

Enterprise Architecture: Design, Integration, Implementierung Fallstudie

Nikita Antonov 01348746
Dina Finkel 01476028
Andrii Nykyforuk 01247560

Juni 2020



1 Einleitung

Modern Car Ride ist ein Unternehmen, das einen Service in Wien anbietet, mit dem Menschen erschwingliche Autos in ihrer Nähe finden und über eine mobile Anwendung mieten können. Es ist sehr schnell, bequem und kostengünstig. Das Unternehmen wurde 2015 gegründet und startete mit 100 Autos. Heute besteht die Flotte aus 350 Autos. Der Service funktioniert nach dem All-Inclusive Konzept, das heißt, dass Parken, Benzin und Versicherung bereits im Servicepreis enthalten sind. Modern Car Ride arbeitet sowohl mit Privatsachen auch mit Firmenkunden zusammen.

Aktuell haben wir folgende Herausforderungen für unseres Unternehmens.

Die Flottenauslastung ist die erste große Herausforderung. Kunden sollten in der Lage sein, ein Auto zu mieten, wenn sie eines wollen, mit minimaler Verzögerung. Gleichzeitig können wir nicht immer und überall genug Autos zu haben.

Die zweite Herausforderung besteht darin, dass die Autos innen und außen sauber sowie in einem gutem technischen Zustand immer sein müssen. Dies kann durch Kundenfeedback in der App und automatisierte technische Überwachung im Auto selbst erreicht werden.

Bei der Präsentation von DSGVO muss besonders darauf geachtet werden, dass

eine angemessene Sicherheit für die Verarbeitung und Speicherung von Kundendaten gewährleistet ist. Dies ist besonders wichtig, da wir folgende sensible Kundendaten wie Name, Adresse, Bankdaten usw. verarbeiten und speichern. Durch End-to-End Verschlüsselung, ausgearbeitete Firewall-Regeln, ständige Überwachung des Netzwerks und regelmäßige Software-Updates bieten wir die bestmögliche Sicherheit für Daten und können sicherstellen, dass unsere Kunden vollständig geschützt sind.

2 Strategy

Unsere Mission ist es, jedem unserer Kunden Freude und Freiheit der privaten Mobilität zu bringen. Mit unseren Dienstleistungen können Sie sich in Wien schneller bewegen und unabhängig von den Fahrplänen der öffentlichen Verkehrsmittel sein.

Um Wachstum und Erfolg zu erzielen, haben wir **Vision** und das folgende Leitbild definiert: "Unsere Vision ist es, das beste Car-Sharing Unternehmen in Wien zu sein, indem wir talentierte Mitarbeiter beschäftigen, engagieren und befähigen, durch erstklassigen Service ein stärkeres Geschäft aufzubauen."

Die Unternehmenswerte sind:

- Respekt und Integrität
- Verantwortlichkeit
- Kundenorientierung
- Der Beste sein

Die Strategie umfasst Elemente für Capabilities und Ressourcen, die dem Unternehmen helfen, seine Ziele zu erreichen. Das Hauptziel unseres Unternehmens ist die Steigerung des Gewinns. Dies kann durch die Steigerung der Kundenzufriedenheit, Umsatzsteigerung und Kosten-senkung erreicht werden.

Durch die Verwendung moderner UI / UX-Designlösungen zusammen mit einem vereinfachten Registrierungsprozess bieten wir unseren Kunden eine intuitive Nutzung der App. Dank der vollständig automatisierten und ausgearbeiteten Algorithmen ist es sehr einfach, das richtige Auto in der Nähe des aktuellen Standorts zu finden. Der Zahlungsvorgang ist ebenfalls voll automatisiert und unsere Kunden erhalten eine detaillierte Rechnung sowie eine Benachrichtigung vor der Lastschrift. Die Rechnung enthält Informationen wie das Datum der Fahrt, Start- und Endadressen, die Gesamtzeit und die gefahrenen Kilometer sowie Informationen über das verwendete Auto.

Um Risiken und Kosten zu minimieren, verwenden wir ein Kundenverwaltungssystem, mit dem Fahrer mit schlechten Gewohnheiten (rücksichtsloses Fahren, Überschreiten von Geschwindigkeitsbegrenzungen, Ignorieren von Parkregeln usw.) gefiltert werden können. Auch der Führerschein wird auf seine Gültigkeit überprüft (z.B. ob die entsprechende Führerscheinklasse erlaubt ist).

Unsere Autos sind voll ausgestattet mit modernster Technologie, einschließlich Software und verschiedenen Sensoren. Dies hilft uns, die geografische Position des Autos, den aktuellen technischen Zustand, die verbleibende Kraftstoffmenge und die Fahrgewohnheiten der Kunden zu kontrollieren. Unsere selbst entwickelten Algorithmen helfen dabei, den Planungsprozess für die Fahrzeugwartung zu automatisieren. Auf diese Weise erreichen wir eine maximale Verfügbarkeit der Autos für unsere Kunden.

Unsere Fahrzeugflotte ist rund um die Uhr verfügbar und unser Support-Team ist auch rund um die Uhr verfügbar. Für unsere Kunden ist es sehr wichtig, sich sicher zu fühlen und zu wissen, dass sie im Notfall die erforderliche Unterstützung erhalten. Alle diese Funktionen werden über den Dispatcher-Service realisiert. Wir wissen immer die genaue Position des Autos, welche Art von Notfall hätte passieren können.

3 Business Architecture

In unserer Fallstudie verwenden wir die Capability Map, um die Geschäftsmöglichkeiten von Modern Car Ride Unternehmen zu beschreiben. Die unten stehende Capability Map richtet sich direkt an ein Carsharing-Unternehmen und spiegelt die fünf Kerncapabilities des Unternehmens wider. In diesen fünf Capabilities werden wiederum detailliertere Capabilities verschachtelt, die genauer das Geschäftspotenzial des Unternehmens widerspiegeln. In unserem Fall haben die fünf wichtigsten Capabilities für das Unternehmen das gleiche Gewicht und werden daher in Form eines einstufigen Systems dargestellt. Im Detail wird jede Gelegenheit unten beschrieben.

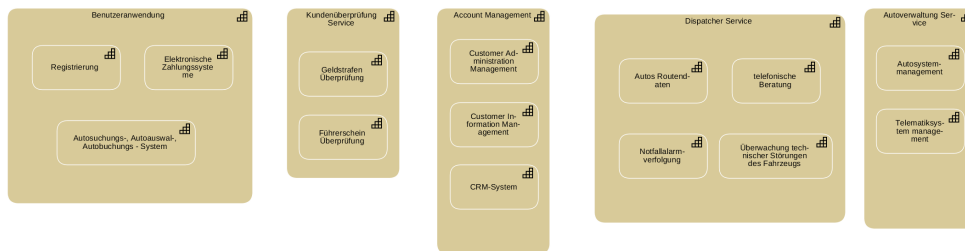


Figure 1: Capability Map

- **Benutzeranwendung:** ermöglicht die Kunde sich im Service zu registrieren, Zahlungsmethoden eingeben und Zahlungsverlauf überprüfen, das Auto mieten. Während des Registrierungsprozesses füllt der Kunde das Registrierungsformular aus und lädt eine Kopie des Führerscheins und des

Meldezettels hoch. dann wählt der Kunde eine Zahlungsmethode (SEPA-Lastschriftmandat oder Kreditkarte) und gibt die Informationen ein. Auch hier kann der Kunde das Auto den gesamten Mietprozess steuern.

- **Kundenüberprüfung Service:** Es spiegelt die grundlegenden Fähigkeiten und Funktionen eines Sicherheitsdienstes wider. Der Kundeüberprüfung Service hat den Kunden auf die Echtheit eines Führerscheins, das Vorhandensein von Geldbußen und andere Kriterien zu überprüfen. Wenn der Überprüfung erfolgreich ist, wird der Fahrer als neuer Kunde des Car-Sharing Unternehmens und wird im CRM-System angemeldet.
- **Account Management:** Dieser Capability stellt auch zur Verfügung, um die Funktionen zur Verwaltung von Kundenaccounts, zur Unterstützung und Information von Kunden bereitzustellen, die über Dienste auf Anwendungsebene implementiert werden.
- **Dispatcher Service:** dient für die Kundenberatung während einer Miete, gibt auch die Möglichkeiten den Standort des Autos auf der Karte und deren technischen Zustand zu überwachen und kann zum Beispiel den Benutzer bei Verstößen blockieren. Bietet telefonischen Support über einen Call Service.
- **Autoverwaltung Service:** Dieser Capability bietet Führungskräften wie Managern und Technikern die Möglichkeit, in Fahrzeuge eingebaute Telematikgeräte zu verfolgen, zu konfigurieren und zu verwalten. Mit diesem Service ist es möglich Telematikdienst anhand Telematiksystemen und Sensoren in dem Fahrzeug kann die Leistung des Autos verfolgen und deren Probleme und Standort melden.

4 Application Architecture

Die Anwendungsebene beschreibt das Zusammenspiel von Services eines Carsharing-Unternehmens und ist sowohl eine separate Struktur als auch unterstützt sie der Capability Map. Eine der Hauptkomponenten des Autoverwaltung Services, um seine Dienste für den Kunden bereitzustellen, ist der Mobile App. Über die Schnittstelle dieser Anwendung kann der Benutzer Dienste wie die Registrierung im System, Suchung und Entsperung eines Autos sowie die Zahlung nutzen. Bei der Registrierung sendet der Kunde Dokumente wie Ausweis und Führerschein zur Überprüfung. Der Sicherheitsdienst überprüft den Kunden auf die Echtheit der Unterlagen. Wenn der Überprüfung erfolgreich ist, wird der Fahrer als neuer Kunde des CarSharing Unternehmens und wird im CRM-System angemeldet. Die Zusammenarbeit von Komponenten wie Customer - Authentication, -Administration, -Information und weiteren implementiert die Arbeit des Kundenverwaltungsdienstes. Zur Verwaltung von Kundendaten nutzen Carsharing-Manager die Client-Datenverwaltungsschnittstelle, die den Zugriff auf alle Komponenten des Customer Administration Services ermöglicht. Eine

notwendige Aufgabe eines Carsharing-Unternehmens ist die Lösung von Kundenproblemen. Somit ist der Call-Center-Service für den Empfang eingehender Beschwerden von Kunden, den Kundendienst und die Lösung komplexer Probleme verantwortlich. Der Telematikdienst interagiert mit dem Autoverwaltungs-dienst und über ihn werden Befehle von der mobilen Anwendung an den Bord-computer des Autos übertragen und Daten über den Zustand der Fahrzeugsysteme und Koordinaten des Autos empfangen.

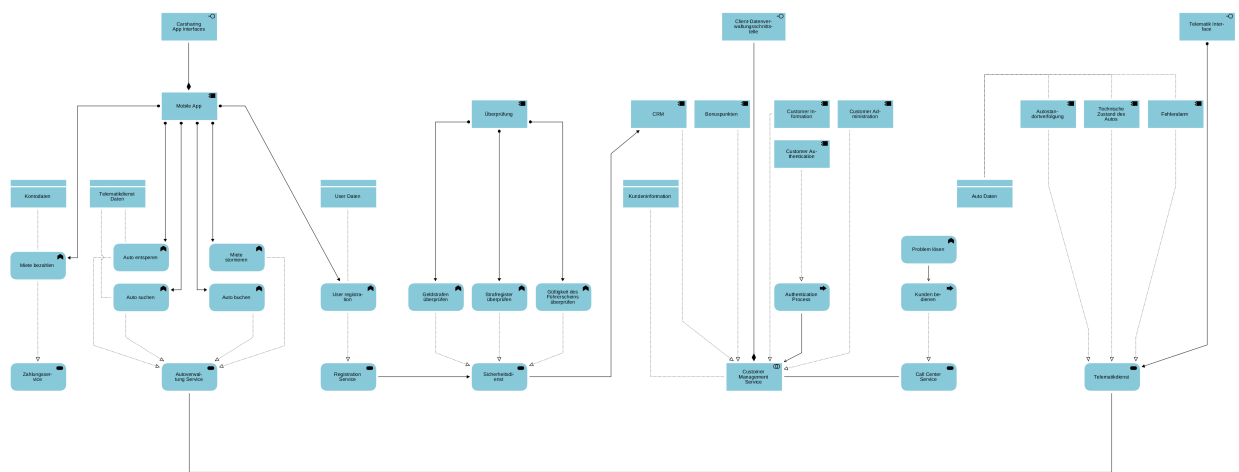


Figure 2: Application Layer

Active Structure Elements:

- **Mobile App:** Der Zugriff auf fast alle Carsharing-Dienste erfolgt über die Anwendung.
- **Überprüfung:** Komponente, die für die Überprüfung der Kundendaten und die Implementierung eines Sicherheitsdienstes verantwortlich ist.
- **Customer Authentication:** ist für die Authentifizierung der Kundendaten verantwortlich.
- **Customer Administration:** Verantwortlich für die Verwaltung der Benutzerkonten.
- **Bonuspunkten:** Bonuspunkten Verwaltung.
- **CRM:** verwaltet alle Informationen über den Kunden und die Chronologie der Zusammenarbeit mit ihm.
- **Customer Information:** führt die Funktionen zur Information des Kunden aus.

- **Autostandortverfolgung:** Verfolgung und Übertragung des Fahrzeugstandorts. Zusammen mit anderen Komponenten implementiert es einen Telematikdienst.
- **Technische Zustand des Autos:** Verfolgung des Betriebs von Fahrzeugsensoren, Eigenschaften des technischen Zustands des Fahrzeugs.
- **Fehleralarm:** Komponente, die für die Erkennung und Meldung einer technischen Störung des Fahrzeugs oder eines Notfalls verantwortlich ist.
- **Customer Management Service:** ist ein internes System des Unternehmens und dient für CRM, Bonuspunkten Verwaltung, Kunden Authentifizierung und Verwaltung.
- **Carsharing App Interface:** Zugriff, über den der Benutzer die Funktionen (Registrierung, Zahlung, Autoverwaltung) der mobilen Anwendung nutzen kann.
- **Client Datenverwaltungsschnittstelle:** Zugriff auf und Verwaltung von Kundendaten, Verwaltung aller vom Customer Management Services bereitgestellten Funktionen.
- **Telematik Interface:** Zugriff auf die Telematikdienste des Fahrzeugs, mit denen Manager und Ingenieure die technische Leistung des Fahrzeugs verfolgen, den Standort des Fahrzeugs ermitteln und technologische Einstellungen vornehmen können.

Behavior Elements:

- **Auto suchen:** über die mobile Anwendung kann der Benutzer das Auto finden.
- **Auto buchen:** über die mobile Anwendung kann der Benutzer das Auto buchen.
- **Auto entsperren:** über die mobile Anwendung kann der Benutzer den Mietwagen entsperren.
- **Miete bezahlen:** über die mobile Anwendung kann der Benutzer die Autovermietung bezahlen.
- **Miete stornieren:** über die mobile Anwendung kann der Benutzer die Reservierung stornieren.
- **User registration:** über die mobile Anwendung kann sich der Benutzer im System registrieren.
- **Geldstrafenüberprüfen:** implementiert einen Sicherheitsdienst, dessen eine Funktion darin besteht, einen neuen Client auf Bußgelder zu überprüfen.
- **Strafregisterüberprüfen:** implementiert einen Sicherheitsdienst, dessen eine Funktion darin besteht, einen neuen Kunden auf Vorstrafen zu überprüfen.

- **Gültigkeit des Führerscheins überprüfen:** implementiert einen Sicherheitsdienst, dessen eine Funktion darin besteht, einen neuen Client auf die Echtheit eines Führerscheins zu überprüfen.
- **Problem lösen:** Call Center sollte bei der Lösung eines Kundenproblems helfen.
- **Authentication Process:** die Gültigkeit der vom Benutzer eingegebenen Informationen wird überprüft mit der Informationen, die in der Datenbank gespeichert werden.
- **Kunden bedienen:** ein Prozess, der den Call Center Service implementiert, dessen Aufgabe es ist, ein Client-Problem zu lösen.
- **Zahlungsservice:** ist ein automatisierter Prozess zur Durchführung von Zahlungen.
- **Autoverwaltung Service:** ist für Mietprozess relevant. Man kann ein freies Auto finden, buchen, Miete straten/beenden.
- **Registration Service:** dient für neuen Kunderegistrierung und entsprechende Daten an Sicherheitsdienst weiterleiten.
- **Sicherheitsdienst:** überprüft Kundedaten während neuer Anmeldung, entstehende Verkehrsstrafen, gültigkeit des Führerscheins.
- **Call Center Service:** unterstützt Kunden bei verschiedenen Anfragen.
- **Telematikdienst:** kontrolliert den technischen Zustand des Fahrzeugs und erhält unterschiedliche Informationen von Fahrzeugsensoren.

Passive Structure Elements:

- **Kontodaten:** Daten für Transaktionen erforderlich.
- **Telematikdienst Daten:** Daten, Telematikdienste, die sowohl für die Suche nach einem Standort als auch für das Autofahren (start/entsperren) erforderlich sind.
- **User Daten:** Daten, die der Benutzer für seine Registrierung im System übermittelt.
- **Kundeninformation:** alle gesammelten Kundeninformationen.
- **Auto Daten:** Informationen über das Auto gesammelt, um die Möglichkeit seines Betriebs zu bewerten.

5 Technology Architecture

Hier ist eine technologische Sicht auf unser Unternehmen. Die technologische Sichtweise enthält die software- und hardwaretechnischen Elemente, die die Anwendungsschicht unterstützen, wie z.B. physische Geräte, Netzwerke oder Systemsoftware (z.B. Betriebssysteme, Datenbanken und Middleware)[6]. Daher versuchten wir, diese Elemente des Unternehmensmodells so objektiv wie möglich darzustellen. Natürlich gab es auch ein gewisses Abstraktionsniveau, wie es bei Modellen sein sollte, und nicht jedes Programm und jeder Draht wurde abgebildet. Andernfalls würde das Modell überflüssig und unlesbar werden und seine Funktion nicht erfüllen. Das Modell konzentriert sich auf Server- und Netzwerkdesign. Dies ist die wichtigste und kritischste Infrastruktur. Dieser Teil des Modells ist der "unterste" im allgemeinen Modell, so dass er keine Verbindungen zu den unteren Ebenen hat. Die Kommunikation mit einer höheren Ebene (nämlich **Information Systems Architecture**) erfolgt durch die Bereitstellung verschiedener von ihr in Anspruch genommener **Services**. Die verschiedenen Teile des Modells werden im Folgenden ausführlicher besprochen:

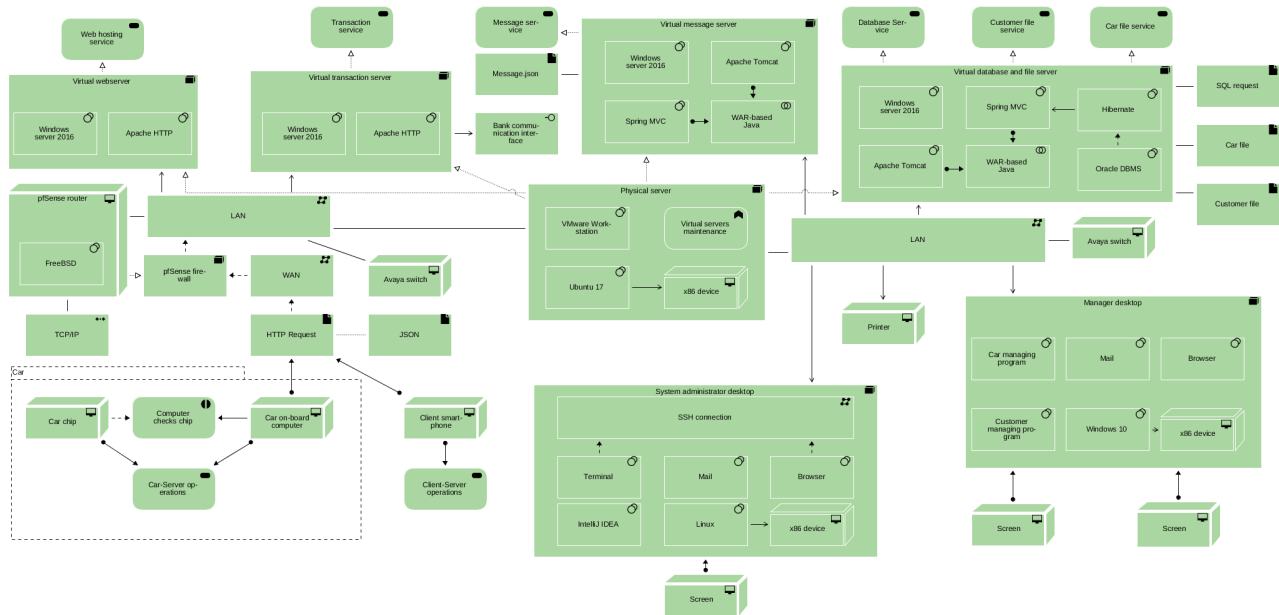


Figure 3: Infrastruktur-Layer

Server:

- **Physical Server:** Server ist ein Computerprogramm oder ein Gerät, welches Funktionalitäten für andere Programme oder Geräte (Clients“) bereitstellt[16]. Unser Unternehmen verfügt über einen physischen Server, auf dem verschiedene spezialisierte virtuelle Server basieren (siehe den

entsprechenden Abschnitt der Beschreibung über diese virtuellen Servern). Sie spart Ressourcen und bildet gleichzeitig ein komplettes System verschiedener Services. Die Hauptkomponenten unseres physischen Servers:

1. **x86 device:** x86 ist die Abkürzung einer Mikroprozessor-Architektur und der damit verbundenen Befehlssätze, welche unter anderem von den Chip-Herstellern **Intel** und **AMD** entwickelt werden[22]. Sodass angegeben werden kann, welcher Prozessor eingebaut werden soll.
 2. **Ubuntu 17:** Ubuntu ist eine Linux-Distribution, die auf Debian basiert[19]. Es ist ein ziemlich bequemes Betriebssystem, um das Gerät zu steuern.
 3. **VMware Workstation:** VMware Workstation Pro ist eine Software des US-amerikanischen Unternehmens VMware zur Virtualisierung von Computern mit x86- bzw. x86-64-Architektur[20]. Dabei handelt es sich um einen Hypervisor, der es uns ermöglicht, unsere verschiedenen virtuellen Server auf derselben physischen Maschine laufen zu lassen.
- **Virtual servers:** Unser Unternehmen verfügt über 4 verschiedene virtuelle Server(**Webserver, Transaction Server, Message Server, Database and file server**). Mit der Virtualisierung stellen wir also 4 logisch verschiedene Server auf einem physischen Gerät bereit. Dies sind die Servern, die die Dienste des übergeordneten Modells(Application) bereitstellen. Das sind **Web hosting service, Transaction service, Message service, Database service, Customer file service** und **Car file service**. Die Hauptkomponenten unseres virtuellen Servers:
 1. **Windows server 2016:** Es ist ein Serverbetriebssystem des Unternehmens Microsoft[21]. Dieses Betriebssystem wird auf allen 4 virtuellen Servern verwendet.
 2. **Apache HTTP:** Es ist ein quelloffenes und freies Produkt der Apache Software Foundation und einer der meistbenutzten Webserver im Internet[7]. Es ist auf **Webserver** und **Transaction Server** installiert.
 3. **Bank communication interface:** Diese Schnittstelle ermöglicht es dem **Transaction Server**, mit Banken und deren Dienstleistungen zu interagieren.
 4. **Spring MVC:** Es ist ein quelloffenes Framework für die Java-Plattform, welches oft für Web-Anwendungen verwendet wird[17]. Das darauf basierende Programm gewährleistet den Betrieb von **Message Server** und **Database and file server**.
 5. **Apache Tomcat:** Apache Tomcat ist ein Open-Source-Webserver und Webcontainer, der die Spezifikation für Java Servlets und JavaServer Pages (JSP) implementiert und es damit erlaubt, in Java geschriebene Web-Anwendungen auf Servlet- beziehungsweise JSP-Basis auszuführen[8]. Es hält die Server mit Spring-Programmen am Laufen.

6. **Oracle DBMS:** Es ist eine Datenbankmanagementsystem-Software des Unternehmens Oracle. Es können sowohl relationale Daten als auch objektrelationale Daten gespeichert werden[13]. Es wird auf dem **Database and file server** verwendet, um die Datenbank zu verwalten.
7. **Hibernate:** Hibernates Hauptaufgabe ist die objektrelationale Abbildung (englisch O-R-Mapping, kurz ORM). Dies ermöglicht es, gewöhnliche Objekte mit Attributen und Methoden (im Java-Umfeld Plain Old Java Objects oder POJOs genannt) in relationalen Datenbanken zu speichern und aus entsprechenden Datensätzen wiederum Objekte zu erzeugen[10]. Es stellt eine Verbindung zwischen der Spring-Anwendung und der Datenbank auf dem **Database and file server** her.
8. **Message.json** Die Nachrichten werden vom **Message server** im JSON-Format ausgetauscht. **JSON** ist als eine Datenaustauschsprache entworfen worden, die von Menschen lesbar und für Computer leicht zu analysieren und zu verwenden ist.[5].
9. **SQL request, Car file, Customer file:** Das sind die verschiedenen Arten von Daten, mit denen der **Database and file server** arbeitet.

Desktops:

- **System administrator desktop:** Es ist der Computer, der von dem für die Wartung des Systems verantwortlichen Systemadministrator verwendet wird. Die Hauptkomponenten der Desktop:
 1. **x86 device:** x86 ist die Abkürzung einer Mikroprozessor-Architektur und der damit verbundenen Befehlssätze, welche unter anderem von den Chip-Herstellern **Intel** und **AMD** entwickelt werden[22]. Sodass angegeben werden kann, welcher Prozessor eingebaut werden soll.
 2. **Linux:** Es ist die freie, unixähnliche Mehrbenutzer-Betriebssysteme, die auf dem Linux-Kernel und wesentlich auf GNU-Software basieren[12]. Es ist das Betriebssystem, das diesen Computer steuert.
 3. **IntelliJ IDEA** Es ist eine integrierte Entwicklungsumgebung (IDE) des Softwareunternehmens JetBrains für die Programmiersprachen Java, Kotlin, Groovy und Scala[11]. Es wird benötigt, um neue Softwarekomponenten zu entwickeln oder eine bestehende testen zu können.
 4. **Terminal:** Es wird vom Administrator benötigt, um Informationen über das System zu verwalten und ein- und auszugeben.
 5. **Mail:** Es ist eines der wichtigsten Programme in der heutigen Welt, das das Senden und Empfangen von elektronischen Nachrichten ermöglicht.
 6. **Browser:** Hierbei handelt es sich um ein Programm zum Surfen im Internet und zum Zugriff auf das Internet.

- **Manager desktop:** Es handelt sich um eine Art Desktop, der von Managern verwendet wird, die nicht in die IT-Verwaltung involviert sind.
 1. **x86 device:** Ähnlich dem vorherigen Desktop-Typ.
 2. **Windows 10:** Diese Desktops verwenden das Windows-Betriebssystem, das bequemer und weiter verbreitet ist. Darüber hinaus verfügt es über ein eigenes Paket verschiedener für das Büro nützlicher Programme.
 3. **Mail:** Ähnlich dem vorherigen Desktop-Typ.
 4. **Browser:** Ähnlich dem vorherigen Desktop-Typ.
 5. **Car managing program:** Es ist ein Programm zur Verwaltung von Fahrzeugdaten.
 6. **Customer managing program:** Es ist ein Programm zur Verwaltung von Kundendaten.
- **Screen:** Verschiedene Bildschirme werden verwendet, um eine Verbindung zu Computern herzustellen und das Bild anzuzeigen.

Netzwerk:

- **WAN:** Es ist im Allgemeinen ein Internet-Netzwerk, das sich außerhalb des Unternehmens befindet.
- **HTTP Request:** Dieses Protokoll wird zur Übertragung von Daten über Netzwerke verwendet.
- **JSON:** Daten in diesem Format werden über das Netzwerk übertragen.
- **pfSense router:** Das sind Netzwerkgeräte, die Netzwerkpakete zwischen mehreren Rechnernetzen weiterleiten können[15]. pfSense ist eine Firewall-Distribution auf der Basis des Betriebssystems **FreeBSD** und des Paketfilters pf[14].
- **pfSense firewall:** Es ist ein Sicherungssystem, das ein Rechnernetz oder einen einzelnen Computer vor unerwünschten Netzwerkzugriffen schützt[9].
- **TCP/IP:** Es ist das Basisprotokoll für die Netzwerkdatenübertragung (Netzwerk- und Transportebene).
- **LAN:** Es ist die Abkürzung für Local Area Network. Es bezeichnet Netzwerke, mit begrenzter räumlicher Ausdehnung[4]. Dieses Netzwerk verbindet Geräte innerhalb des Unternehmens (Server, Computer usw.).
- **Avaya Switch:** Es bezeichnet ein Kopplungselement in Rechnernetzen, das Netzwerksegmente miteinander verbindet[18]. Avaya ist einer der Hersteller dieser Geräte.

Andere Elemente:

- **Car chip:** Es ist ein im Auto installierter Chip, der es dem Benutzer ermöglicht, mit ihm zu interagieren.
- **Car on-board computer:** Es ist der Bordcomputer eines Autos, der es Ihnen ermöglicht, mit ihm zu interagieren, ihn einzustellen und einen Internetzugang zu haben.
- **Client smartphone:** Es ist ein Client-Smartphone mit der von unserem Unternehmen installierten Anwendung, die es ermöglicht, mit dem Auto zu interagieren.
- **Printer:** Dokumentendrucker.

6 Assessment

Der Bewertungsfragebogen ist eine sehr effektive Möglichkeit, mit den Kunden zu kommunizieren und herauszufinden, was sie über Unternehmen und Produkt denken. Es liefert die realen Daten, die verwendet werden können, um optimale Geschäftsentscheidungen zu treffen, das Produkt oder die Dienstleistung zu verbessern.

EA Assessment						
	Architecture Element Type	Specialisation	Question	Source	Category	User Type
1	Application Component	Mobile App	Ich denke, dass ich das System häufig benutzen würde	System Usability Scale/URL	User Experience	Kunde
2	Application Component	Mobile App	Ich fand das System unnötig komplex.	System Usability Scale/URL	User Experience	Kunde
3	Application Component	Mobile App	Ich fand das System einfach zu bedienen.	System Usability Scale/URL	User Experience	Kunde
4	Application Component	Mobile App	Ich denke, dass ich die Unterstützung einer technischen Person benötigen würde, um dieses System verwenden zu können.	System Usability Scale/URL	User Experience	Kunde
5	Application Component	Mobile App	Ich fand die verschiedenen Funktionen in diesem System gut integriert.	System Usability Scale/URL	User Experience	Kunde
6	Application Component	Mobile App	Ich dachte, dass dieses System zu inkonsistent ist.	System Usability Scale/URL	User Experience	Kunde
7	Application Component	Mobile App	Ich würde mir vorstellen, dass die meisten Leute sehr schnell lernen würden, dieses System zu benutzen.	System Usability Scale/URL	User Experience	Kunde
8	Application Component	Mobile App	Ich fand das System sehr umständlich zu bedienen.	System Usability Scale/URL	User Experience	Kunde
9	Application Component	Mobile App	Ich war sehr zuversichtlich mit dem System.	System Usability Scale/URL	User Experience	Kunde
10	Application Component	Mobile App	Ich musste viele Dinge lernen, bevor ich mit diesem System anfangen konnte.	System Usability Scale/URL	User Experience	Kunde
11	Application Component	Mobile App	Ich mag den Interface der App	SERVQUAL	User Experience	Kunde
12	Application Component	Mobile App	Es ist einfach, die Informationen zu finden, die ich brauche	SERVQUAL	User Experience	Kunde
13	Application Component	Mobile App	Immer wenn ich mit der App einen Fehler mache, kann ich mich einfach und schnell erholen	SERVQUAL	User Experience	Kunde
14	Application Component	Mobile App	Die Werbung und Promotionen in der App spiegeln die Realität wider.	SERVQUAL	User Experience	Kunde
15	Application Component	Mobile App	Die App verwendet modernes Design und verständliches Layout.	SERVQUAL	User Experience	Kunde

Figure 4: Enterprise Architecture Assessment[3]

EA Assessment						
	Architecture Element Type	Specialisation	Question	Source	Category	User Type
16	Technology Component	Database Service	Ich fand das System unnötig komplex.	System Usability Scale/URL	Technische Realisierung	IT-Mitarbeiter
17	Technology Component	Database Service	Ich fand das System einfach zu bedienen.	System Usability Scale/URL	Technische Realisierung	IT-Mitarbeiter
18	Technology Component	Database Service	Ich fand die verschiedenen Funktionen in diesem System gut integriert.	System Usability Scale/URL	Technische Realisierung	IT-Mitarbeiter
19	Technology Component	Database Service	Ich dachte, dass dieses System zu inkonsistent ist.	System Usability Scale/URL	Technische Realisierung	IT-Mitarbeiter
20	Technology Component	Database Service	Ich würde mir vorstellen, dass die meisten Leute sehr schnell lernen würden, dieses System zu benutzen.	System Usability Scale/URL	Technische Realisierung	IT-Mitarbeiter
21	Technology Component	Database Service	Ich fand das System sehr umständlich zu bedienen.	System Usability Scale/URL	Technische Realisierung	IT-Mitarbeiter
22	Technology Component	Database Service	Ich musste viele Dinge lernen, bevor ich mit diesem System anfangen konnte.	System Usability Scale/URL	Technische Realisierung	IT-Mitarbeiter
23	Technology Component	Database Service	Ich mag den Interface des Systems.	SERVQUAL	Technische Realisierung	IT-Mitarbeiter
24	Technology Component	Database Service	Es ist einfach, die Informationen zu finden, die ich brauche.	SERVQUAL	Technische Realisierung	IT-Mitarbeiter
25	Technology Component	Database Service	Immer wenn ich mit de, System einen Fehler mache, kann ich mich einfach und schnell erholen.	SERVQUAL	Technische Realisierung	IT-Mitarbeiter

Figure 5: Enterprise Architecture Assessment[1]

EA Assessment						
	Architecture Element Type	Specialisation	Question	Source	Category	User Type
26	Capability Component	Account Management	Organisation hat eine klare Kundenbeziehung	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
27	Capability Component	Account Management	Die Organisation betrachtet ihre Kundenbeziehung als Kommunikation zur Beschreibung der Ziele	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
28	Capability Component	Account Management	Die Kundenbeziehung unterstützt die Beschreibung der aktuellen Beziehung zwischen der Organisation und dem Kunden	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
29	Capability Component	Account Management	Kundenbeziehung ist ein wichtiger Weg, um eine erfolgreiche Beziehung zu den Kunden aufzubauen	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
30	Capability Component	Account Management	Unternehmen nutzt jede grundlegende Informationen über die Kunden, um sie zu gewinnen.	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
31	Capability Component	Account Management	Unternehmen nutzen verschiedene Werkzeuge, um die Kunden zu gewinnen.	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
32	Capability Component	Account Management	Die Qualität der vorhandenen Daten wirkt sich auf den aktuellen Kunden aus	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
33	Capability Component	Account Management	Customer Knowledge Capture hilft zu verstehen, wie das benötigte Wissen erfasst werden kann	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
34	Capability Component	Account Management	Customer Knowledge Capture muss die Quelle des Kundenwissens bestimmen	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
35	Capability Component	Account Management	Customer Knowledge Capture ist wichtig, um die Zuverlässigkeit und Richtigkeit des Kundenwissens für die weitere Verarbeitung zu testen	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
36	Capability Component	Account Management	Die Phase der Customer Knowledge Capture konzentriert sich auf die Erfassung des in den Kunden vorhandenen Kundenwissens	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
37	Capability Component	Account Management	Die Qualität der Datenvollständigkeit wirkt sich auf die Analysephase aus	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
38	Capability Component	Account Management	Die Analyse von Kundendaten kann dabei helfen, das Verhalten der Kunden prognostizieren	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
39	Capability Component	Account Management	Das Analysieren von Daten erfordert die Klassifizierung der zusammengesetzten Daten	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
40	Capability Component	Account Management	Das Unternehmen muss Werbung einsetzen, um neue Kunden zu gewinnen	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
41	Capability Component	Account Management	Das Unternehmen adaptiert bestimmte Techniken zur Gewinnung neuer Kunden	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
42	Capability Component	Account Management	Marketingkommunikationstools werden zur Gewinnung neuer Kunden eingesetzt	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung
43	Capability Component	Account Management	Das Unternehmen verwendet Kundenprofile	Process Maturity Level	Kunden Management	Geschäftsführung, Marketing Abteilung

Figure 6: Enterprise Architecture Assessment[2]

7 Anhang

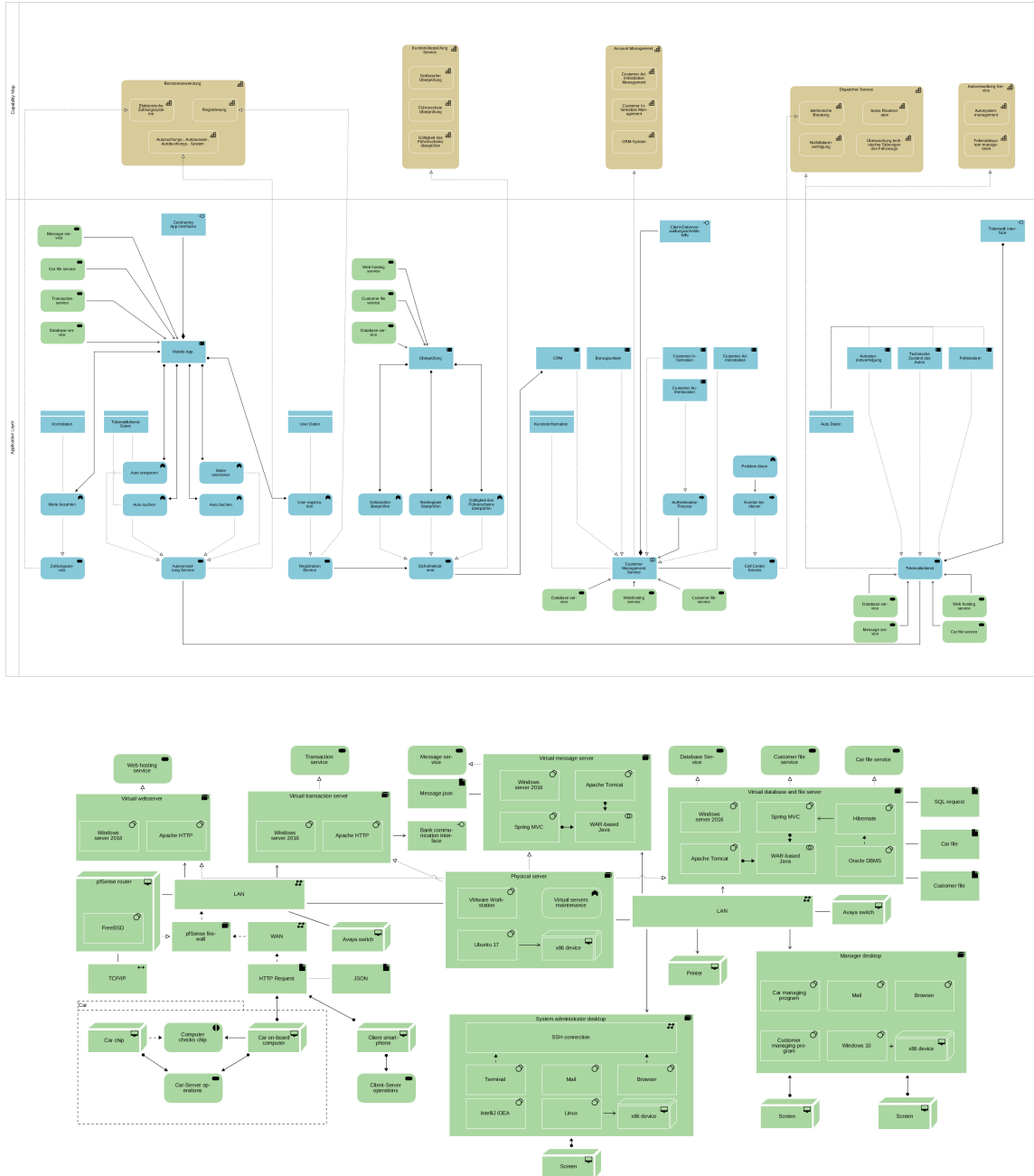


Figure 7: Gesamtarchitektur

Hier ist ein Modell unseres gesamten Unternehmens. Man sieht, dass alle seine Ebenen miteinander verbunden sind. Auf diese Weise entsteht ein vollwertiges Modell, das es ermöglicht, alle Interaktionen, die im Unternehmen stattfinden, zu verstehen. Die **Capabilities** sind direkt mit den entsprechenden Elementen auf der **Applikationsebene** verbunden. In unserem Fall ist es die verständlichste und bequemste Zusammensetzung, weil man sofort verstehen kann, welches Element welche **Capabilities** haben sollte. Die Kommunikation der **Applikations-** und **Technologie-Ebenen** erfolgt über **Services**. Dies ermöglicht es, spezifische, klare und verständliche Interaktionspunkte zu identifizieren. Dies sind die **Services**, die in unserem Modell verwendet werden:

- **Web hosting service:** dieser Service speichert Informationen auf dem Server und betreibt Webservices
- **Transaction service:** dieser Service bietet die Infrastruktur für die Durchführung von Transaktionen
- **Message service:** dieser Service bietet eine Infrastruktur für die Nachrichtenübermittlung über APIs
- **Database service:** dieser Service stellt die Verbindung zur Datenbank her
- **Customer file service:** dieser Service ermöglicht es, Kundendaten zu empfangen und zu senden
- **Car file service:** dieser Service ermöglicht es, Autodaten zu empfangen und zu senden

8 Anmerkungen

Generelle Anmerkungen:

- Der Infrastruktur-Layer ist nicht ausreichend tief beschrieben. In den nachfolgenden Websourcen finden Sie typische Elemente der Technologieebene (bspw. Elemente zur Beschreibung von Netzwerktopologien wie Switch, Router etc.) Solche Elemente sollten sich auch in Ihrem Architekturmodell wiederfinden: **Unsere Infrastruktur wird ausführlich beschrieben, einschließlich der vollständigen Struktur des Netzwerks (Switches, Router usw.) und der Server.**
- Viele von Ihnen haben einfach die Capability Map einer Versicherung verwendet. Das ist grundsätzlich OK. Allerdings wurde diese vielfach nicht ausreichend auf ein Car Sharing Unternehmen angepasst. Tw. sind noch "Relikte" wie Money Management enthalten, die sich auf das Versicherungsgeschäft beziehen und mit Car Sharing nichts zu tun haben. **In**

der Funktionsübersicht haben wir nur die Fähigkeiten verwendet, die wir für das Carsharing-Unternehmen als notwendig erachtet haben, und dabei von allgemeinen Fähigkeiten wie beispielsweise Finanzmanagement oder Fahrzeugwartung von Drittunternehmen abstrahiert, die nicht nur für Carsharing Unternehmen innewohnt sind. Deswegen haben wir solche Capability wie Autowartung ausgeschlossen.

- Jede der Capabilities muss in der finalen Ausarbeitung mit zumindest einem Satz erklärt werden: **Alle Elemente unseres Modells sind im Text beschrieben**
- Capabilities wurden tw. nicht mit den Webservices verknüpft. Ziel in der finalen Ausarbeitung muss es sein, dass alle Ebenen verknüpft werden: **Alle unsere Ebenen sind miteinander verbunden, konkret sind diese Verbindungen in der Anhang ersichtlich.**
- Schneiden Sie Ihre Gesamtarchitektur in einzelne Viewpoints (in der finalen Abgabe): **Unsere Arbeit gliedert sich in drei Ebenen (Business, Application und Technology).**
- In einigen der Arbeiten wurde das Klassifikationsschema Architecture Element—Specialisation—Question—Source—Category—User Type nicht verwendet: **Alle Fragen sind nach Klassifikationsschema angepasst.**
- Die Fragen sollten nicht spezifisch für ein Car Sharing Unternehmen sein. Sie sollen für jedes Unternehmen anwendbar sein: **Alle Fragen sind neutral definiert.**
- Die Fragen sollen nicht selbst ausgedacht werden. Vielmehr sollen Sie Fragebögen wie SUS, SERVQUAL etc. recherchieren und die Fragen dann in das Klassifikationsschema einordnen: **Alternativ Umfragen wurden analysiert - Qualtrics, Software Usability Measurement Inventory, Standardized User Experience Percentile Rank Questionnaire, Questionnaire for User Interaction Satisfaction.**
- Einige Gruppen haben nichts zur Datenarchitektur gemacht. Geschäftsobjekte und Datenobjekte sollten modelliert und zumindest den Webservices und den Schnittstellen zwischen den Webservices (Informationsflüsse) zugeordnet werden: **Wir haben viele Datenelemente hinzugefügt, sowohl auf der Applikations- als auch auf der Technologieebene. Im Einzelnen sind sie auf den entsprechenden Schemas ersichtlich.**
- Jedes der Elemente Ihrer Architektur MUSS mit mindestens einem Satz beschrieben sein: **Alle Elemente unseres Modells sind im Text beschrieben**
- Jeder Teilausschnitt der Architektur (Viewpoint) muss mit einem kurzen Absatz beschrieben werden. **Jeder viewpoint haben wir zuerst allgemein beschrieben und dann detailliert.**

- Die einzelnen Layer müssen untereinander verknüpft werden (Cross-Layer-Viewpoint): **Alle unsere Ebenen sind miteinander verbunden, konkret sind diese Verbindungen in der Anhang ersichtlich.**

Unsere persönliche Anmerkungen:

- LAN, Router, Switch etc. Bitte googlen Sie nach Definitionen und verwenden Sie diese inkl. Quellenangaben: **Elemente der Infrastruktur werden detailliert beschrieben, darunter auch Zitate aus dem Internet. Die Zitate haben entsprechende Quellenangaben.**
- Operating System besser in ArchiMate als System Software abbilden; **Operating System ist jetzt als System Software abgebildet.**
- Server, Server Operating System etc. Bitte ebenfalls nach Definitionen googlen. Dann für das Car Sharing konkrete "Produkte" (z.B. Windows Server 1016 etc.) auswählen: **Wo möglich, werden nun spezifische Produkte angegeben (z.B. Windows, Apache HTTP, etc.)**
- Die Verdichtung auf genau ein Technologieservice passt nicht: **Jetzt gibt es 5 verschiedene Technologieservices.**

References

- [1] Usability testing & information architecture. <https://www.usabilitest.com>.
- [2] ALHAWARI, S. Evaluating customer process to contribute to customer acquisition: A quantitative study in jordanian banking sector. <https://ibimapublishing.com/articles/IBIMABR/2012/382039/>.
- [3] FATEMEH NIKPAY, RODINA AHMAD, C. Y. K. A hybrid method for evaluating enterprise architecture implementation. <https://doi.org/10.1016/j.evalprogplan.2016.09.001>.
- [4] IONOS. Was ist lan? local area network erklärt! <https://www.ionos.de/digitalguide/server/knowhow/lan/>.
- [5] NURSEITOV, N., PAULSON, M., REYNOLDS, R., AND IZURIETA, C. Comparison of json and xml data interchange formats: a case study. *Caine 9* (2009), 157–162.
- [6] PARADIGM, V. Archimate viewpoint guide. https://www.visual-paradigm.com/support/documents/vpuserguide/4455/4456/86483_technologyvi.html.
- [7] WIKIPEDIA. Apache http server. https://de.wikipedia.org/wiki/Apache_HTTP_Server.

- [8] WIKIPEDIA. Apache tomcat. https://de.wikipedia.org/wiki/Apache_Tomcat.
- [9] WIKIPEDIA. Firewall. <https://de.wikipedia.org/wiki/Firewall>.
- [10] WIKIPEDIA. Hibernate(framework). [https://de.wikipedia.org/wiki/Hibernate_\(Framework\)](https://de.wikipedia.org/wiki/Hibernate_(Framework)).
- [11] WIKIPEDIA. IntelliJ idea. https://de.wikipedia.org/wiki/IntelliJ_IDEA.
- [12] WIKIPEDIA. Linux. <https://de.wikipedia.org/wiki/Linux>.
- [13] WIKIPEDIA. Oracle(datenbanksystem). [https://de.wikipedia.org/wiki/Oracle_\(Datenbanksystem\)](https://de.wikipedia.org/wiki/Oracle_(Datenbanksystem)).
- [14] WIKIPEDIA. pfsense. <https://de.wikipedia.org/wiki/PfSense>.
- [15] WIKIPEDIA. Router. <https://de.wikipedia.org/wiki/Router>.
- [16] WIKIPEDIA. Server. <https://de.wikipedia.org/wiki/Server>.
- [17] WIKIPEDIA. Spring(framework). [https://de.wikipedia.org/wiki/Spring_\(Framework\)](https://de.wikipedia.org/wiki/Spring_(Framework)).
- [18] WIKIPEDIA. Switch(netzwerktechnik). [https://de.wikipedia.org/wiki/Switch_\(Netzwerktechnik\)](https://de.wikipedia.org/wiki/Switch_(Netzwerktechnik)).
- [19] WIKIPEDIA. Ubuntu. <https://de.wikipedia.org/wiki/Ubuntu>.
- [20] WIKIPEDIA. Vmware workstation. https://de.wikipedia.org/wiki/VMware_Workstation.
- [21] WIKIPEDIA. Windows server 2016. https://de.wikipedia.org/wiki/Windows_Server_2016.
- [22] WIKIPEDIA. x86-prozessor. <https://de.wikipedia.org/wiki/X86-Prozessor>.