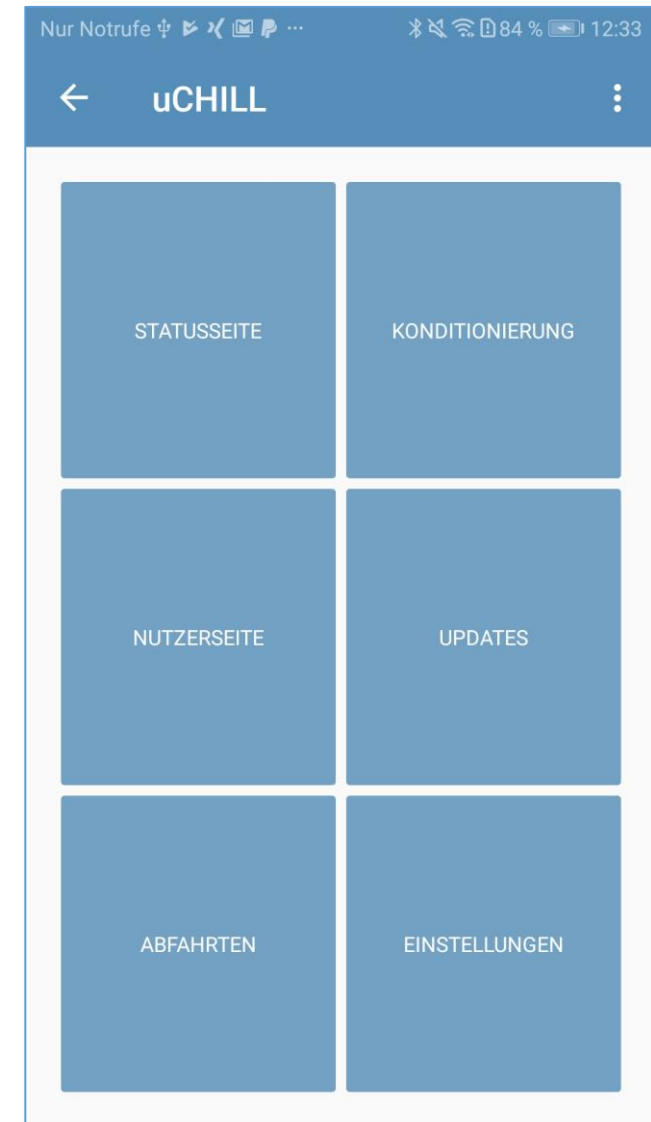
Antwort von Server wird verarbeitet:

- 0 : "Registrierung Abgeschlossen!"
- 1 : "Name schon vergeben"
- 2 : "Nicht erfolgt"

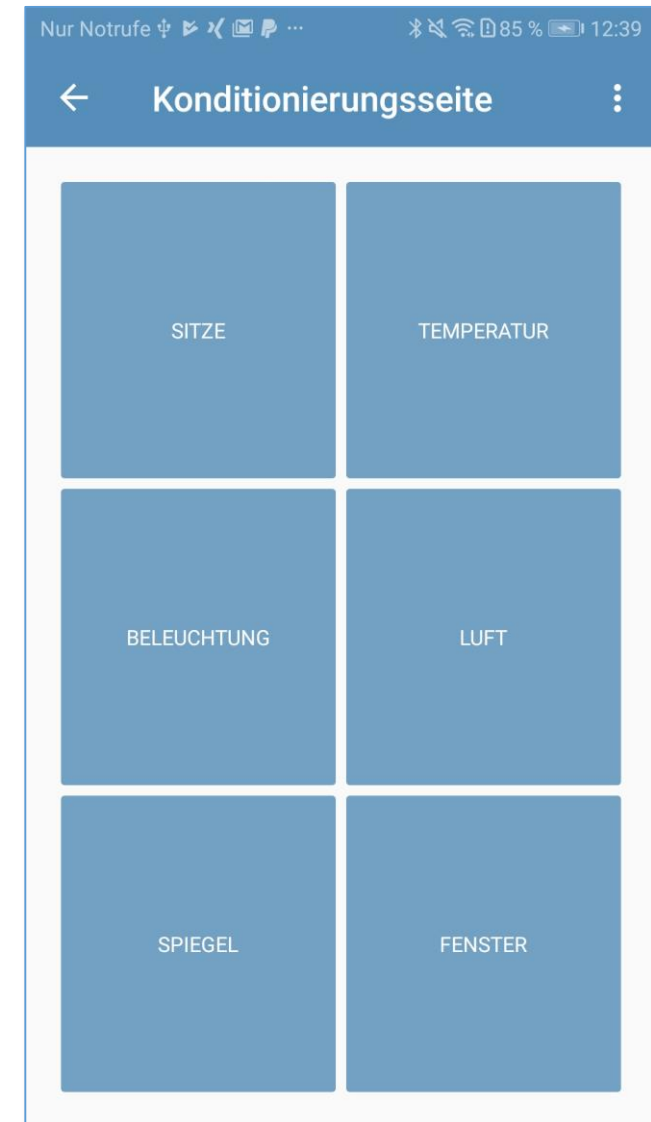
- Eingabe von Nutzernamen und Passwort
- Überprüfung vom Server
- Möglichkeit der Registrierung bei Erstanmeldung
- Einmaliger Registrierungscode pro Modul



- Statusseite
  - Aktuelle Statusdaten des Fahrzeuges
- Nutzerseite
  - Ändern des Passwortes
- Updates
- Einstellungen
  - Bereitstellungsintervall einstellen
  - Urlaubsmodus einstellen
- Konditionierung
- Abfahrten



- Temperatureinstellungen
- Sitzkonfiguration
- Fenstereinstellungen
- Beleuchtungseinstellung
- Spiegelkonfiguration
- Lufteinstellungen

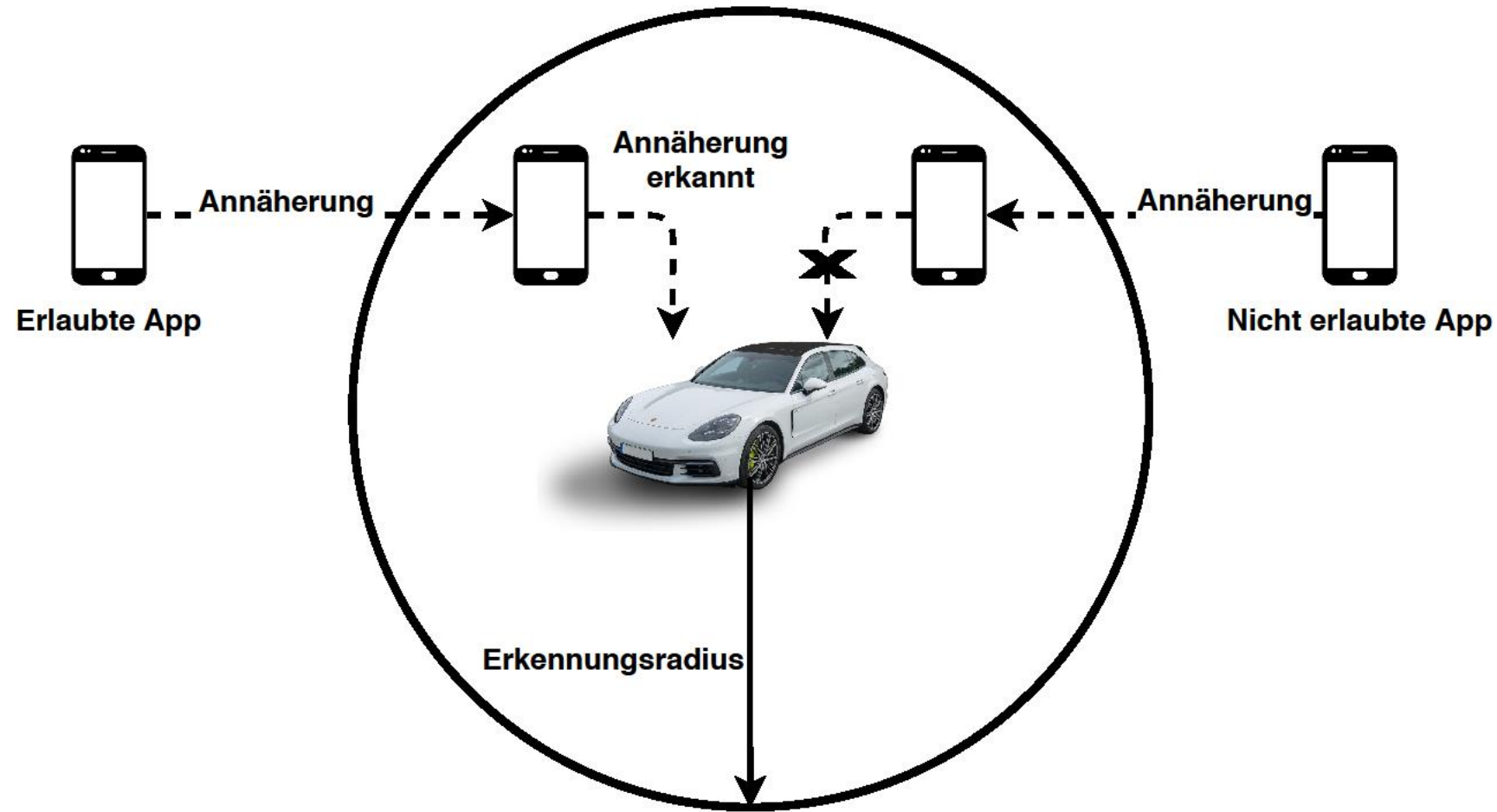


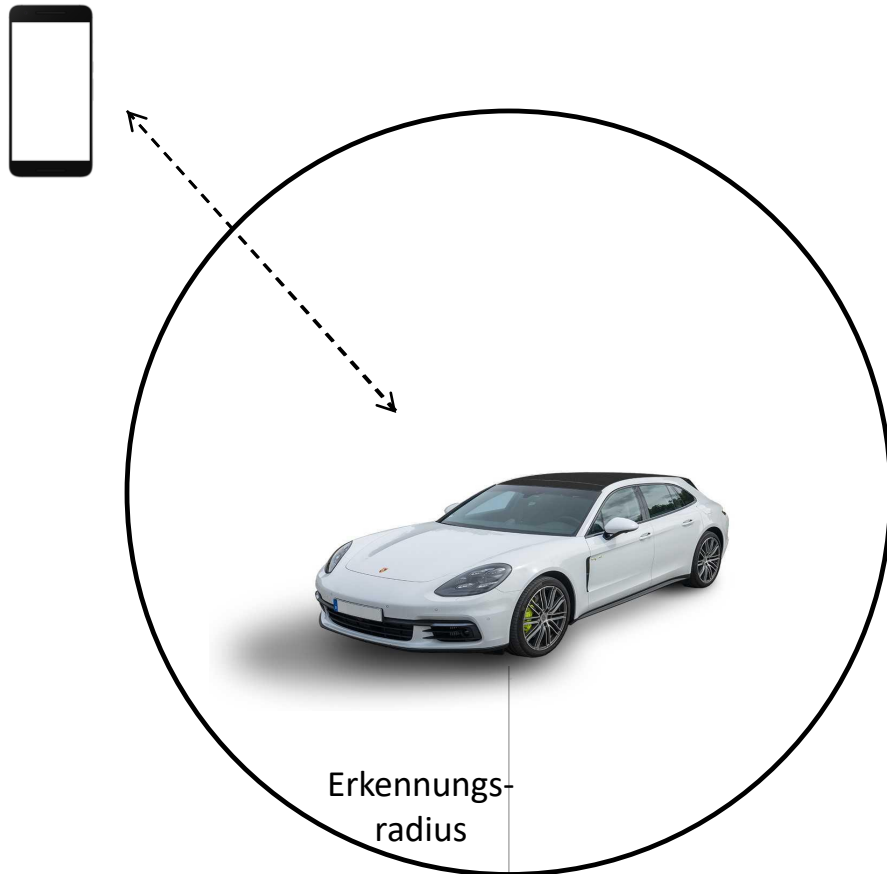
- Nächste geplante Abfahrt einstellen
- Liste der Termine einsehen
- Gewählten Termin übertragen
- Termine aus der Liste löschen





# Annäherungserkennung





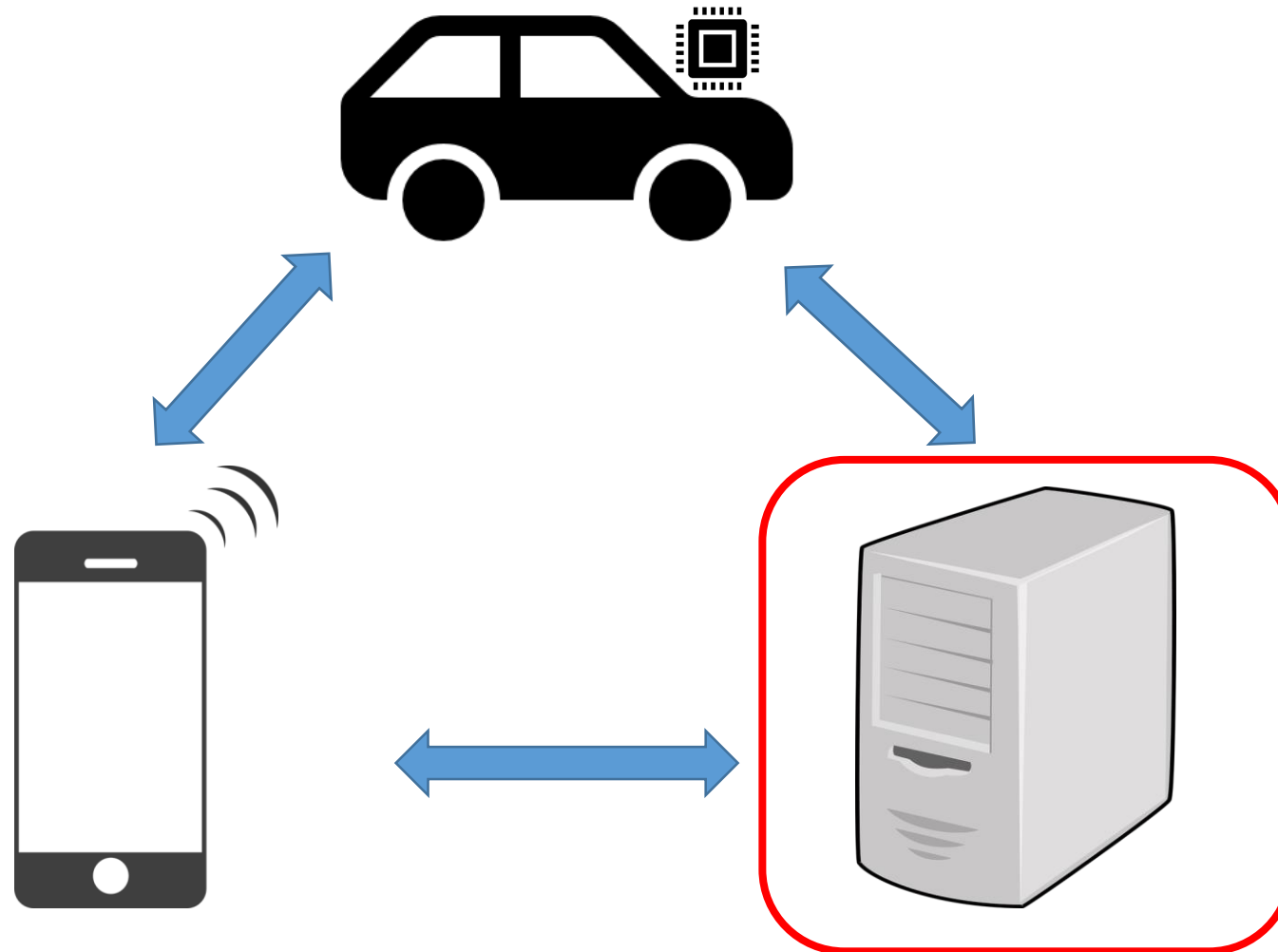
- CarPC stellt Access Point für Direktverbindung bereit
- App sendet Nachricht an Modul
- Überprüfung Annäherungsstatus in Jobausführung
- Überprüfung des Fahrzeugtürstatus zur Entscheidung ob Fahrer einsteigt

- Zuhilfenahme der Xamarin-Klasse WifiManager
  - Asynchron scannen nach Access Point
  - Verbinden, wenn in Reichweite
  - Auslesen des RSSI-Wertes aus dem WifiManager
- Senden der Approach-Nachricht, wenn das Signal stärker ist als der definierte Schwellwert

```
{  
  "Handle": "ApproachDetected",  
  "AppID": "App1",  
  "MAC": "AA:BB:CC:DD:EE:FF"  
}
```

- Verbindung mit dem Access Point dauert zu lange
  - „Überprüfung der Verbindung“
  - Wenn bereits mit Access Point verbunden, keine Zeitprobleme
- Nur Access Point darf im Smart Device eingespeichert sein
  - Android verbindet sich sonst unabhängig von App mit stärkstem WLAN
- Keine API für WLAN unter iOS
- RSSI und somit Erkennungsradius abhängig von diversen Faktoren
  - Abschirmung durch Fahrzeugkarosserie, wenn Kofferraum zu
  - Umstehende Fahrzeuge

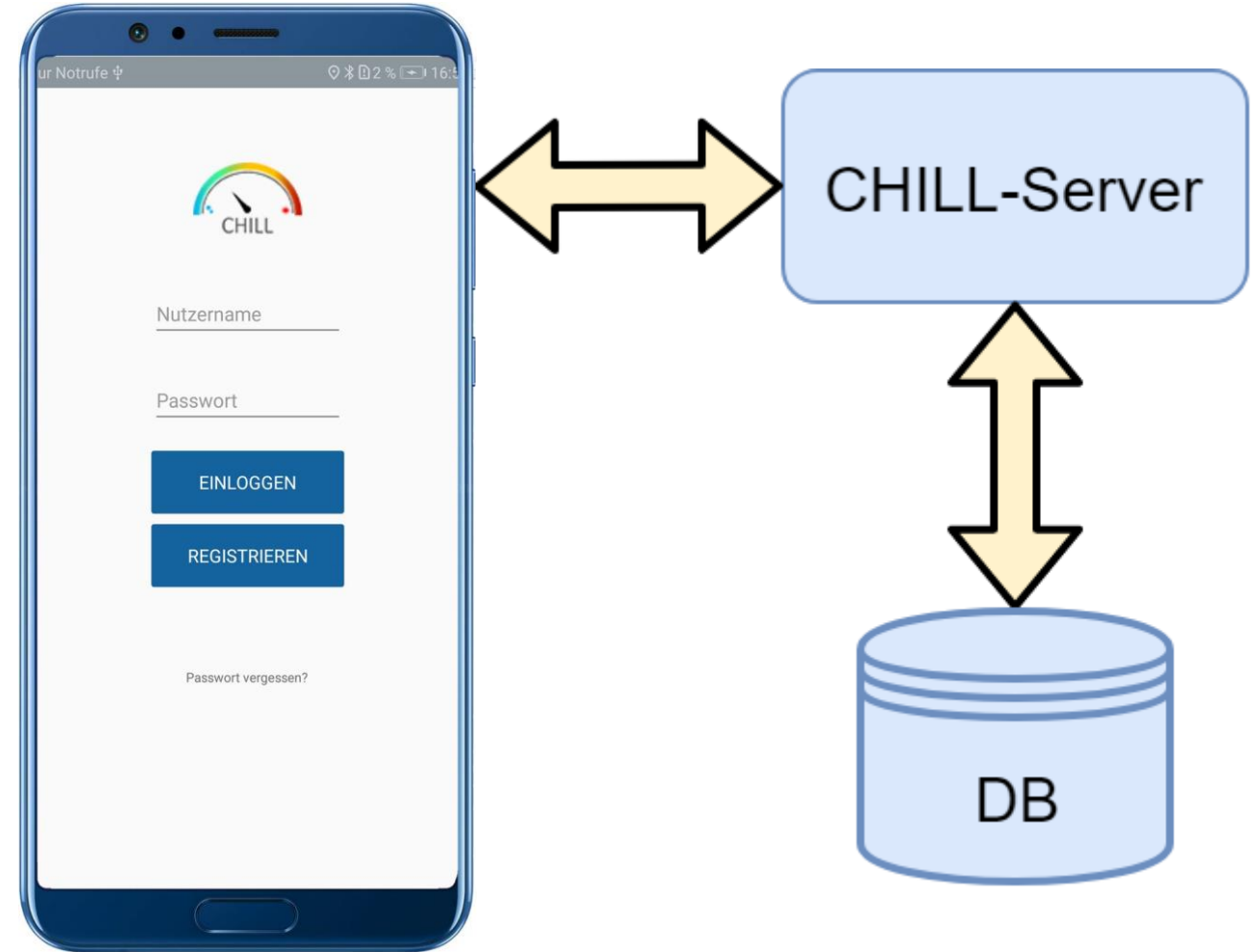
- Direktverbindung: Anbringen von WLAN-Antenne außerhalb des Fahrzeuges
  - Erhöhung der Reichweite
- Für Annäherungserkennung Bluetooth verwenden
  - Zeitaufwendiges Verbinden mit WLAN nicht nötig
- Weitere Funktionen in der App:
  - Kalenderintegration
  - Verwaltung Nutzerprofile
  - User Experience



# Aufgaben des CHILL-Servers



- Registrierung und Login
- Verbindungsglied
- Zwischenspeicherung von Aufträgen
- Externe Daten
- OTA-Updates



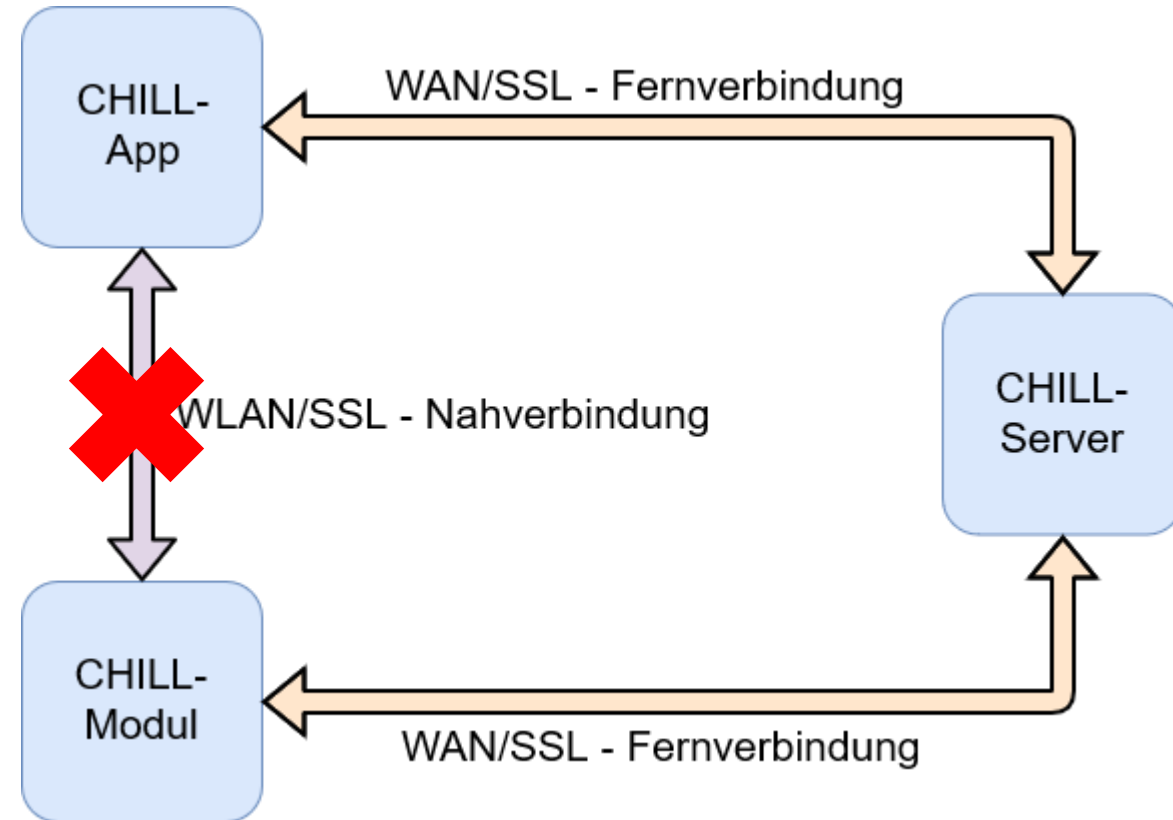
- PostgreSQL Datenbank
- Apps: Benutzername, Passwort, Modul-ID
- Module: Modul-ID
  
- Registrierung: Neuer Eintrag in die Tabelle
- Login: Vergleich der Anfragedaten mit den vorhandenen Daten aus der Tabelle



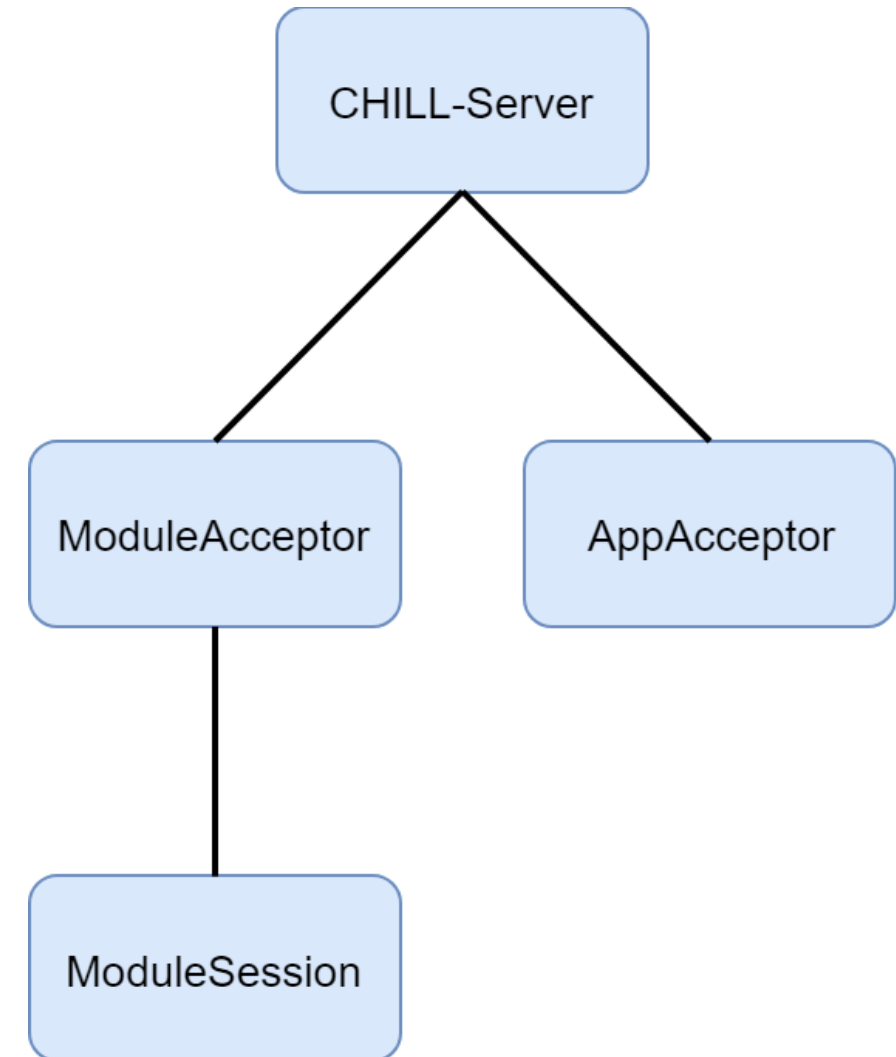
# Aufgaben des CHILL-Servers



- Registrierung und Login
- **Verbindungsglied**
- Zwischenspeicherung von Aufträgen
- Externen Daten
- OTA-Updates



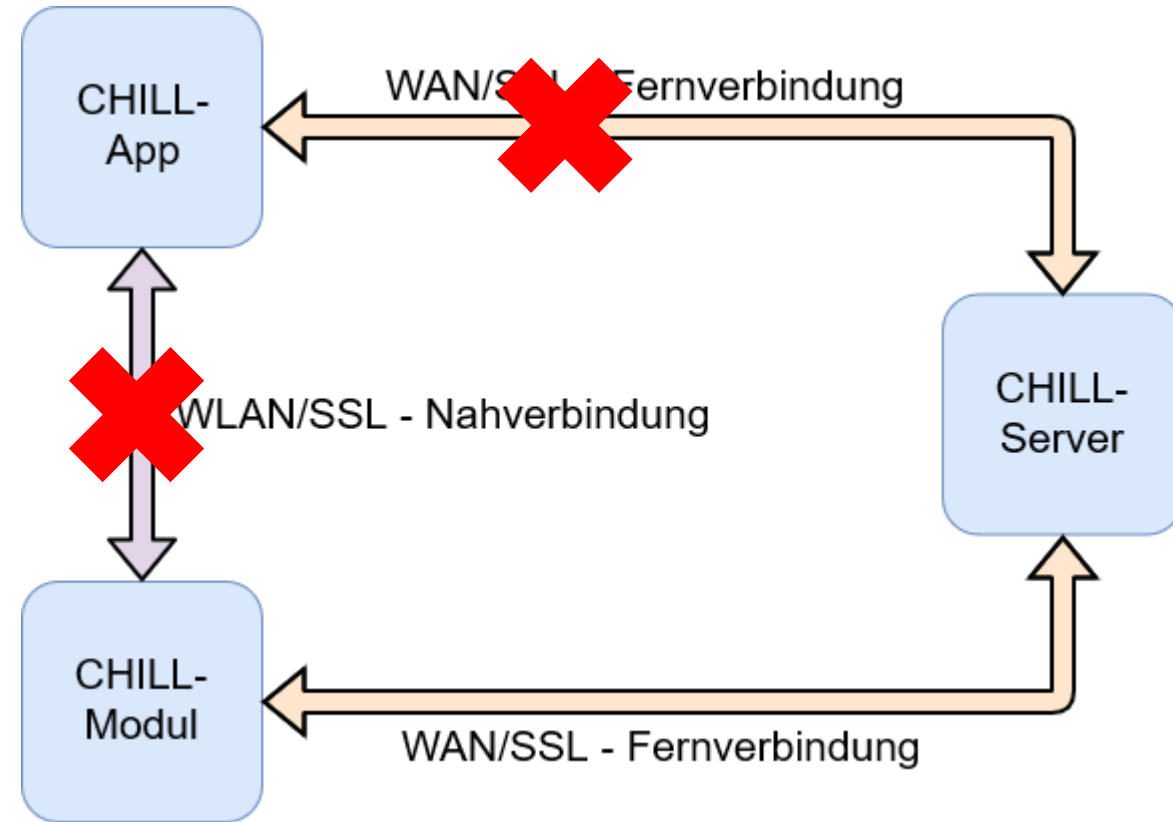
- App- und ModuleAcceptor werden bei Start des Servers erzeugt
- Bei Verbindungsanfrage wird eine entsprechende Session erstellt
- Nach Schließen der Verbindung wird die Session beendet



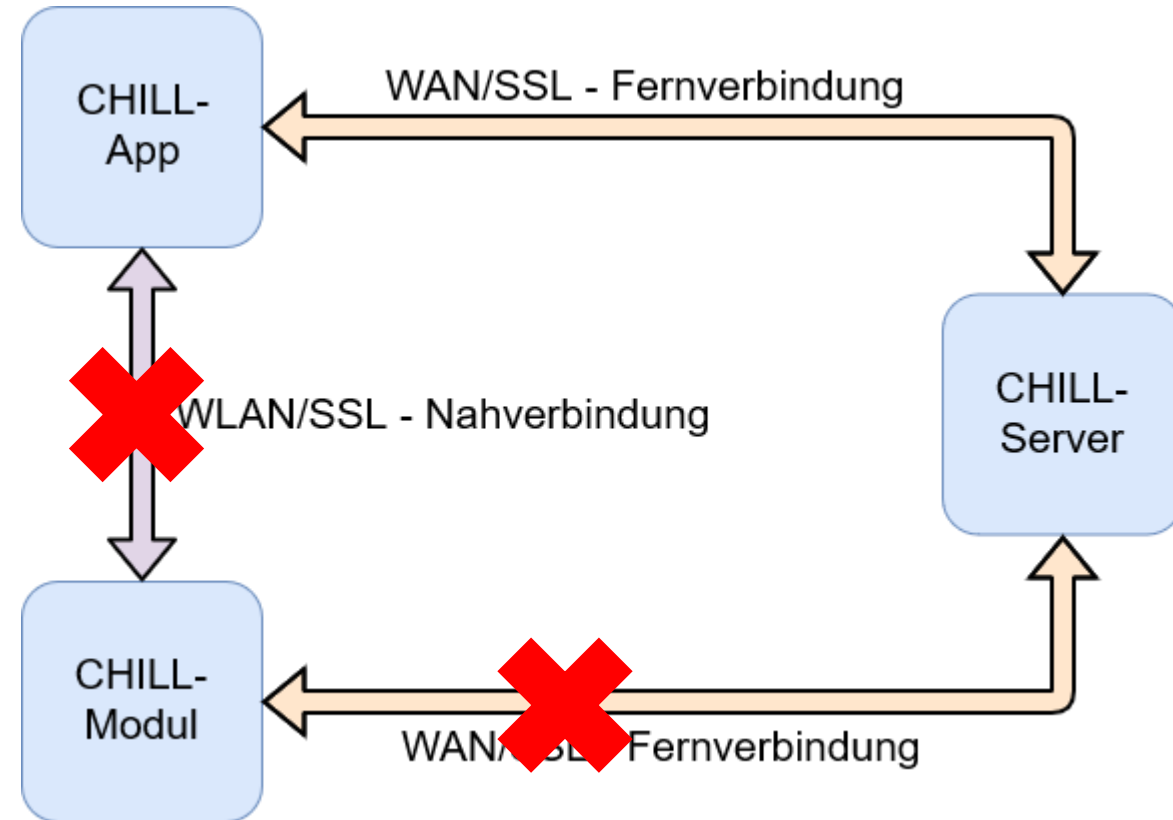
# Aufgaben des CHILL-Servers



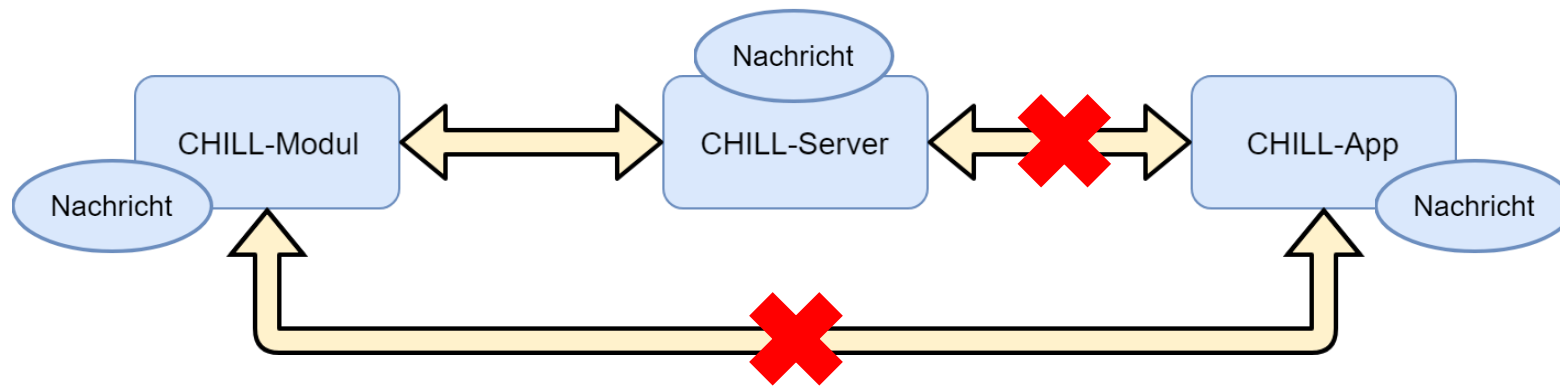
- Registrierung und Login
- Verbindungsglied
- **Zwischenspeicherung von Aufträgen**
- Externe Daten
- OTA-Updates



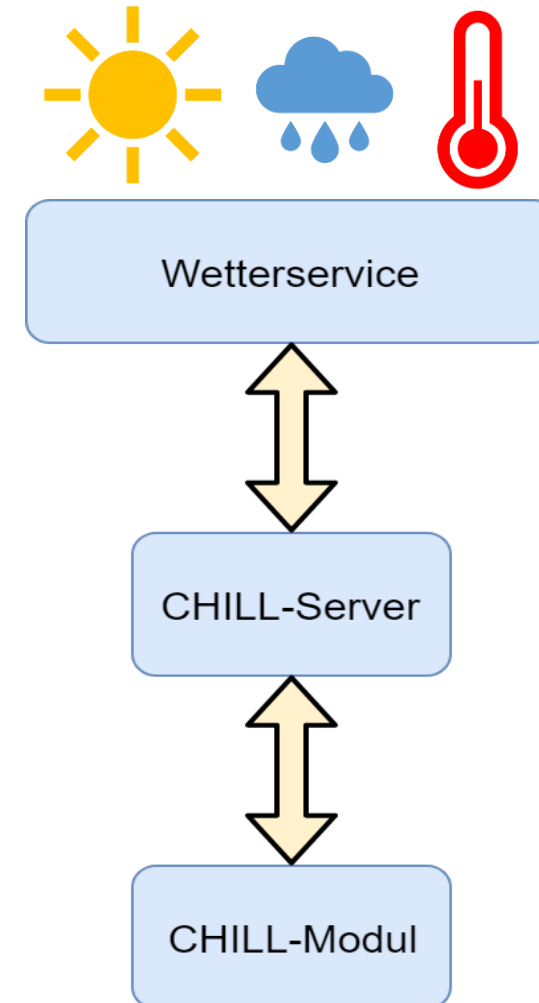
- Registrierung und Login
- Verbindungsglied
- **Zwischenspeicherung von Aufträgen**
- Externe Daten
- OTA-Updates



- Besteht keine Verbindung zum Empfänger der gerade erhaltenen Nachricht, wird diese in einem Buffer gespeichert
- Ist der Empfänger das nächste Mal verbunden, erhält er die gespeicherte Nachricht, welche dann aus dem Speicher entfernt wird

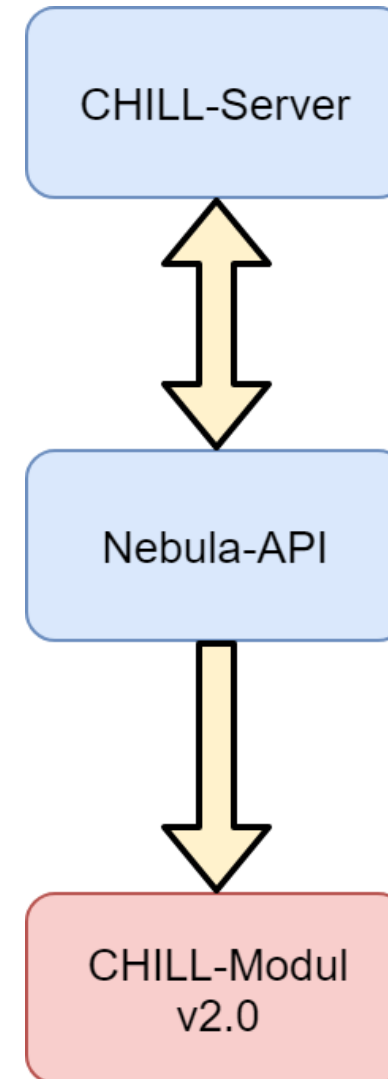


- Registrierung und Login
- Verbindungsglied
- Zwischenspeicherung von Aufträgen
- **Externe Daten**
- OTA-Updates



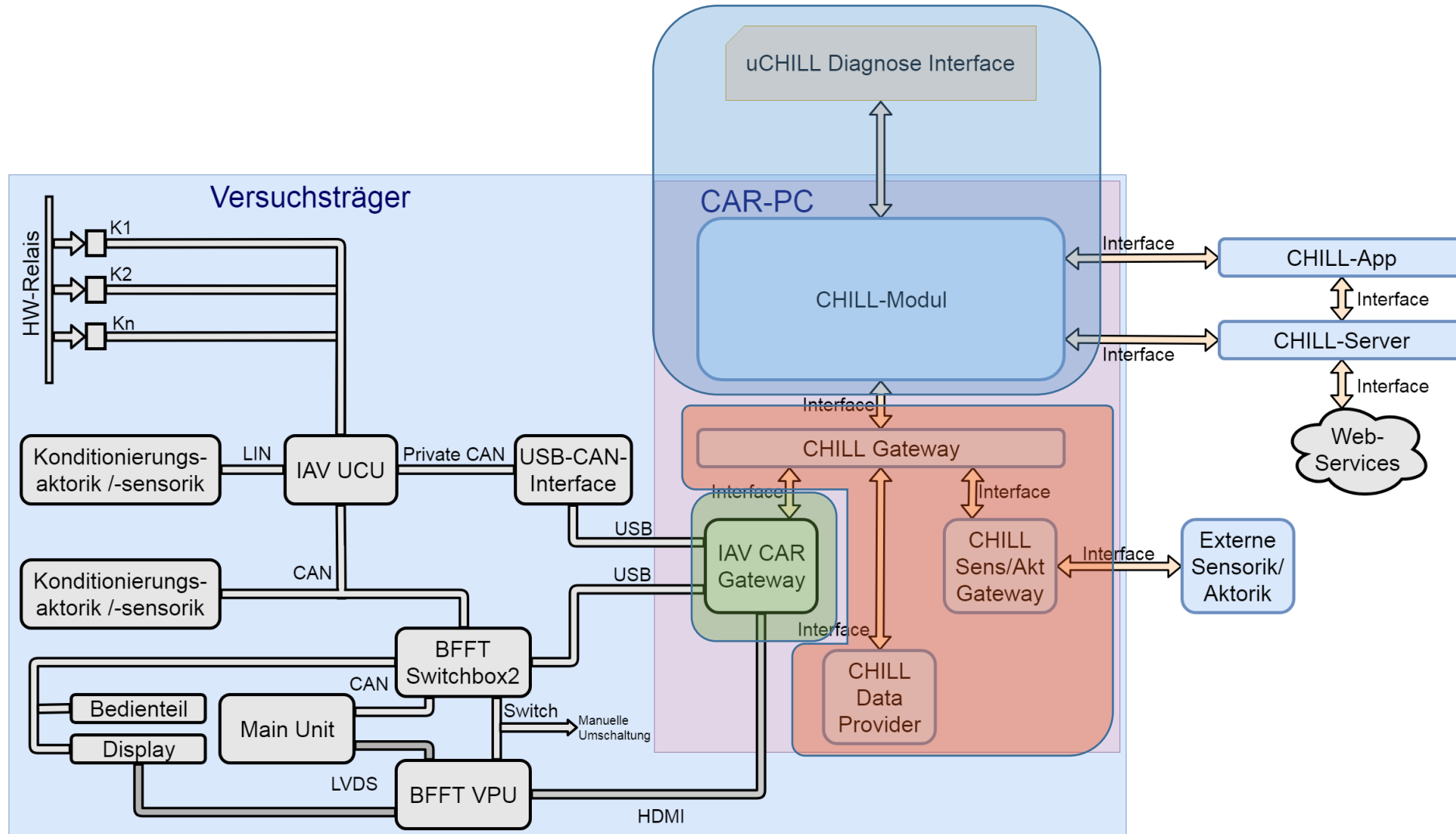
- Modulanfrage nach aktuellem Wetter für den momentanen Standort des Fahrzeugs
- HTTP-Anfrage
- Koordinaten an die API von OpenWeatherMap
- Antwort mit aktuellen Wetterdaten
  - Temperatur
  - Niederschlag
  - Windgeschwindigkeit
  - Luftfeuchte

- Registrierung und Login
- Verbindungsglied
- Zwischenspeicherung von Aufträgen
- Externe Daten
- **OTA-Updates**

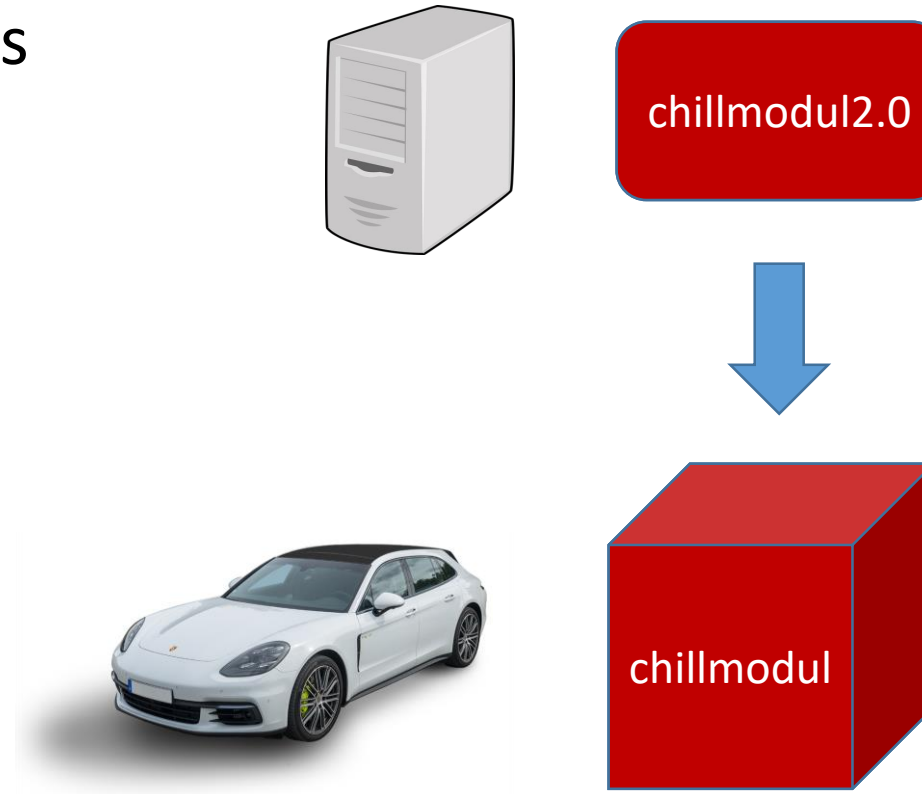




# OTA-Updates: Nebula



- Docker Image des CHILL-Moduls wird als Container gestartet
  - Nebulactl oder „CreateApp“
- Aktualisieren des Images: Neue Softwareversion
- Neustart des Containers, um neue Version aufzuspielen
  - Nebulactl oder „RestartApp“

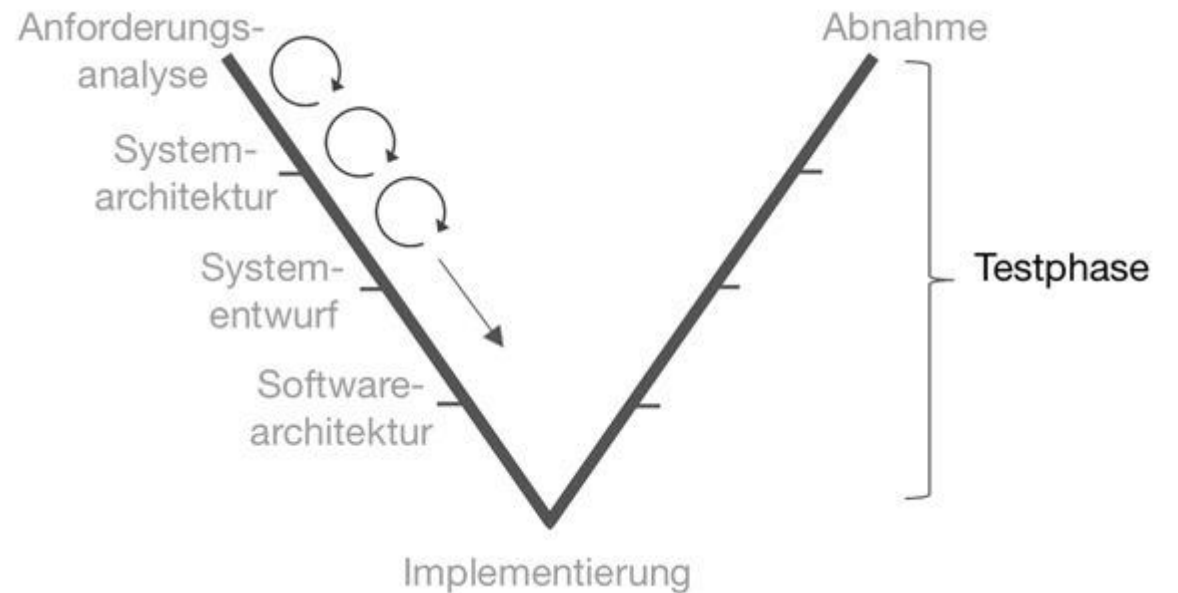


- Abfrage der Softwareversion durch CHILL-App
  - Image name beinhaltet die jeweilige Versionsnummer (z.B. „chillmodul2.3“)
  - „get app config“ liefert die Eigenschaften eines Containers
  - Mit „get app config“ den Namen des Images erhalten, aus diesem die Versionsnummer ziehen

- Bestätigungsanfrage an den Nutzer vor dem OTA-Update
  - Aktualisiere erst, wenn der Nutzer in der CHILL-App dem Update zustimmt
- Statusanfrage an das CHILL-Modul (Fahrzeugzustand)
  - Aktualisiere erst, wenn der Zustand des CHILL-Moduls (und des Fahrzeugs) passend ist
- Automatisches OTA-Update
  - Sobald neue Version des Images vorhanden ist, automatisch OTA-Nachrichten an CHILL-App und CHILL-Modul

# Anforderungsabdeckung & Testmanagement

- Anforderungsabdeckung
- Infrastruktur
- Verwaltung
- Organisation
- Ergebnisse
- Fazit



Priorität	Anforderungen	Erfüllt
Muss	560	493
Soll	186	144
Kann	63	15
Gesamt	809	652
Top-Level	10	10

- 67 nicht erfüllte **muss**-Anforderungen
  - 60 aufgrund nicht im Versuchsträger vorhandener Sensorik / Aktorik
  - Lernen der Fenstereinstellungen
- 42 nicht erfüllte **soll**-Anforderungen
  - Absicherung der Kommunikation
  - Zusätzliche Benachrichtigungsfeatures
  - Senden von Logdateien an den Server
  - Weniger relevante Aktorik / Sensorik
- 48 nicht erfüllte **kann**-Anforderungen
  - Zusatzfunktionen

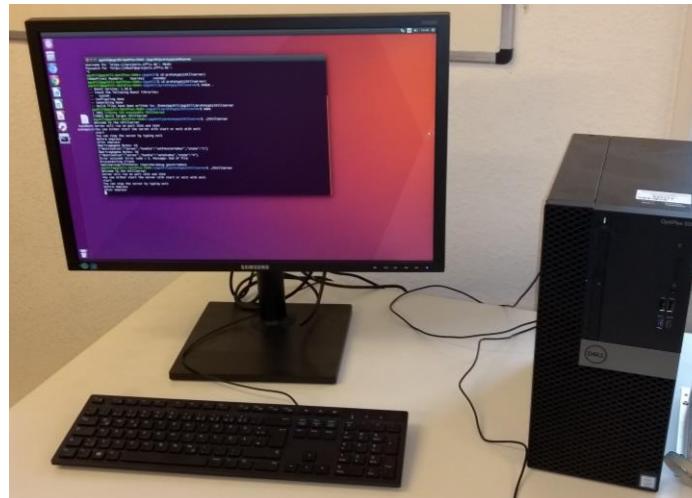
- ✓ Fertigstellen der Dokumentation
- ✓ Fertigstellen von Architektur und Schnittstellen
- ✓ Verfeinern von Konzepten
- ✓ Safetyanalyse erweitern
- ✓ Erfüllung der Anforderungen
  - Prototyp erstellen
  - Vorbereitung auf Veranstaltungen
  - Ausblick auf Erweiterungen



- CarPC
  - Im Gruppenraum
  - Im Versuchsträger



- Server im Gruppenraum
- Testdatenbank



Testfall ID	Bezeichnung	Anforderung	Testtyp	Version	Verantwortlich	Deadline	Ergebnisse
AppComm-0	App Communication Tests		M	0.1			0
AppComm-0.05	➔ SendToModuleFirst		Komponente	0.2	Johann Loose	2018-11-23	1
AppComm-0.1	➔ Erfolgreiches Empfangen von Nachrichten vom Server		Komponente	0.1	Johann Loose	2018-11-23	1
AppComm-0.2	➔ Senden von Nachrichten an den Server		Komponente	0.1	Wiebke Marx	2018-11-23	1
AppComm-0.3	➔ Erfolgreiches Aufrufen des Konstruktors		Komponente	0.5	Wiebke Marx	2019-04-02	1
AppComm-0.4	➔ getHandle Methode		Komponente	0.4	Johann Loose	2019-01-04	1
AppComm-1	Outgoing Messages		Komponente	0.1	Johann Loose	2019-04-02	0
AppComm-1.01	➔ SendAMCarStatusRequest		Komponente	0.2	Johann Loose	2019-04-02	1
AppComm-1.02	➔ SendAMCarOutsideRequest		Komponente	0.2	Johann Loose	2019-04-02	1
AppComm-1.03	➔ SendAMJobQueueRequest		Komponente	0.2	Johann Loose	2019-04-02	1
AppComm-1.04	➔ SendAMJobStatusRequest		Komponente	0.2	Johann Loose	2019-04-02	1
AppComm-1.05	➔ SendAMLearnDataRequest		Komponente	0.2	Johann Loose	2019-04-02	1
AppComm-1.06	➔ SendAMVacationStatusGetRequest		Komponente	0.4	Johann Loose	2019-04-02	1
AppComm-1.07	➔ SendASOTAVersionRequest		Komponente	0.3	Johann Loose	2019-04-02	1
AppComm-1.08.1	➔ SendAMSendJobPlain - kein korrekt		Komponente	0.2	Johann Loose	2019-04-02	1
AppComm-1.08.2	➔ SendAMSendJobPlain - Sendeauftruf		Komponente	0.2	Johann Loose	2019-04-02	1
AppComm-1.09.1	➔ SendAMSendJob - kein korrekt		Komponente	0.3	Johann Loose	2019-04-02	1
AppComm-1.10	➔ SendAMSetToleranceRequest		Komponente	0.2	Johann Loose	2019-04-02	1
AppComm-1.11	➔ SendAMVacationStatusSetRequest		Komponente	0.1	Johann Loose	2019-04-02	1
AppComm-1.12	➔ SendASLoginRequest		Komponente	0.2	Johann Loose	2019-04-02	1
AppComm-1.13	➔ SendASPasswordChangeRequest		Komponente	0.1	Johann Loose	2019-04-02	1
AppComm-1.14	➔ SendASRegistrationRequest		Komponente	0.1	Johann Loose	2019-04-02	1
AppComm-1.15	➔ SendASOTAUpdateResponse		Komponente	0.1	Johann Loose	2019-04-02	1

- C++ : CHILL-Modul & CHILL-Server

- BoostTest
- GoogleMock



googletest  
Google C++ Testing Framework

- C# : CHILL-App

- MSTest
- Moq



[https://de.wikipedia.org/wiki/Boost\\_\(C%2B%2B-Bibliothek\)](https://de.wikipedia.org/wiki/Boost_(C%2B%2B-Bibliothek))  
<https://prdeving.wordpress.com/2017/03/28/unit-test-in-c-with-google-test/>  
<https://venturebeat.com/2017/03/07/microsoft-launches-visual-studio-2017/>  
<https://github.com/moq>



**Testfall:** Learning-0.1.1 -- Learning::addDataPoint() daten hinzufügen

**Testfalltyp:** Komponente

**Anforderung:**

**UseCase:**

**Bezug zu:**



**Testobjekt(e):** Learning

**Verantwortlich:** Raphael Boomgaarden

**Deadline:** 2019-04-04



**Vorbedingungen:** -Ein CarStatus-Objekt anlegen  
-Das CarStatus-Objekt mit Werten in outsideTemperature, globalInsideTemp, humidity, currentAmbientColor.red, currentAmbientColor.green, currentAmbientColor.blue, steeringHeatingStatus, seatHeatingLevelFL und seatVentilationLevelFL versehen  
-Eine AppID ausdenken



**Auslösendes Ereignis:** Learning::addDataPoint mit dem CarStatus-Objekt und der AppID aufrufen



**Erwartetes Ergebnis:** -Eine Datei mit dem Namen train\_data.csv wurde angelegt

-Eine Datei mit dem Namen position.txt wurde angelegt

-Der Inhalt der Datei train\_data.csv entspricht den vorher eingestellten Daten in kategorialer form

-Wird addDataPoint mehrfach aufgerufen, wird der neue Datenpunkt ans Ende eingefügt. Wenn die Datei 1000 Zeilen hat, wird die Datei von vorne überschrieben

## Ergebnis für Testfall: Learning::addDataPoint() daten hinzufügen, Version: 0.2

**Tester**

- Lynn Liebert
- Mario Meinen
- Moritz Buhr
- Patrick Schuster
- Raphael Boomgaarden
- Stefan Puch
- Wiebke Marx

**Erfolgreich:** Ja ▼

**Ergebnis:** Erwartetes Ergebnis ist eingetreten.

Menü			
<a href="#">Testfälle</a>			
Berichte			
<a href="#">Testbericht (LaTeX)</a>			
<a href="#">Download als ZIP</a>			
Statusübersicht			
Typ	#	✓	✗
Gesamt	293	227	
Komponente	246	189	0
Komponentenintegration	29	27	0
Teilsystem	0	0	0
Kommunikation	15	10	0
Teilsystemintegration	1	1	0
System	2	0	0
Abnahme	0	0	0
Legende			
	Ergebnisse anzeigen		
	Gültiges Ergebnis vorhanden		
	Kein gültiges Ergebnis vorhanden		
	Kein Testergebnis vorhanden		
	Eintrag bearbeiten		
	Neuen Untertestfall anlegen		
	Neues Testergebnis eintragen		
	Testfall rekursiv löschen		
	Version erhöhen		

Testfall ID	Bezeichnung	Anforderung	Testtyp	Version	Verantwortlich	Deadline	Ergebnisse	Neuen Testfall anlegen
AppComm-0	<a href="#">App Communication Tests</a>		Komponente	0.2	Johann Loose	2018-11-23	1	
AppComm-0.05	➔ <a href="#">SendToModuleFirst</a>		Komponente	0.2	Johann Loose	2019-04-02	1	
AppComm-0.1	➔ <a href="#">Erfolgreiches Empfangen von Nachrichten vom Server</a>		Komponente	0.1	Johann Loose	2018-11-23	1	
AppComm-0.2	➔ <a href="#">Senden von Nachrichten an den Server</a>		Komponente	0.1	Wiebke Marx	2018-11-23	1	
AppComm-0.3	➔ <a href="#">Erfolgreiches Aufrufen des Konstruktors</a>		Komponente	0.5	Wiebke Marx	2019-04-02	1	
AppComm-0.4	➔ <a href="#">getHandle Methode</a>		Komponente	0.4	Johann Loose	2019-01-04	1	
AppComm-1	<a href="#">Outgoing Messages</a>		Komponente	0.1	Johann Loose	2019-04-02	1	
AppComm-1.01	➔ <a href="#">SendAMCarStatusRequest</a>		Komponente	0.2	Johann Loose	2019-04-02	1	
AppComm-1.02	➔ <a href="#">SendAMIsOutsideInsideRequest</a>		Komponente	0.2	Johann Loose	2019-04-02	1	
AppComm-1.03	➔ <a href="#">SendAMJobQueueRequest</a>		Komponente	0.2	Johann Loose	2019-04-02	1	
AppComm-1.04	➔ <a href="#">SendAMJobStatusRequest</a>		Komponente	0.2	Johann Loose	2019-04-02	1	
AppComm-1.05	➔ <a href="#">SendAMLearntDataRequest</a>		Komponente	0.2	Johann Loose	2019-04-02	1	
AppComm-1.06	➔ <a href="#">SendAMVacationStatusGetRequest</a>		Komponente	0.4	Johann Loose	2019-04-02	1	



Implementieren der  
Funktion, Definieren  
von Testfällen

Entwickler: 14 Tage



Schreiben der  
Testimplementierung,  
Ausführen und  
Kommentieren

Tester: 4 Tage



Überarbeiten der  
Implementierung,  
Erneutes Ausführen  
der Tests

Entwickler: 3 Tage

Typ	Testfälle	Erfolgreich
Komponente	246	208
Komponentenintegration	29	28
Kommunikation	15	13
Teilsystemintegration	1	1
System	6	6
<b>Gesamt</b>	<b>296</b>	<b>256</b>

- ✗ Startschwierigkeiten bei Durchführung der Tests
- ✗ Verschiebung der Testphase nach hinten

- ✓ Gute Umsetzung von Kreuztests
- ✓ Nutzung der Testdatenbank
- ✓ Hohe Quote erfolgreicher Tests
- ✓ Vollständige Abdeckung bei Systemtests



# Live-Demonstration

- ✓ Fertigstellen der Dokumentation
- ✓ Fertigstellen von Architektur und Schnittstellen
- ✓ Verfeinern von Konzepten
- ✓ Safetyanalyse erweitern
- ✓ Erfüllung der Anforderungen
- ✓ Prototyp erstellen
  - Vorbereitung auf Veranstaltungen
  - Ausblick auf Erweiterungen

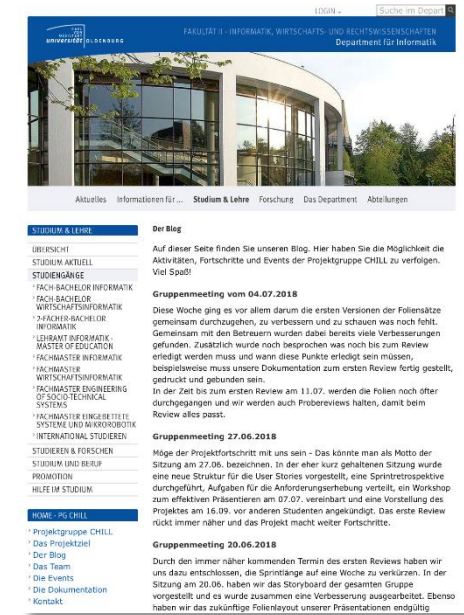
# Abschluss

- Öffentlichkeitsarbeit
- Fazit
- Ausblick

# Öffentlichkeitsarbeit - Übersicht



- Webseite
  - Informieren über die PG
  - Dokumente bereitstellen
- Versuchsträger
  - Interesse wecken
  - Zugehörigkeit zeigen
- Konferenzen
  - Austausch mit anderen
  - Neue Impulse bekommen



- Automotive Software Kongress Landshut
  - Paper (10 Seiten)
  - Vortrag (30 Minuten)
- Studierendenkonferenz an der Ruhr-Universität
  - Paper (12 Seiten)
  - Vortrag (20 Minuten)
- IdeenExpo Hannover
  - 15. – 23.06.2019
  - Präsentation des Versuchsträgers



3. STUDENTISCHE KONFERENZ



- ✓ Fertigstellen der Dokumentation
- ✓ Fertigstellen von Architektur und Schnittstellen
- ✓ Verfeinern von Konzepten
- ✓ Safetyanalyse erweitern
- ✓ Erfüllung der Anforderungen
- ✓ Prototyp erstellen
- ✓ Vorbereitung auf Veranstaltungen
  - Ausblick auf Erweiterungen

	Zielerreichung	Projektablauf	Internes Zusammenspiel
Positiv	<ul style="list-style-type: none"> <li>- Das Ziel der Projektgruppe wurde erfüllt</li> <li>- Schwierigkeiten konnten überwunden werden</li> </ul>	<ul style="list-style-type: none"> <li>- Leben eines industriellen Prozesses</li> <li>- Erlernen wichtiger Kompetenzen</li> <li>- Arbeiten entlang der Pläne</li> <li>- Durchgehend motiviert</li> </ul>	<ul style="list-style-type: none"> <li>- Gute Stimmung in der Projektgruppe</li> <li>- Viel gegenseitige Hilfe und Unterstützung durch die Betreuer und die IAV</li> <li>- Entwicklung einer Kritikkultur</li> <li>- Persönliche Weiterentwicklung</li> </ul>
Negativ	<ul style="list-style-type: none"> <li>- Teilweise wurden Sprintziele nicht erreicht</li> </ul>	<ul style="list-style-type: none"> <li>- Teilweise längere Eingewöhnungsphasen</li> </ul>	<ul style="list-style-type: none"> <li>- Teilweise kurzzeitige Spannungen zwischen Projektmitgliedern</li> </ul>



- Weiterentwicklung der Konzepte
- Entwicklung neuer Features
  - Integration weiterer Hardware
    - Außenbeleuchtung einstellen
    - Duft im Innenraum integrieren
  - Verwendung des Multimedia-Displays
    - Integration eines Nutzerinterfaces



- ✓ Arbeiten entlang der Pläne
- ✓ Fertigstellen der Dokumentation
- ✓ Erfüllung der Anforderungen
- ✓ Architektur und Schnittstellen
- ✓ Verfeinern von Konzepten
- ✓ Safetyanalyse erweitern
- ✓ Prototyp erstellen
- ✓ Vorbereitung auf Veranstaltungen
- ✓ Ausblick auf Erweiterungen

# Vielen Dank für die Aufmerksamkeit



[www.uni-oldenburg.de/chill](http://www.uni-oldenburg.de/chill)

# Fragen?