



Template

2021-02-22

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

© We acknowledge that this document uses material from the arc42 architecture template, <http://www.arc42.de>. Created by Dr. Peter Hruschka & Dr. Gernot Starke.



Inhalt

Einführung und Ziele.....	3
Aufgabenstellung.....	4
Qualitätsziele.....	5
Stakeholder.....	5
Randbedingungen	6
Technische Bedingungen.....	6
Organisatorische Bedingungen	6
Kontextabgrenzung	7
Fachlicher Kontext	7
Technischer Kontext.....	8
Lösungsstrategie.....	9
Bausteinsicht.....	Fehler! Textmarke nicht definiert.
Ebene 1	10
Ebene 2	11
Laufzeitsicht	12
<i>Buchen einer Fahrt</i>	12
<i>Bezahlen einer Fahrt</i>	13
Whitebox Gesamtsystem	14
Verteilungssicht	15
Infrastruktur Ebene 1	15
Qualitätsmerkmale.....	15
Zuordnung von Bausteinen zur Infrastruktur.....	15
Infrastruktur Ebene 2	16
Querschnittliche Konzepte	16
<i>Entwicklungskonzept</i>	16
<i>Design- & Architekturkonzept</i>	16
<i>Sicherheitskonzept</i>	16
Entwurfsentscheidungen	17
Qualitätsanforderungen.....	17
Risiken und technische Schulden.....	18
Glossar	Fehler! Textmarke nicht definiert.

Einführung und Ziele

In einer zunehmend vernetzten Welt ist Mobilität mehr als nur Transport. CarFlux wurde geschaffen, um den Herausforderungen urbaner Mobilität mit einer technisch fortschrittlichen Plattform zu begegnen.

Unsere App integriert innovative Technologien wie präzise Geolokalisierungsdienste, fortschrittliche Datenübermittlung für die Fahrzeugüberwachung und sichere Authentifizierungssysteme, um eine reibungslose und sichere Nutzererfahrung zu gewährleisten. Durch die Nutzung von GPS und Echtzeitdaten optimieren wir nicht nur die Fahrzeugverfügbarkeit, sondern ermöglichen es auch den Nutzern, ihre Fahrten genau zu verfolgen und die schnellste, aber auch umweltschonendste Route zu wählen.

Bei CarFlux setzen wir auf eine skalierbare Infrastruktur, um eine hohe Leistungsfähigkeit und schnelle Reaktionszeiten sicherzustellen, selbst bei starkem Nutzerverkehr. Die Integration von Push-Benachrichtigungen sorgt dafür, dass unsere Nutzer stets auf dem Laufenden sind, sei es für Buchungsbestätigungen oder Standortaktualisierungen.

Unsere Sicherheitsarchitektur gewährleistet nicht nur eine sichere Authentifizierung, sondern auch die Verschlüsselung sensibler Daten, um die Privatsphäre unserer Nutzer zu schützen. Mit einem klaren Fokus auf Skalierbarkeit und Flexibilität ist CarFlux nicht nur eine App – sie ist eine fortschrittliche Lösung, die sich den Anforderungen der modernen Mobilität anpasst.

Bequemlichkeit – Sicherheit – Nachhaltigkeit – CarFlux

Aufgabenstellung

Um die Bedürfnisse der Benutzer nahtlos zu erfüllen und im selben Moment für Sicherheit, Nachhaltigkeit und Effizienz zu sorgen, muss die App folgende Aufgaben erfüllen.

Aufgabenstellung	Beschreibung
Benutzerfreundliche Buchung	Implementierung einer Buchungsfunktion, die es unseren Benutzern ermöglicht, Fahrzeuge schnell und einfach zu reservieren und/oder zu buchen.
Fahrzeugverfügbarkeit und -ortung	Wir stellen Echtzeitinformationen über verfügbare Fahrzeuge sowie deren genaue Standorte bereit, um eine effiziente Nutzung zu gewährleisten.
Sichere Authentifizierung und Identifikation	Implementierung sichere Authentifizierungsmechanismen für Fahrer und Passagiere, um die Identität zu gewährleisten und die Sicherheit zu erhöhen.
Zahlungsabwicklung	Implementierung einer sicheren und benutzerfreundlichen Zahlungsabwicklung für Fahrten, die verschiedene Zahlungsmethoden unterstützt.
Umweltaspekte	Erstellung einer Funktion, um den CO2-Fußabdruck zu berechnen und umweltfreundliche Fahroptionen zu fördern.
Analysen und Berichterstattung	Verwendung von Analysewerkzeugen, um Daten über Fahrgewohnheiten, Beliebtheit von Strecken und andere relevante Informationen zu sammeln.
Sicherheitsmerkmale	Implementieren Sie Sicherheitsfunktionen wie Notruftknöpfe, um im Notfall schnelle Unterstützung zu ermöglichen.
Bewertungs- und Rückmeldesystem	Bewertungs- und Rückmeldesystem für Fahrer und Passagiere ermöglichen, um die Qualität der Dienstleistung zu gewährleisten und Vertrauen aufzubauen.

Qualitätsziele

Qualitätsziel	Kriterien
Benutzerfreundlichkeit	Benutzer sollten in der Lage sein einfach, eine Fahrt innerhalb von 3 Minuten oder weniger zu buchen.
Sicherheit	Unterdurchschnittlich gemeldete Sicherheitsverletzungen im Jahr.
Zuverlässigkeit	Eine Verfügbarkeit der App von mindestens 99,9 % im Jahr.
Skalierbarkeit	Skalierbarkeit, um einen Anstieg von mindestens 20 % an gleichzeitigen Benutzern während Stoßzeiten zu bewältigen.
Leistung	Ladezeiten sollten innerhalb von 3 Sekunden für gängige Benutzerprofile liegen.

Stakeholder

Rolle	Kontakt	Erwartungshaltung
Investor (John Flix)	<i>John.flix@flix-u-fertig.at</i>	Erwartet sich eine Rendite aus dieser Investition.
Marketing Experte (Gina Linetti)	<i>Gina.Linetti@ak.group</i>	Präzise Anforderung der Zielgruppe und der Alleinstellungsmerkmale.
Legal (Rosa Diaz)	<i>Rosa.Diaz@ak.group</i>	Einhaltung der rechtlichen Bestimmungen. (DSGVO, Sicherheitsstandard)
Business Development Manager (Raymond Holt)	<i>Raymond.Holt@ak.group</i>	Fokus auf den Wachstum und die geschäftlichen Erfolge der Anwendung.
Sustainability (Jacob Peralta)	<i>Jacob.Peralta@ak.group</i>	Überprüfung der Maßnahmen, die für die Umwelt getroffen werden können
User Experience (Terry Jeffords)	<i>Terry.Jeffords@ak.group</i>	Sorgt für die benutzerfreundliche Oberfläche der Applikation.

Randbedingungen

Technische Bedingungen

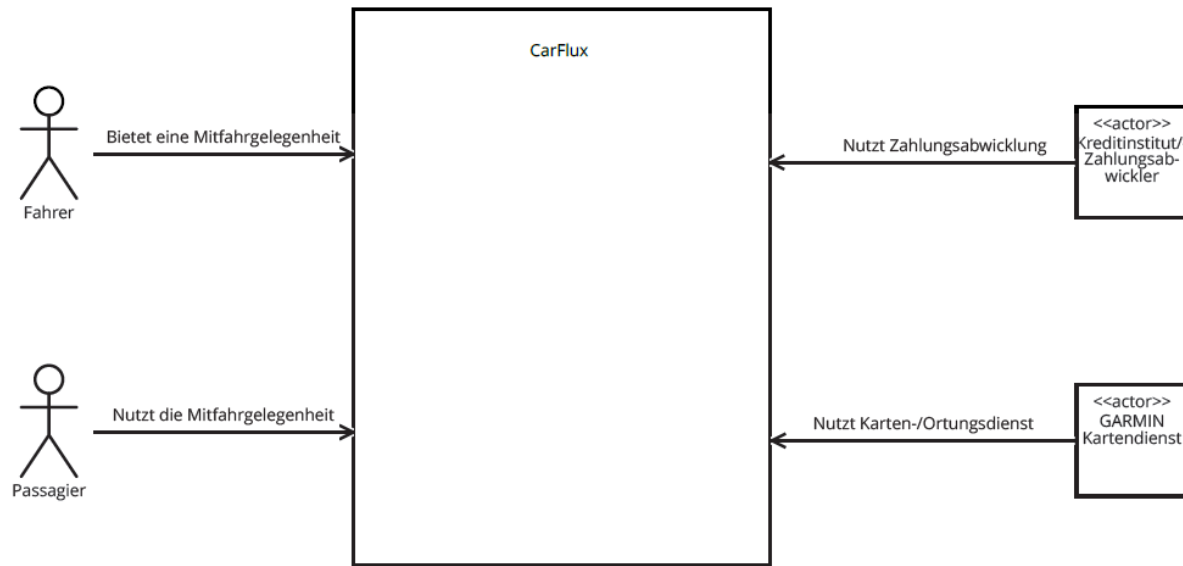
ID	Bedingung	Hintergründe & Motivation
1	Sichere Authentifizierung & Autorisierung	Implementierung robuster Mechanismen für die sichere Authentifizierung von Fahrern und Passagieren sowie klare Autorisierungsregeln.
2	Zahlungsintegration	Integration von sicheren Zahlungsgateways für reibungslose und sichere Transaktionen.
3	Geolokalisierungsdienste	Integration von präzisen Geolokalisierungsdiensten für die genaue Standortbestimmung von Fahrern und Fahrzeugen.
4	Plattformunabhängigkeit	Die App sollte auf den gängigen mobilen Plattformen wie iOS und Android lauffähig sein, um eine breite Nutzerbasis zu erreichen.
5	Fehlerprotokollierung und Überwachung	Implementierung von Mechanismen zur Fehlerprotokollierung und kontinuierlichen Überwachung der App-Performance.

Organisatorische Bedingungen

ID	Bedingung	Hintergründe & Motivation
1	Finanzielle Ressourcen	Sicherstellung ausreichender finanzieller Ressourcen für die Entwicklung, Wartung und Skalierung der App.
2	Rechtliche Compliance	Einhaltung aller rechtlichen Anforderungen und Vorschriften für den Betrieb von CarFlux in verschiedenen Regionen.
3	Risikomanagement	Entwicklung eines Risikomanagementplans, um potenzielle Risiken zu identifizieren, zu bewerten und proaktiv anzugehen.

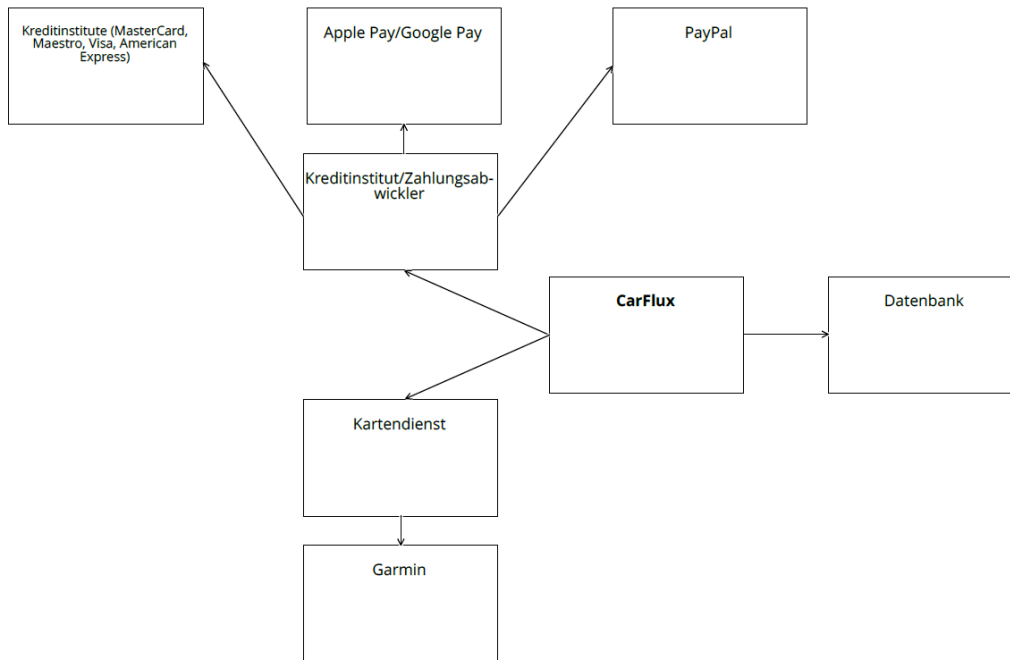
Kontextabgrenzung

Fachlicher Kontext



Nutzer/Actor	Beschreibung	Input	Output
Fahrer	Bietet eine Mitfahrgelegenheit	Der Fahrerstandort wird durchgehend aktualisiert. Der Fahrer kann Fahrten akzeptieren oder ablehnen.	Sobald ein Passagier eine Fahrt bestellt, wird dem nächsten gelegenen Fahrer dieser Auftrag zugewiesen und er kann ihn durchführen.
Passagier	Sucht bzw. nutzt eine Mitfahrgelegenheit	Der Passagier gibt in der App seinen Standort und die Zieladresse ein. Ihm wird innerhalb weniger Minuten ein Fahrer zugewiesen.	Die App liefert innerhalb weniger Minuten eine Mitfahrgelegenheit und den dazugehörigen Preis.
Kreditinstitut/ Zahlungsabwickler	Abwicklung der Zahlungen	Die App versendet nach abgeschlossener Fahrt die Bankdaten inkl. Rechnungsbetrag des Passagiers an das Kreditinstitut/ Zahlungsabwickler.	Das Kreditinstitut/ Zahlungsabwickler verarbeitet die übermittelten Daten und bestätigt die Transaktion.
Kartendienst	Stellt Karten-/Ortungsdienste zur Verfügung	Die App stellt die Karten-/Ortungsdienste von Garmin zur Standort- und Zielermittlung zur Verfügung.	Die genauen Standorte und die Fahrtstrecke werden vom Kartendienst an die App zurückgeschickt.

Technischer Kontext



Komponente	Beschreibung
CarFlux	Die Anwendung, über der die Fahrer und Passagiere miteinander interagieren.
Datenbank	Speichert Benutzerdaten wie z. B. Zahlungsdetails, Personendaten, Bereits absolvierte Fahrten und Bewertungen.
Kreditinstitut/Zahlungsabwickler	Führt die Zahlung anhand der erhaltenen Daten durch.
Kartendienst	Stellt Karten- und Ortungsdienste zur Verfügung.
Kreditinstitute	Externe Zahlungsdienstleister
Apple Pay/Google Pay	Externe Zahlungsdienstleister
PayPal	Externer Zahlungsdienstleister
Garmin	Externer Dienstleister für Karten- und Ortungsdienste

Lösungsstrategie

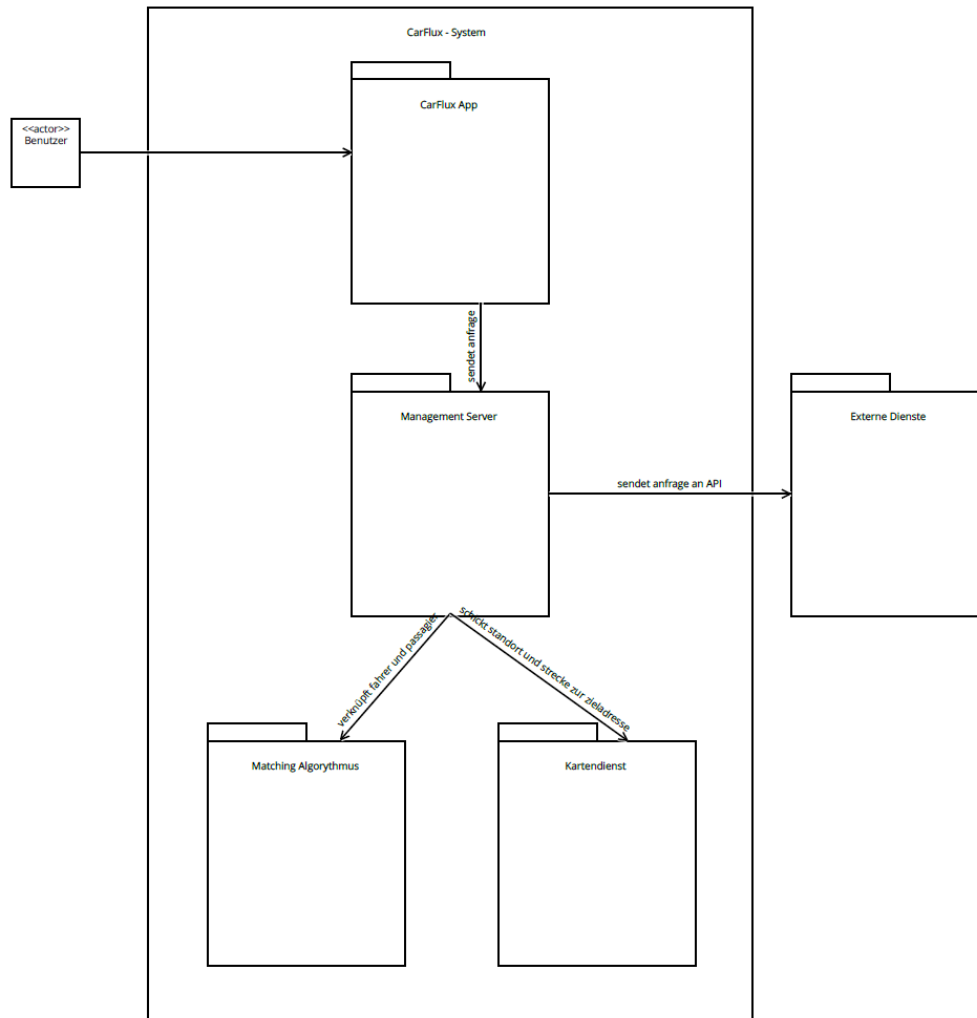
Problem	Lösung
Security	JSON Web Tokens (JWTs) ermöglichen zustandslose Authentifizierung und sparen Serverressourcen, während SSL/TLS eine sichere Übertragung von JWTs gewährleistet, indem es eine verschlüsselte Verbindung zwischen Client und Server herstellt. Diese Kombination bietet eine effiziente und sichere Authentifizierungslösung.
Integration	Unsere Applikation integriert APIs von GARMIN, Paypal, Apple Pay und verschiedenen Kreditinstituten, um die Grundfunktionen Zahlung und Lokalisation effektiv zu unterstützen.
Datenbankmanagement	Aufgrund seiner weitverbreiteten Nutzung wird PostgreSQL bevorzugt, wenn es um Datenbanklösungen geht.
Frontend	Vue.js Native wird als Frontend-Framework eingesetzt, um sowohl Android als auch iOS effektiv zu unterstützen.
Backend	Node.js eignet sich besonders gut für Anwendungen mit zahlreichen gleichzeitigen Verbindungen und Echtzeit-Anforderungen.

Unsere CarFlux Plattform basiert auf einer Microservices-Architektur, die sich als optimale Wahl erwiesen hat. Diese Aufteilung in unabhängige, hochspezialisierte Dienste ermöglicht höhere Flexibilität, Skalierbarkeit und Wartbarkeit. Die klare Abgrenzung der Funktionalitäten fördert Ressourcenoptimierung, verbessert die Fehlertoleranz und ermöglicht eine agile Entwicklung. Diese moderne Architektur ermöglicht eine skalierbare Plattform, die sich schnell an sich ändernde Anforderungen anpassen kann.

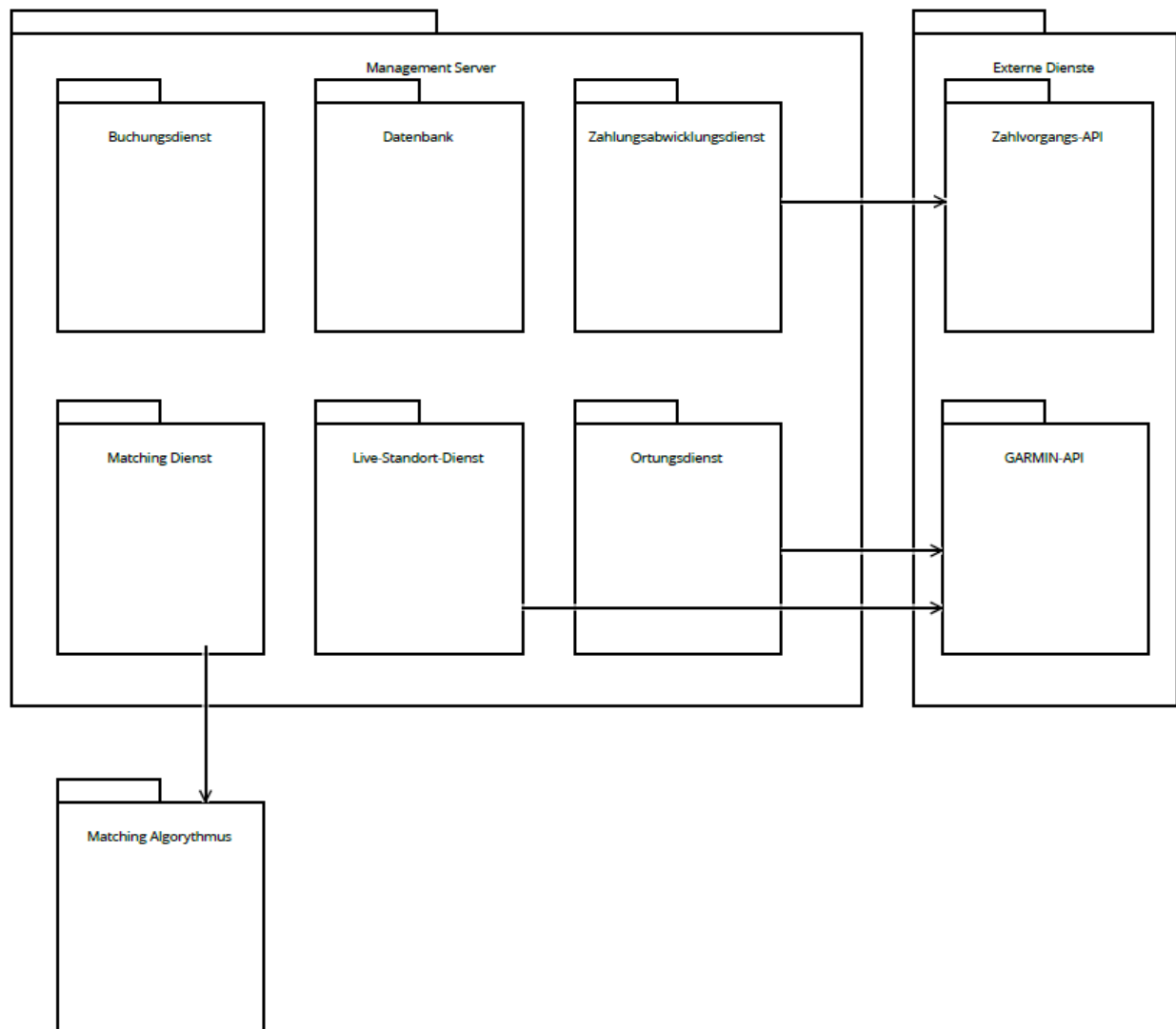
Building Block View

- Ebene 1: Bietet eine Gesamtansicht des Systems.
- Ebene 2: Vertieft die Betrachtung ausgewählter Bausteine der 1. Ebene.

Ebene 1

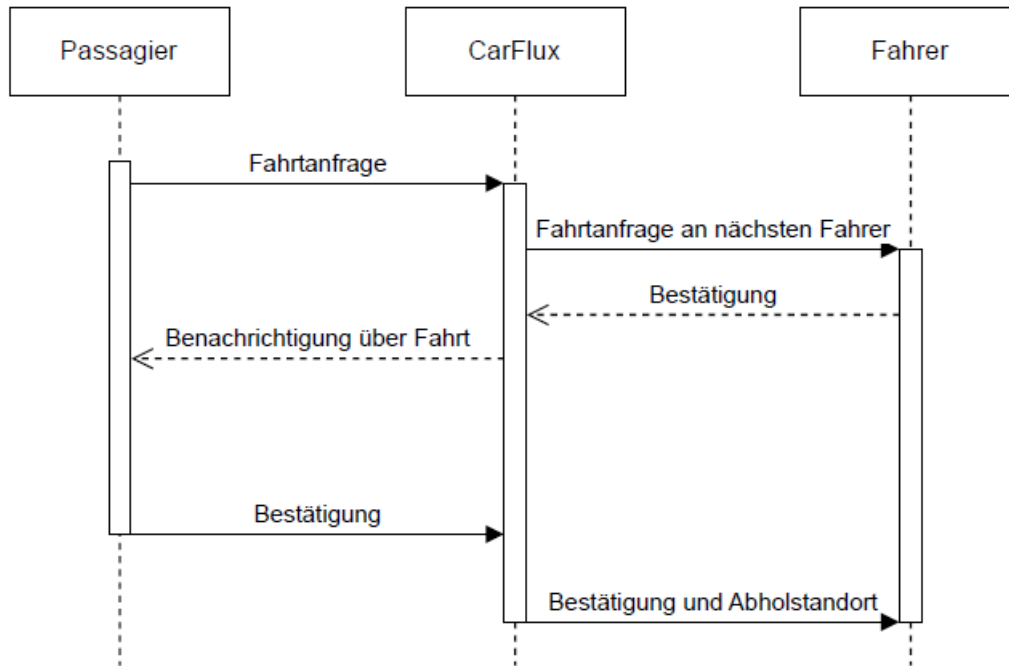


Ebene 2



Laufzeitsicht

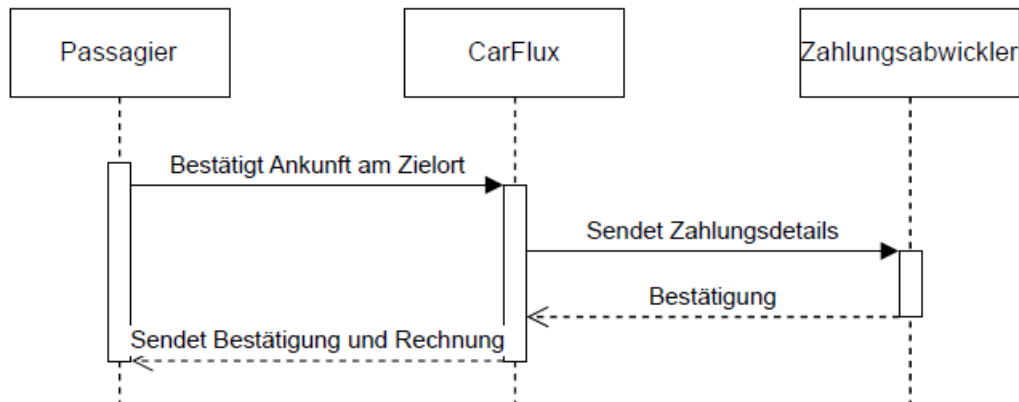
Buchen einer Fahrt



Das Buchen einer Fahrt bei CarFlux erfolgt in der Regel über die mobile Anwendung des Unternehmens. Hierbei durchläuft der Nutzer folgende Schritte:

- **Standortauswahl:** Nach erfolgreicher Anmeldung ermittelt die App automatisch den aktuellen Standort des Nutzers über GPS. Alternativ kann der Nutzer auch manuell einen Abholort eingeben.
- **Zieleingabe:** Anschließend gibt der Nutzer das gewünschte Ziel ein. Die App schätzt daraufhin die voraussichtliche Fahrtdauer und den Fahrtpreis.
- **Fahrtbestätigung:** Nach der Zuweisung erhält der Nutzer eine Zusammenfassung der Fahrt, einschließlich geschätzter Ankunftszeit und Fahrtpreis. Er kann die Buchung durch Bestätigung abschließen.

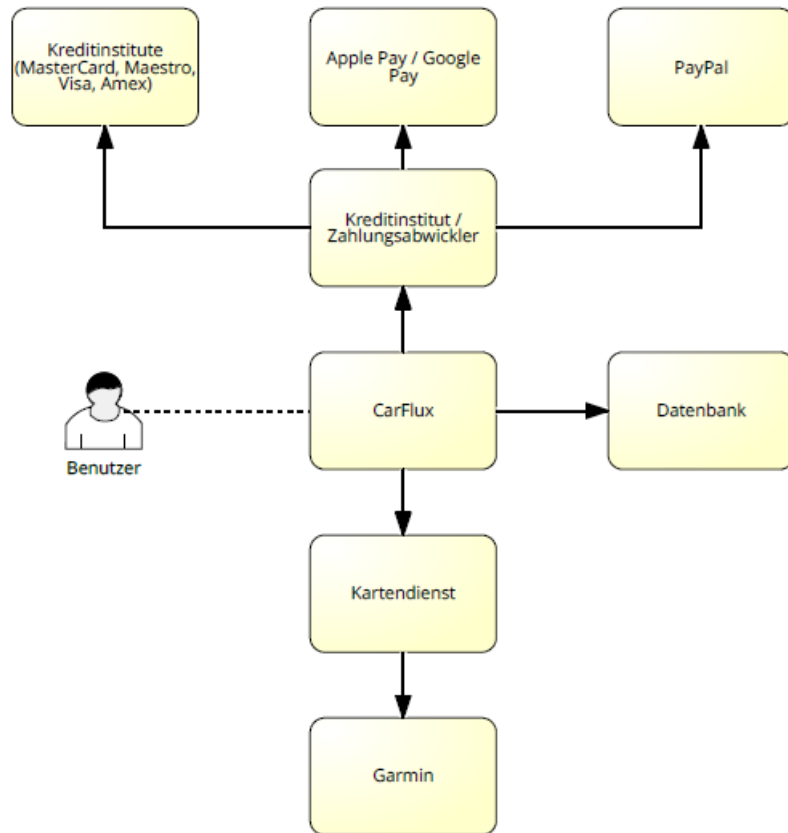
Bezahlen einer Fahrt



Das Bezahlen einer Fahrt bei CarFlux erfolgt ausschließlich auf elektronischem Wege und ist nahtlos in die mobile Anwendung integriert. Hier sind die Schritte, die typischerweise bei der Bezahlung einer CarFlux-Fahrt befolgt werden:

- **Automatische Abrechnung:** CarFlux verwendet eine automatische Abrechnungsmethode, bei der die Fahrtkosten automatisch über das hinterlegte Zahlungsmittel abgebucht werden. Dieses Zahlungsmittel wird während der Registrierung in der App vom Benutzer hinterlegt.
- **Fahrabschluss:** Nachdem die Fahrt abgeschlossen ist, bestätigt der Fahrgast diese und die automatische Berechnung der Fahrkosten erfolgt auf Grundlage der zurückgelegten Strecke und der Dauer der Fahrt.
- **Quittung und Fahrtübersicht:** Der Nutzer erhält unmittelbar nach Fahrtende eine elektronische Quittung über die in der App angegebene E-Mail-Adresse. Diese Quittung enthält detaillierte Informationen zur Fahrt, einschließlich Fahrtroute, Dauer und Fahrtpreis.

Whitebox Gesamtsystem



Durch die Segmentierung des CarFlux-Systems in verschiedene essenzielle Module wird eine effiziente Verwaltung der vielfältigen App-Funktionen ermöglicht. Diese Struktur fördert die Autonomie der einzelnen Module und erleichtert sowohl die Wartung als auch mögliche zukünftige Erweiterungen des Systems.

Verteilungssicht

Server-Hardware: 2x Dell R740XD 24 SFF mit folgenden Daten

- Prozessor: Intel Xeon Gold 6226R 2.9 GHz
- Arbeitsspeicher: 128 GB DDR5-SDRAM
- Speicher: 2x 4 TB NVMe SSD

Betriebssystem: Windows Server Standard 2022

Datenbank: PostgreSQL

Infrastruktur Ebene 1

Hat eine verteilte Infrastruktur, um eine robuste und skalierbare Lösung zu gewährleisten. Die Verteilung ermöglicht eine effiziente Lastenverteilung und verbessert zeitgleich die Ausfallsicherheit des Systems. Die Auswahl von Rechner und Prozessoren erfolgt somit auf der Grundlage, ob sie den Anforderungen der App standhalten können.

Qualitätsmerkmale

Hohe Verfügbarkeit: Die verteilte Struktur stärkt die Widerstandsfähigkeit der App gegenüber Ausfällen einzelner Komponenten oder Standorte.

Skalierbarkeit: Die Infrastruktur erlaubt eine nahtlose Skalierung, um steigende Benutzerzahlen effektiv zu bewältigen.

Schnelligkeit: Die Auswahl leistungstarker Hardware und geeigneter Prozessoren gewährleistet eine effiziente und somit schnelle Ausführung der App.

Zuordnung von Bausteinen zur Infrastruktur

Die Datenbank ist auf einem Server gespeichert und wird von dort aus verwaltet.

Der Zahlungsdienstleister und der Kartendienstleister sind über das Internet mit den entsprechenden externen Diensten verbunden.

Infrastruktur Ebene 2

Die zweite Ebene unserer Infrastruktur konzentriert sich auf die nahtlose Integration externer Dienste in die Produktionsumgebung.

Zahlungsabwicklungs-Schnittstelle:

Die Zahlungsabwicklungs-Schnittstelle ermöglicht sichere Zahlungstransaktionen. Regelmäßige Überwachung und umfassende Logging-Mechanismen gewährleisten eine präzise Fehlerprävention.

Garmin-Schnittstelle:

Die Schnittstelle integriert nahtlos Kartendaten und Standortdienste. Durch regelmäßige Tests und aktive Überwachung wird eine zuverlässige Funktion sichergestellt.

Monitoring und Fehlerprävention:

Regelmäßige Audits und Penetrationstests identifizieren potenzielle Schwachstellen und sorgen für deren Behebung.

Performance-Tests gewährleisten eine effiziente Funktion auch bei steigenden Nutzerzahlen.

Automatisierte Systeme überwachen Protokolle auf Unregelmäßigkeiten.

Diese Maßnahmen garantieren, dass die CarFlux-App störungsfrei mit externen Diensten interagiert und den Nutzern eine zuverlässige Erfahrung bietet.

Querschnittliche Konzepte

Entwicklungskonzept

Fahrvermittlungsalgorithmus:

Ein robuster Algorithmus, um Fahrer mit Fahrgästen basierend auf Faktoren wie Standort, Ziel und Fahrtpräferenzen abzugleichen ist essenziell. Wir würden mit einer dynamischen Preisgestaltung, basierend auf Angebot und Nachfrage, arbeiten.

Design- & Architekturkonzept

Microservices-Architektur:

Zerlegt die Anwendung in kleine, unabhängige Dienste, die separat entwickelt und bereitgestellt werden können. Das verbessert die Skalierbarkeit, Wartbarkeit und Fehlerisolierung.

Sicherheitskonzept

In-App-Kommunikation:

Erleichtert eine sichere Kommunikation zwischen Fahrer und Passagier innerhalb der App. Daten wie Telefonnummern bleiben weiterhin anonymisiert, da die Privatsphäre geschützt bleiben soll.

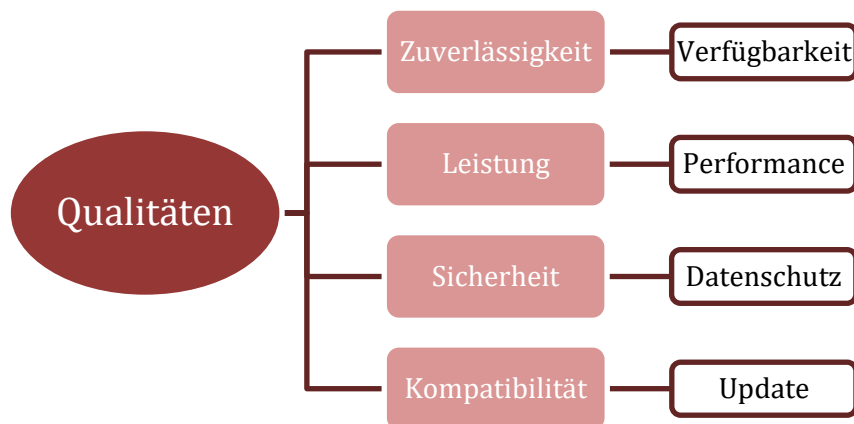
Entwurfsentscheidungen

Unser Entwurf von CarFlux basiert auf einer Microservices-Architektur, die Skalierbarkeit, Flexibilität und Wartbarkeit gewährleistet. Jeder Microservice repräsentiert eine klar definierte Funktion und kann unabhängig entwickelt, bereitgestellt und skaliert werden. Der Technologiestack umfasst Frameworks wie Spring Boot, Vue.js, PostgreSQL-Datenbanken, , Orchestrierung durch Kubernetes, ein API-Gateway für den Zugriff auf Microservices und Sicherheitsmechanismen.

Die Vorteile dieser Entwurfsentscheidung liegen in der klaren Trennung der Funktionen in Microservices, durch welche die Wartbarkeit erleichtert wird. Die Integration von Message-Broker-Systemen verbessert die Zuverlässigkeit der Kommunikation zwischen den Microservices.

Qualitätsanforderungen

Qualitätsbaum



Qualitätsszenarien

	Ziel	Quelle
Zuverlässigkeit	Eine maximale Ausfallzeit von 0,1 % pro Monat erreichen.	Die App sollte eine Verfügbarkeit von 99,9 % aufrechterhalten.
Leistung	Eine Ladezeit von unter 3 Sekunden für 95 % der Benutzer aufrechterhalten.	Die App sollte durchschnittlich innerhalb von 3 Sekunden für Benutzer mit einer Standard-Internetverbindung geladen werden.
Sicherheit	Gewährleisten, dass Benutzerdaten geschützt sind und keine gemeldeten Sicherheitsverletzungen vorliegen.	Implementierung sicherer Authentifizierung und Datenverschlüsselung, um unbefugten Zugriff zu verhindern.
Kompatibilität	Die App regelmäßig aktualisieren, um die Kompatibilität mit den neuesten Betriebssystemversionen aufrechtzuerhalten.	Die App sollte mit den neuesten Versionen der wichtigsten mobilen Betriebssysteme kompatibel sein.

Risiken und technische Schulden

Sicherheit	Unzureichende Maßnahmen
Fehlerhandling	Mangelnde Fehlerbehandlung
Leistungsrisiken	aufgrund schlechter Performance vom Nutzer abgelehnt