



2024-01-14

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.

Einführung und Ziele

Aufgabenstellung

RideShare App

- Einfacher und unkomplizierter Transport
- Kostengünstiger als Marktkonkurrenz
- Emissionseinsparung durch Fahrgemeinschaften
- Echtzeitüberweisungen

Qualitätsziele

Priorität	Quality	Motivation
1	Security	Daten von Kunden sind uns sehr wichtig und sollen sicher bleiben
2	Interoperability	App soll auf allen gängigen Modellen laufen (iOS/Android)
3	Performance Efficiency	Gleichzeitige Verwendung & Echtzeitüberweisungen möglich

Stakeholder

Rolle/Name	Erwartungen
Investor /Maryam Patel	TBD
Business Development / Raj Gupta	TBD
UX Designer / Megan Chen	Die App soll für alle Kunden leicht verständlich und zugänglich sein.
Fahrer / Amirah Rahman	TBD
Umweltaktivist / Javier Gomez	Gründung von Fahrgemeinschaften wäre ein Vorteil für die ökologischen Fußabdruck. Die Fahrzeuge müssen gewisse Anforderungen erfüllen, um Teil der Flotte zu werden.
Investor / John	Bequemlichkeit der App und niedrigere Kosten für unsere Nutzer sowie eine hohe Kapitalrendite, ein starkes Team, einzigartige Funktionen und ein Wertversprechen

Randbedingungen

Problem	Considered Alternatives	Decision
Implementierung für viele Betriebssysteme	Java, Kotlin, Swift	IOS & Android Implementierung
Authentifizierung von Usern und Fahrern	Drittanbieter Software kaufen Eigene App entwickeln	Wir werden aus Kostengründen eine Drittanbieter Software zukaufen
Große Daten Mengen bzw. sensible Daten speichern	In der Cloud speichern Lokal speichern	Wir werden die Daten in der AWS Cloud speichern

Risk/Technical Debt	Description
KYC	Eigens entwickelte Identifizierungslösungen entsprechen nicht den regulatorischen Anforderungen
AML	Transaktionen in gewisser Höhe müssen überwacht werden bzw. unterliegen strengen Kontrollen/Regulierungen
Cloud	Kosten in Zukunft unklar
Dependency on external services	AI Integration
High costs of CDN (Content Delivery Network) servers	Kosten für CDN Server sind sehr hoch
Lack of Flutter developers	Entwickler für Flutter nicht ausreichend verfügbar

Kontextabgrenzung

Constraints	Background and/or motivation
Einhaltung von Vorschriften	Die Anwendung muss sich an regionale Vorschriften halten. Dies gewährleistet die Legalität und den reibungslosen Betrieb in allen Regionen und fördert das Vertrauen der Benutzer.
Plattformübergreifende Kompatibilität	Um eine breitere Benutzerbasis zu erreichen, muss die Anwendung sowohl auf iOS als auch auf Android nahtlos funktionieren, wodurch eine Konsistenz in Erlebnis und Funktionen gewährleistet wird.
Zahlungssicherheit (PCI DSS-Konformität)	Bei finanziellen Transaktionen ist eine sichere Handhabung von Benutzerzahlungsdaten von größter Bedeutung. Die Einhaltung der PCI DSS-Standards hilft, das Vertrauen der Benutzer zu erhalten und Betrug zu verhindern.

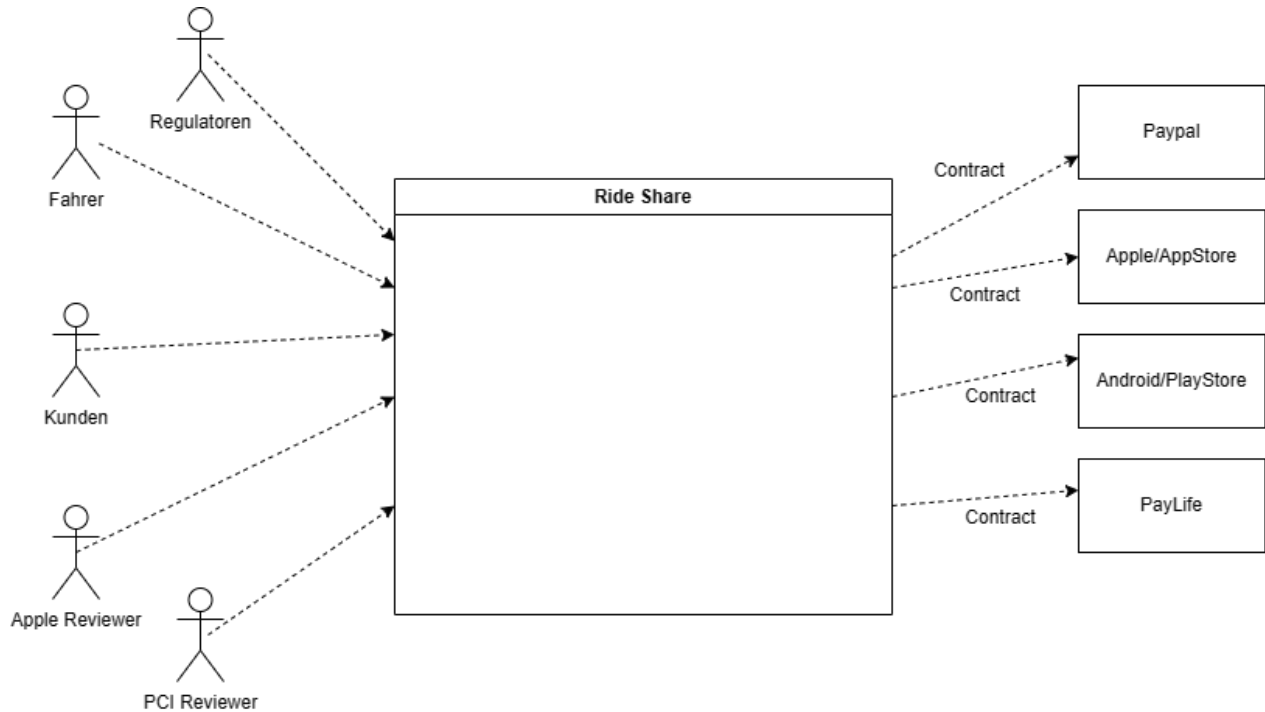
Ressourcen

Angeichts der Natur digitaler Anwendungen ist es entscheidend, die Ressource der Architektur so zu gestalten, dass ein mögliches Wachstum der Benutzerbasis und des Verkehrs bewältigen kann und so ein konstantes Benutzererlebnis gewährleistet.

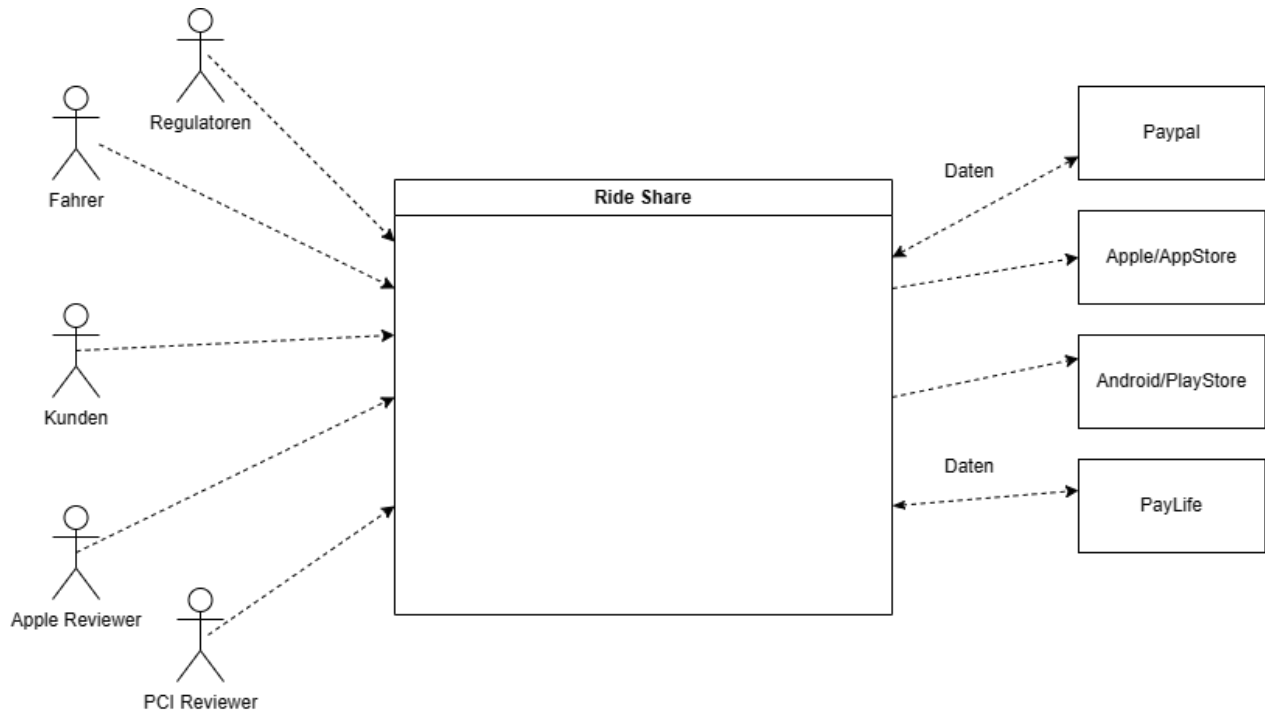
Verwendung von standardisierten Protokollen für die Integration

Für alle externen Integrationen sollten standardisierte Protokolle wie HTTPS und OAuth verwendet werden. Dies gewährleistet nicht nur die Sicherheit, sondern vereinfacht auch den Prozess der Integration mit Drittsystemen.

Fachlicher Kontext



Technischer Kontext

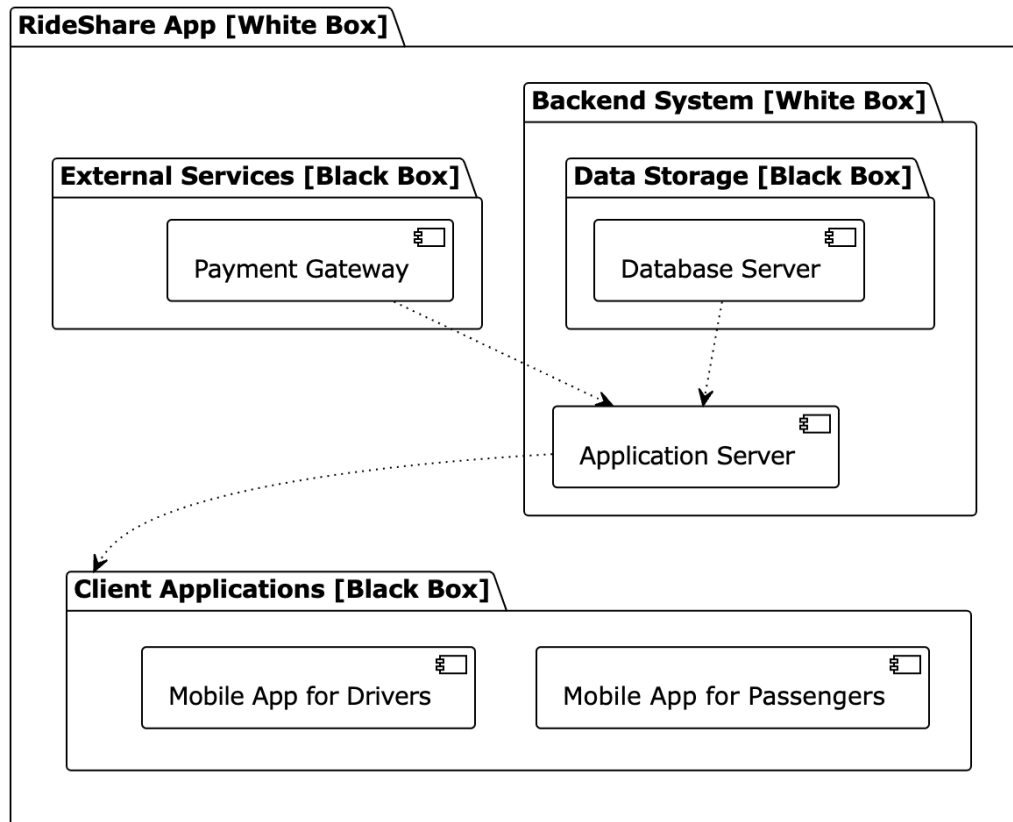


Lösungsstrategie

Goal/Requirements	Architectural Approach
Leichte Einbindung von Zahlungsmethoden	Einbindung von modularen Zahlungsmöglichkeiten (Paypal, Kreditkarte, Bankeinzug(eps))
Wartbarkeit (Fehlerbehebung und Anpassungen)	Code soll leicht anpassbar sein und einzelne Elemente sollen schnell verändert können
Applikation sicher und skalierbar	SOA - App soll Daten redundant und sicher abspeichern

Bausteinsicht

Whitebox Gesamtsystem



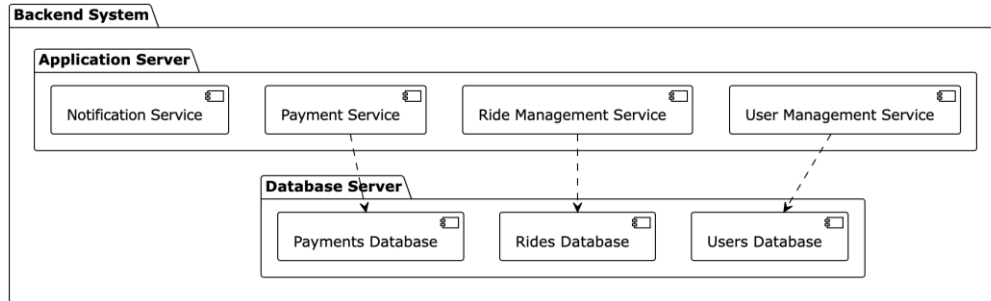
Enthaltene Bausteine

- External Services: Alle externen Verbindungen zu Drittanbieter, wie Apple/Google, Paylife, etc.
- Data storage: Datenbank für Kunden und Zahlungsinformationen
- Client Application: App am Smartphone des Fahrers/Kunden

Wichtige Schnittstellen

- Payment: Schnittstelle für die Zahlungsabwicklung
- Application-Database Server: Schnittstelle zwischen Applikationsserver und Datenbank
- Client Application- Application Server: Schnittstelle zwischen App und Applikationsserver

Ebene 2



Whitebox <Baustein 1>

Notification Service

Whitebox <Baustein 2>

Payment Service

Whitebox <Baustein 3>

Ride Management Service

Whitebox <Baustein 4>

User Management Service

Whitebox <Baustein 5>

Payments Database

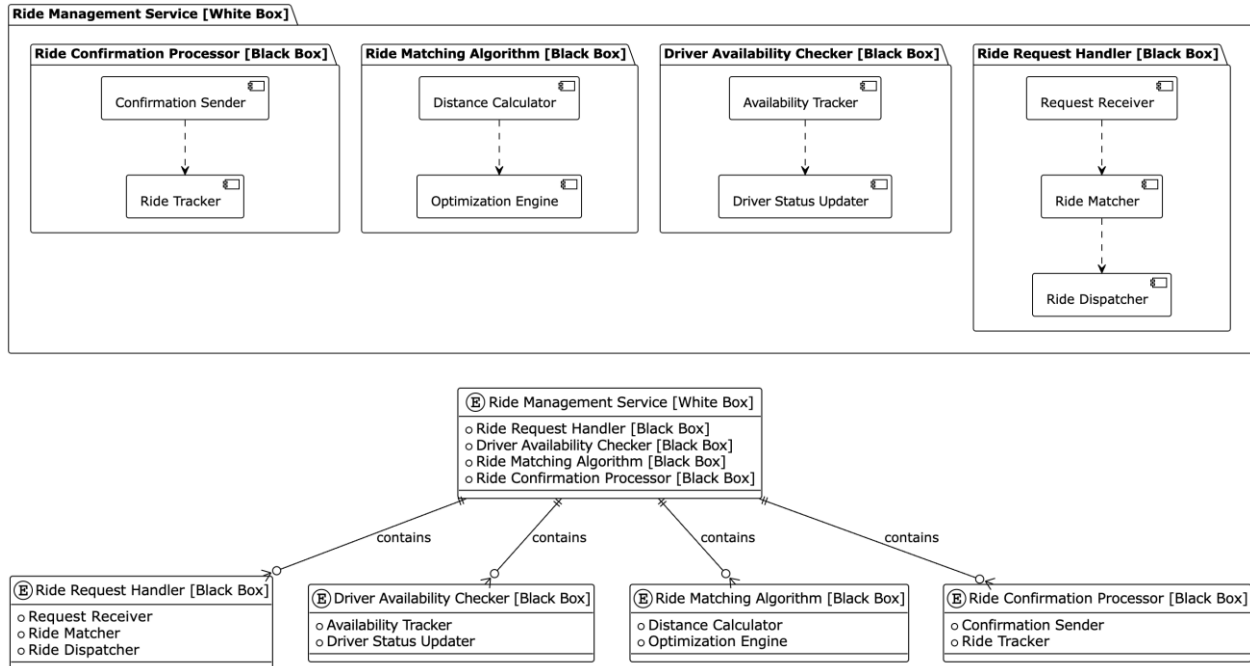
Whitebox <Baustein 6>

Rides Database

Whitebox <Baustein 7>

Users Database

Ebene 3



Whitebox <Baustein 1.1>

Ride Confirmation Processor

Whitebox <Baustein 1.2>

Ride Matching Algorithm

Whitebox <Baustein 1.3>

Driver Availability Checker

Whitebox <Baustein 1.4>

Ride Request Handler

Laufzeitsicht

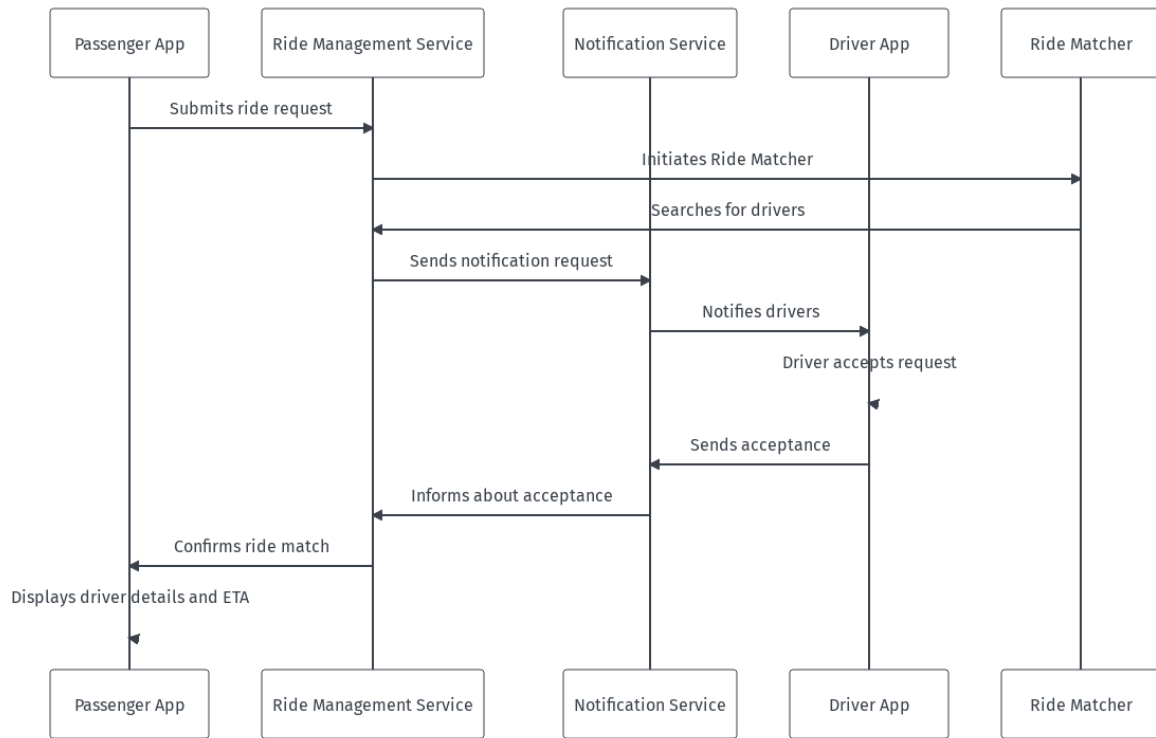


Figure 1- This diagram illustrates the sequence of events from a passenger requesting a ride to a driver accepting the request

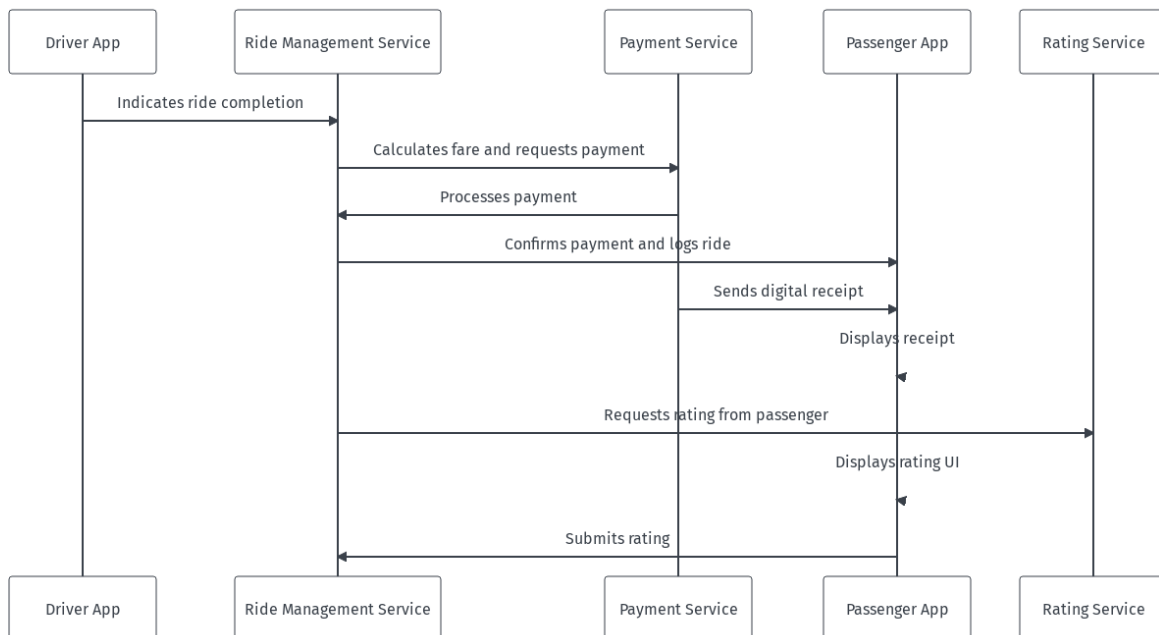
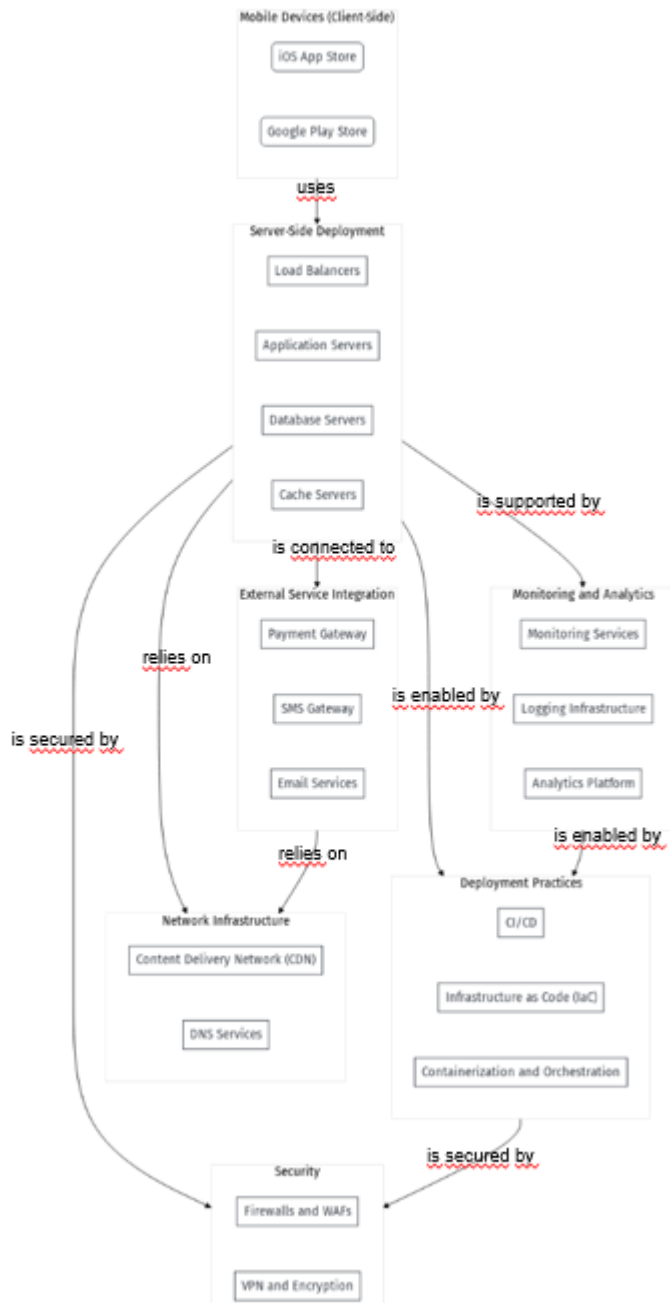


Figure 2 - This diagram shows the steps involved from the completion of a ride to the processing of the payment and rating submission

Verteilungssicht



Querschnittliche Konzepte

Development concepts

- Kontinuierliche Integration/Kontinuierliche Bereitstellung (CI/CD), um schnelle Iterationen und schnelle Bereitstellung von Funktionen zu gewährleisten

Architecture and design patterns

- Service-Orientierte Architektur (SOA) zur besseren Modularität und Erweiterbarkeit. Jedes Modul (wie Zahlung, Bewertung, Preisgestaltung) wird als ein eigenständiger Service dargestellt, der lose mit anderen Diensten gekoppelt ist
- Verwendung des Beobachtermusters für Echtzeitbenachrichtigungen an Benutzer über den Fahrstatus

Safety and security concepts

- Zwei-Faktor-Authentifizierung für Fahrer und Fahrgäste zur Verbesserung der Kontosicherheit
- Datenverschlüsselung für alle persönlichen Daten und Zahlungsinformationen, um Datenschutz und Schutz vor Datenverletzungen zu gewährleisten

Entwurfsentscheidungen

Context	Decision	Consequences
Um die Sicherheit der Benutzer zu gewährleisten und Vertrauen in das System zu schaffen	Wir werden ein umfassendes Bewertungssystem für Fahrer und Fahrgäste implementieren	Erhöht das Vertrauen der Benutzer, erfordert jedoch möglicherweise Moderation, um falsche Meldungen oder Vorurteile zu behandeln
Um den Zahlungsprozess zu vereinfachen und sichere Transaktionen zu gewährleisten	Wir werden uns mit einem bekannten Drittanbieter-Zahlungsprozessor integrieren	Vereinfacht die Zahlung, macht die App jedoch abhängig von der Verfügbarkeit von Drittanbieterdiensten
Um in dicht besiedelten städtischen Gebieten einen wettbewerbsfähigen Service anzubieten	Wir werden die Entwicklung eines dynamischen Preismodells nach Nachfrage priorisieren	Kann die Verfügbarkeit von Fahrern zu Stoßzeiten optimieren, könnte jedoch Kritik ausgesetzt sein, wenn die Preise zu stark ansteigen

Qualitätsanforderungen

Qualitätsbaum

- **Funktionalität**
 - Vollständigkeit der Funktionen
 - Richtigkeit der Operationen
 - Sicherheit der Transaktionen
- **Zuverlässigkeit**
 - Verfügbarkeit (Betriebszeit, Failover-Mechanismen)
 - Fehlertoleranz (Fehlerbehandlung, Wiederherstellungsverfahren)
- **Benutzerfreundlichkeit**
 - Erlernbarkeit (Intuitivität der Benutzeroberfläche)
 - Effizienz der Nutzung (minimale Schritte für Aufgaben)
 - Zugänglichkeit (Unterstützung für Benutzer mit Behinderungen)
- **Leistung**
 - Reaktionsfähigkeit (Verarbeitungszeit für Transaktionen)
 - Skalierbarkeit (Bewältigung des Wachstums der Benutzeranzahl)
- **Unterstützbarkeit**
 - Wartbarkeit (Einfachheit der Durchführung von Änderungen)
 - Anpassungsfähigkeit (Flexibilität für neue Anforderungen)
 - Testbarkeit (Einfachheit des automatisierten Testens)
 - **Sicherheit**
 - Applikation unterstützt die letzten 3 Android/iOS Versionen
 - Integrität (Code obfuscation)

Qualitätsszenarien

- **Wartbarkeit**
 - **Ziel:** Das System sollte einfache Aktualisierungen und Wartungsarbeiten ermöglichen

- **Szenario:** Wenn ein Entwickler die Benutzeroberfläche aktualisieren muss, sollten die Änderungen innerhalb von drei Stunden umgesetzt werden können, ohne die Systemverfügbarkeit zu beeinträchtigen
- **Leistung**
 - **Ziel:** Die App sollte Transaktionen effizient unter normalen Betriebsbedingungen verarbeiten
 - **Szenario:** Wenn ein Benutzer eine Transaktion initiiert, sollte die durchschnittliche Verarbeitungslatenz zwei Sekunden betragen, mit einem Zielbereich zwischen 1,5 und 2,5 Sekunden
- **Benutzerfreundlichkeit**
 - **Ziel:** Die App sollte eine benutzerfreundliche Erfahrung bieten und die Auswirkungen von Fehlern minimieren
 - **Szenario:** Wenn ein Benutzer während einer Transaktion auf einen Fehler stößt und sich entscheidet, abubrechen, sollte der Abbruchprozess weniger als eine Sekunde dauern

Risiken und technische Schulden

- **Risiko:** Eine zu starke Abhängigkeit von Drittanbieterdiensten (wie Zahlungsverarbeitern) könnte zu Systemausfällen führen, falls diese Dienste ausfallen
 - **Gegenmaßnahme:** Implementierung von Ausfallmechanismen und Sicherstellung von Dienstgütevereinbarungen (SLAs)
- **Technische Schulden:** Unzureichende automatisierte Tests könnten dazu führen, dass potenzielle Fehler in die Produktion gelangen
 - **Gegenmaßnahme:** Erhöhung der Investitionen in die Abdeckung automatisierter Tests und kontinuierliche Integrationsprozesse
- **Risiko:** Erhöhte Integrations- und Verwaltungskomplexität in einer Service-Orientierten Architektur (SOA)
 - **Gegenmaßnahme:** Einsatz eines Enterprise Service Bus (ESB) zur Vereinfachung der Kommunikation zwischen den Services, Definition klarer Schnittstellen und Verträge, sowie Implementierung einer Service Registry für das Management der Service-Discovery