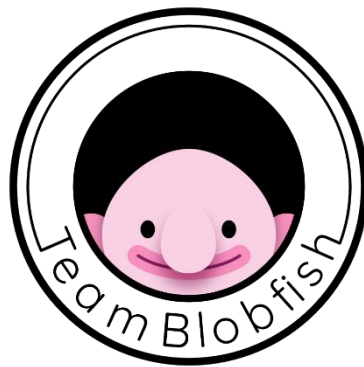


# Architekturdokumentation nach ARC42

Version 2.0 | 22.06.2021

Von Team-Blobfish im Rahmen des SEP SS2021



Kontakt: [HSTeamBlobfish@gmx.de](mailto:HSTeamBlobfish@gmx.de)

Kunde: Capgemini

Autoren: Exoucia Mukubay und Maximilian Stinner

## Versionshistorie

Version	Datum	Vorgenommene Änderung	Autor
2.0	22.06.2021	Korrekturen	Maximilian Stinner
1.6	20.06.2021	Feedback des Managements einarbeiten	Exoucia Mukubay
1.5	17.06.2021	Feedback des Managements einarbeiten	Exoucia Mukubay
1.4	16.06.2021	Feedback des Managements einarbeiten	Exoucia Mukubay
1.3	15.06.2021	Feedback des Managements einarbeiten	Exoucia Mukubay
1.2	14.06.2021	Feedback des Managements einarbeiten	Exoucia Mukubay
1.1	11.06.2021	Feedback des Managements einarbeiten Kapitel 11 hinzugefügt	Exoucia Mukubay
1.0	18.05.2021	Korrekturen	Maximilian Stinner, Sven Reinemuth, Florian Braasch Exoucia Mukubay
0.8	18.05.2021	Kapitel 6 Neues Sequenzdiagramm zu Alexa	Exoucia Mukubay
0.7	17.05.2021	Verbesserungsvorschläge	Maximilian Stinner
0.6	17.05.2021	Kapitel 8 und Kapitel 5 Inhalte hinzugefügt	Exoucia Mukubay
0.5	16.05.2021	Kapitel 1 Inhalte ergänzt, Kapitel 3 neues Diagramm gezeichnet, Kapitel 4 Inhalte hinzugefügt, Sequenzdiagramm, Verteilungsdiagramm	Exoucia Mukubay
0.4	14.05.2021	Inhalte hinzugefügt + erste Korrekturen Kapitel 6 Sequenzdiagramme gezeichnet	Maximilian Stinner, Exoucia Mukubay
0.3	13.05.2021	Kapitel 3, Kapitel 5 -7, Kapitel 10-11	Exoucia Mukubay
0.2	11.05.2021	Kapitel 1-2	Exoucia Mukubay
0.1	10.05.2021	Dokumentstruktur angelegt, Verteilungsdiagramm gezeichnet	Exoucia Mukubay

Die erste Nachkommastelle der Versionsnummer wird jedes Mal, wenn eine Änderung vollzogen wird hochgezählt. Ganzzahlige Versionsnummern stellen ein Inkrement dar, welches dem Management abgegeben wird.

## Inhaltsverzeichnis

1.	Einführung und Ziele .....	4
1.1	Aufgabenstellung.....	4
1.2	Qualitätsziele.....	7
2.	Randbedingungen .....	8
2.1	Technische Randbedingungen.....	8
2.2	Organisatorische Randbedingungen .....	8
2.3	Konventionen .....	8
3.	Kontextabgrenzung .....	10
3.1	Fachlicher und Technischer Kontext .....	10
3.1.1	Tabellarische Auflistung der Nachbarsysteme und Benutzer .....	11
4.	Lösungsstrategien.....	12
5.	Bausteinsicht .....	12
5.1	Ebene 1 - Gesamtsystem .....	13
5.2	Ebene 2 - Frontend .....	13
5.3	Ebene 2 - Backend .....	15
5.4	Ebene 2 – Alexa .....	15
6.	Laufzeitsicht.....	16
6.1	Sequenzdiagramm 1: Nutzer Registration .....	16
6.2	Sequenzdiagramm 2: Tisch reservieren über Alexa-Home .....	18
7.	Verteilungssicht .....	19
8.	Entwurfsentscheidungen und Ihre Auswirkungen .....	21
8.1	Wie soll Alexa mit unserem lokalen My-Thai-Star Webserver kommunizieren?.....	21
8.1.1	Alternative 1: Docker-Container .....	21
8.1.2	Alternative 2: Ngrok „Tunnel“ .....	21
	Entscheidung: .....	22
9.	Software Glossar .....	22
10.	Glossar .....	22
11.	Ergänzungen .....	27
12.	Quellenverzeichnis .....	27

# 1. Einführung und Ziele

Dieses Dokument beschreibt die Software-Architektur des "My-Thai-Star"-Systems. Wir orientieren uns nach dem [arc42 Template](#), weil es das einzige Template ist, welches wir in der Software Engineering Vorlesung kennengelernt haben. Einige Kapitel des arc42 Templates wurden weglassen, weil sie nicht Teil der Anforderung des Managements waren.

"My-Thai-Star" ist ein Restaurant-Managing-System, das dem Projektteam vom Auftraggeber zur Verfügung gestellt wurde.

Das Ziel des Projekts ist es, einen Beitrag zur Entwicklung von "My-Thai-Star" zu leisten. Die Vision stellt ein Produkt dar, welches seinen Nutzen in einem echten, modernisierten Restaurant finden könnte.

## 1.1 Aufgabenstellung

Grundsätzlich gibt es vier verschiedene Rollen, die einen Bezug zum System haben und deren Funktionen erweitert werden müssen. Die Funktionen für den Manager wurden schon vorher implementiert und müssen nicht erweitert werden. Zusammengefasst bestehen die Anforderungen aus der Implementierung von Funktionen für das Kellner-Cockpit, einer Oberfläche für Administratoren und verschiedenen Alexa-Funktionen für den Kunden.

Die geforderten Features werden in [Abbildung 1](#) Use Case Diagramm - Kellner und Administrator und [Abbildung 2](#) Use Case Diagramm - Kunde. ~~Abbildung 2 Use Case Diagramm - Kunde~~ Form von Use Case Diagramme nochmal ausführlich aufgelistet. Nach Kundenwunsch werden die Use Cases in zwei Diagrammen aufgeteilt.

Eine ausführliche Beschreibung der einzelnen Use-Cases sind im [Pflichtenheft](#) Kapitel 4.2 vorzufinden.

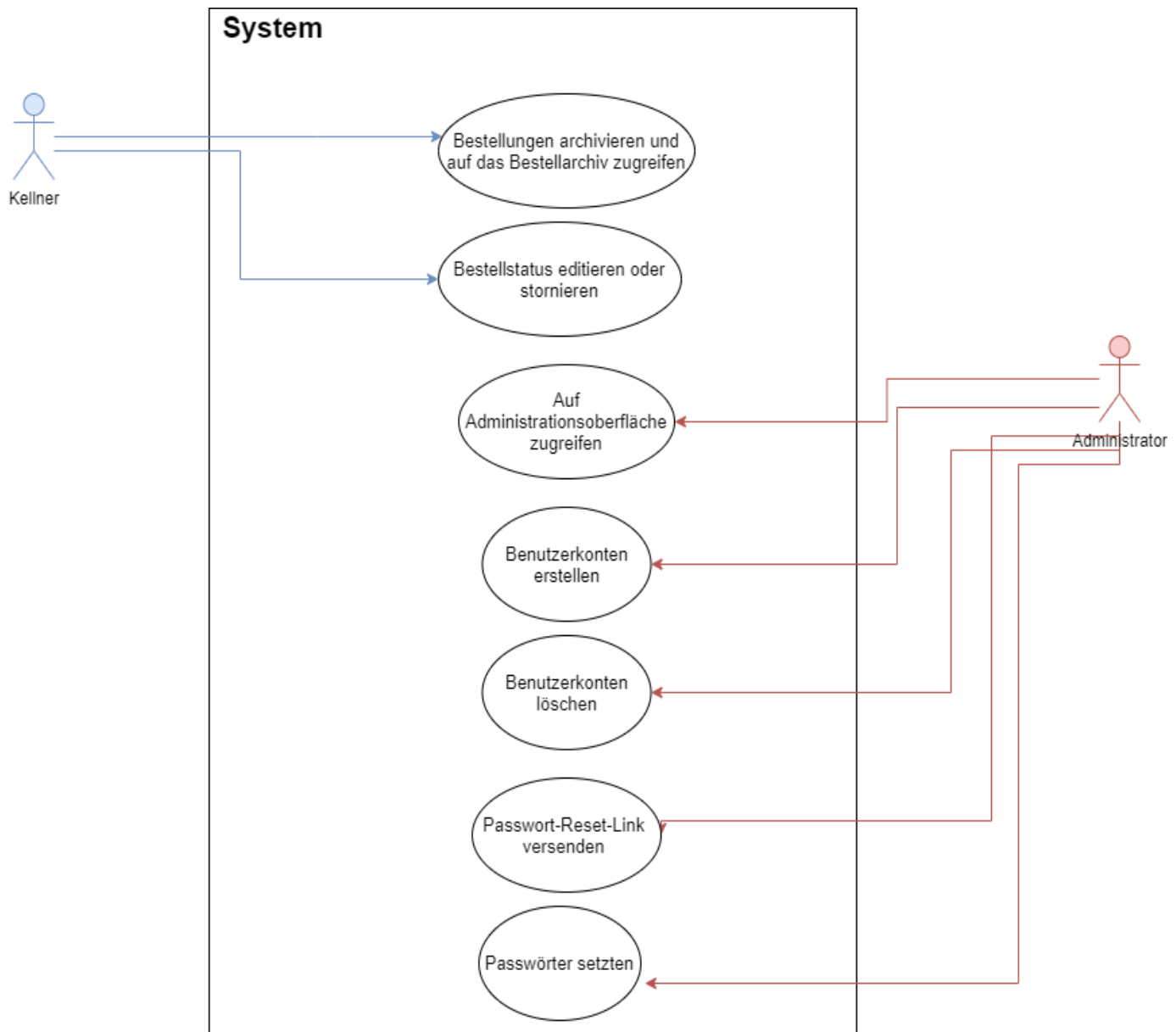


Abbildung 1 Use Case Diagramm - Kellner und Administrator

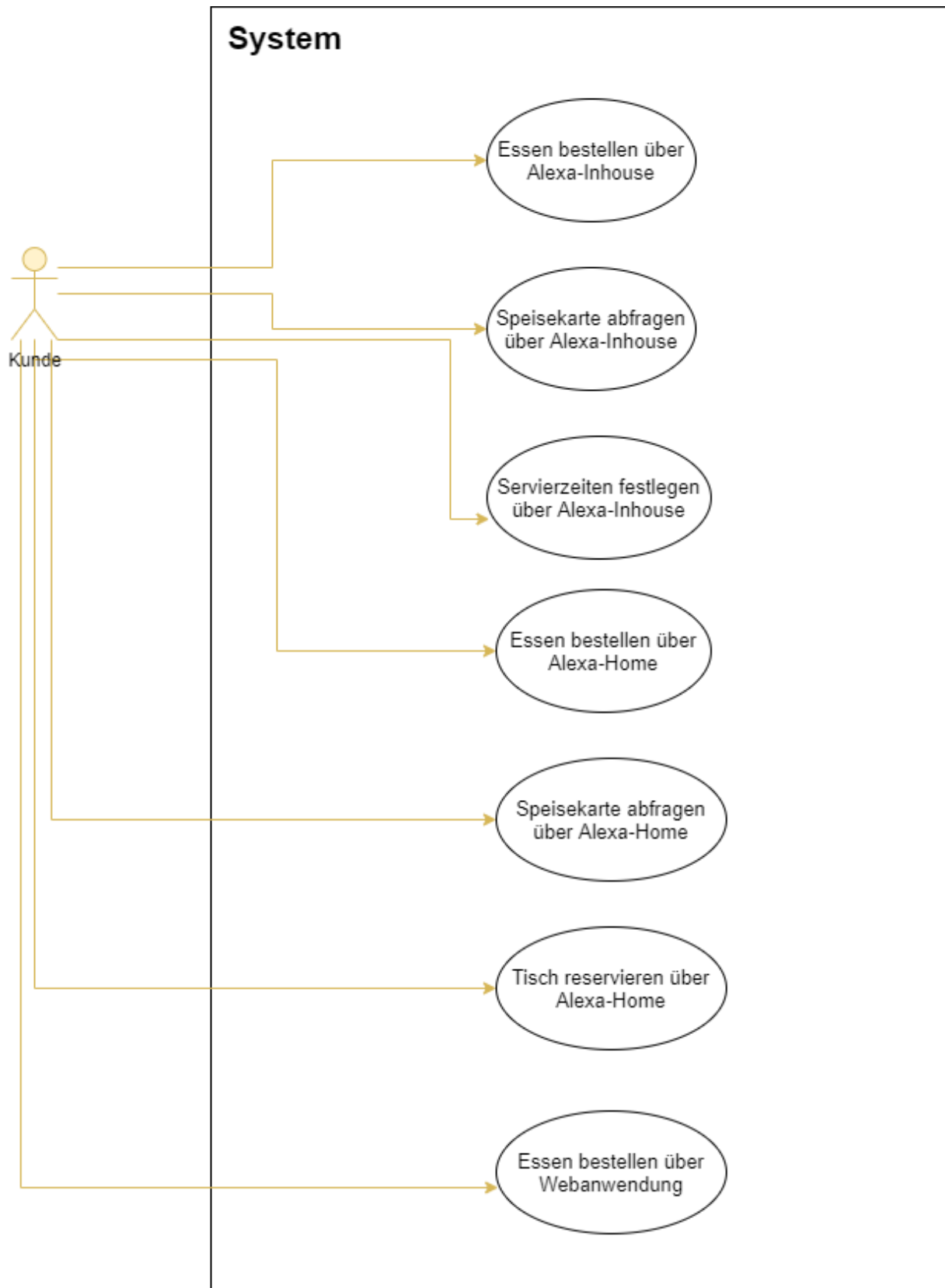


Abbildung 2 Use Case Diagramm - Kunde

## 1.2 Qualitätsziele

Für die Weiterentwicklung und Erweiterung des „My-Thai-Stars“-Systems sind gewisse Qualitätsmerkmale entscheidend, um dem Anwendungsfall des Systems gerecht zu werden.

Die folgende Tabelle beschreibt die zentralen Qualitätsziele der „My-Thai-Star“-Applikation.

Qualitätsziel	Motivation und Erläuterung
Sicherheit	Das System soll verhindern, dass jeder Nutzer unberechtigten Zugriff auf Nutzerdaten hat. Jede Nutzerrolle hat seine eigenen Rechte. Nur der Administrator hat Zugriff auf die Nutzerkonten und kann diese verändern.
Benutzbarkeit	Das „My-Thai-Star“-System soll für jeden Kunden leicht zu bedienen sein. Das Kellner-Cockpit und die Administrationsoberfläche sollen für den Kellner und Administrator einfach zu bedienen sein.
Verfügbarkeit	Das System soll jederzeit erreichbar sein und ein Mehrbenutzerbetrieb ermöglichen. Auftretende Fehler dürfen das System nicht zum Absturz bringen.

## 2. Randbedingungen

### 2.1 Technische Randbedingungen

Framework	Es wurde vom Kunden vorgegeben, das devonFw-Framework und damit verbundenen Technologien und Programmiersprachen zu verwenden. Darunter wird für das Frontend das Angular Framework und für das Backend das Spring Framework verwendet.
Programmiervorgaben	Die Implementierung soll in Java (Backend) und TypeScript (Frontend) erfolgen.
Weboberfläche	Die Weboberfläche soll mit der Verwendung von der Softwareentwicklungsplattform <a href="#">devonfw</a> entwickelt werden. Es wurden keine speziellen Anforderungen an das UI oder in Hinblick auf Usability vom Auftraggeber gestellt.
Sprachsteuerung	Bei der Implementierung der Sprachsteuerung ist durch den Kunden die Verwendung von Amazon Alexa vorgeschrieben.
On-Premise Lösung	Der Kunde hat die Anforderung gestellt, dass die Anwendung in einer On-Premise Lösung (siehe Folie 24 in [3] im Quellenverzeichnis) realisiert werden soll. Das heißt, dass die Anwendung lokal auf dem eigenen Rechner direkt und ohne zusätzliche Komplikationen laufen soll.
Datenbankanbindung und Persistenz	Die H2-Datenbank wurde aufgesetzt um die Daten zu speichern. Die Datenbank ermöglicht schnelle Zugriffsgeschwindigkeit und wird mit Hilfe der Spring Data JPA gespeichert.

### 2.2 Organisatorische Randbedingungen

Organisation und Struktur	Das Entwicklerteam besteht aus Sven Reinemuth, Tony Friedrich, Exoucia Mukubay, Florian Braasch und Maximilian Stinner. Weitere Stakeholder sind dem Projekthandbuch zu entnehmen.
---------------------------	--

### 2.3 Konventionen

Architekturdokumentation	Terminologie und Gliederung nach dem deutschen arc42-Template.
--------------------------	--



Sprache	Die Klassen, Methoden und Kommentare im Java- und TypeScript-Quellcode sollen auf Englisch sein.
---------	--

### 3. Kontextabgrenzung

Die folgende Unterkapitel zeigen die Einbettung unseres Systems in seine Umgebung.

#### 3.1 Fachlicher und Technischer Kontext

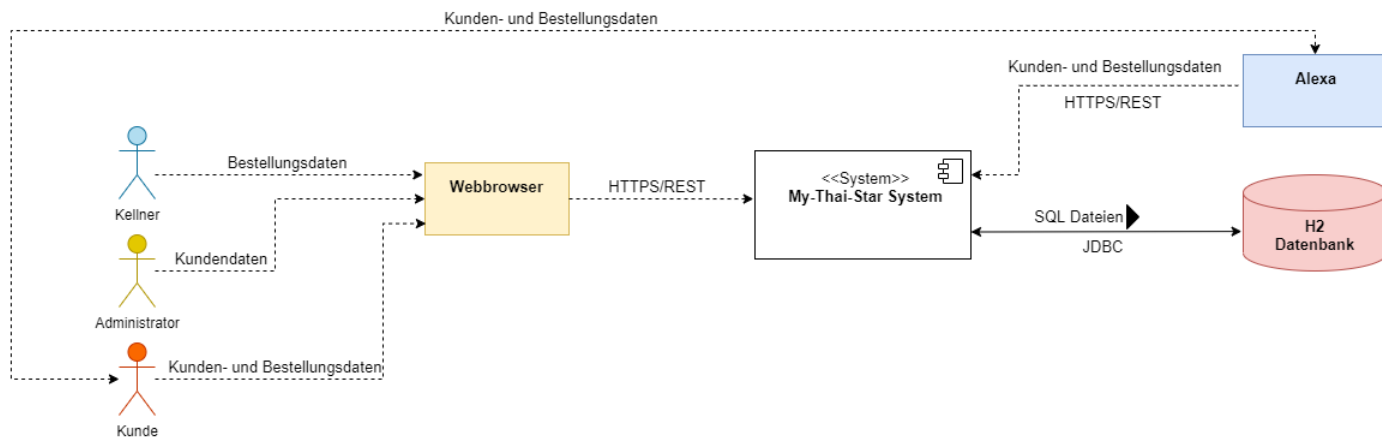


Abbildung 3 Kontextdiagramm des My-Thai-Star Systems

Die Abbildung zeigt die Einbettung des My-Thai-Star Systems in seine Umgebung mit den jeweils verwendeten Protokollen, Schnittstellen und Eingabeformaten.

Die Benutzer des Systems kommunizieren über den Webbrowser mit dem My-Thai-Star System. Der Webbrowser leitet die eingegebenen Daten über HTTP-Requests weiter an den Server (My-Thai Star System). Der Server implementiert die REST-Schnittstelle, um die http-Requests anzunehmen und zu verarbeiten. Nach der Verarbeitung der Requests wird eine http-Response zurückgeschickt. Der Server verarbeitet die http-Requests, indem er neue Daten in der Datenbank einfügt oder Daten aus der Datenbank holt. Die JDBC Schnittstelle, die der Server implementiert hat, ermöglicht den Zugriff auf die Datenbank. Am Ende wird die http-Response an den Webbrowser zurückgeschickt.

Findet eine Kommunikation zwischen Alexa und dem My-Thai-Star System statt, verläuft diese ebenfalls über die REST-Schnittstelle und es werden http-Requests und http-Response verschickt.

Das fachliche und technische Kontextdiagramm ist nach arc42 voneinander zu trennen. Wir haben uns dazu entschieden die Diagramme zusammenzufassen, weil die Elemente und Struktur beider Diagramme sich sehr ähnlich waren und gut kombiniert werden konnten.

Eine detaillierte Beschreibung des Deployments der Front- und Backendkomponente erfolgt im Abschnitt Verteilungssicht.

Der Benutzer „Manager“ und seine Aufgaben werden nicht in der Abbildung dargestellt, da sie nicht für das Projekt relevant sind und nicht zu den Anforderungen des Kunden zählen.

### 3.1.1 Tabellarische Auflistung der Nachbarsysteme und Benutzer

Tabelle 1 Auflistung der Nachbarsysteme und Benutzer

Nachbarsystem /Benutzer	Beschreibung	Input	Output
Kellner	Der Kellner hat eine Übersicht aller Bestellungen und kann den Bestellstatus ändern.	Bestellungs- und Tischreservierungsdaten, Status einer Bestellung	Übersicht über alle Bestellungen und Reservationen, Ansicht aller archivierten Bestellungen
<b>Administrator</b>	Der Administrator kann Nutzer erstellen, löschen und ihre Rollen zuweisen oder ändern.	Kundendaten	Benutzeransicht
Kunde	Der Kunde hat die Möglichkeit über Alexa oder über den Webbrowser Kunden- und Bestellungsdaten an das System senden.	Kunden- und Bestellungsdaten	Buchungsbestätigung per E-Mail, eigene Benutzeransicht, Speisekarte
Alexa	Das externe System besitzt eine Amazon-Developer Webseite, das ermöglicht Spracheingaben zu tätigen. Das System sendet http-Requests direkt zum My-Thai-Star System (zum Backend).	Kunden- und Bestellungsdaten	Speisekarte, Bestellsbestätigung, Buchungsbestätigung einer Tischreservation
H2 Datenbank	Die Datenbank verwaltet die Kunden- und Bestellungsdaten. Bei Anfragen sendet die Datenbank Daten an das "My-Thai-Star"-System	SQL-Anweisungen, um Daten wie Buchungen, Bestellungen und Nutzer hinzuzufügen, zu verändern oder zu löschen	Kunden- und Bestellungsdaten

## 4. Lösungsstrategien

Die folgende Tabelle stellt die Qualitätsziele von My-Thai-Star (siehe Abschnitt [Qualitätsziele](#)) den passenden Architekturansätzen gegenüber, und erleichtert so einen Einstieg in die Lösung.

Qualitätsziel	Motivation und Erläuterung	Lösungsstrategien
Sicherheit	Das System soll verhindern, dass jeder Nutzer unberechtigten Zugriff auf Nutzerdaten hat. Jede Nutzerrolle hat seine eigenen Rechte. Nur der Administrator hat Zugriff auf die Nutzerdaten und kann diese verändern.	<ul style="list-style-type: none"><li>- Nutzerrollen und ihre Rechte festlegen</li><li>- Administrator darf nicht jeden Nutzer löschen</li></ul>
Benutzbarkeit	Das My-Thai-Star System soll für jeden Kunden leicht zu bedienen sein. Das Kellner-Cockpit und die Administrationsoberfläche sollen für den Kellner und Administrator einfach zu bedienen sein.	<ul style="list-style-type: none"><li>- Usability Tests</li></ul>
Verfügbarkeit	Das My Thai Star soll in einer Cloud-Umgebung über Kubernetes realisiert werden.	<ul style="list-style-type: none"><li>- Kubernetes Cluster</li></ul>

## 5. Bausteinsicht

Dieser Abschnitt beschreibt die Zerlegung des My-Thai-Star Systems in Komponenten.

Die statische Darstellung des My-Thai-Star Systems in seinen Komponenten werden nur bis zur zweiten Ebene gezeigt. Weitere Ebenen werden nicht gezeigt, da das Dokument sonst zu umfangreich wäre und durch die zwei Ebenen schon alle architekturelevanten Details herausgearbeitet sind. Die Strukturen und Abhängigkeiten zwischen den einzelnen Komponenten sind sichtbar und würden bei einer weiteren Verfeinerung nur komplexer und unübersichtlicher werden, da eine gute Darstellungsart von so vielen Komponenten in dieser Dokumentform schwer umsetzbar ist.

## 5.1 Ebene 1 - Gesamtsystem

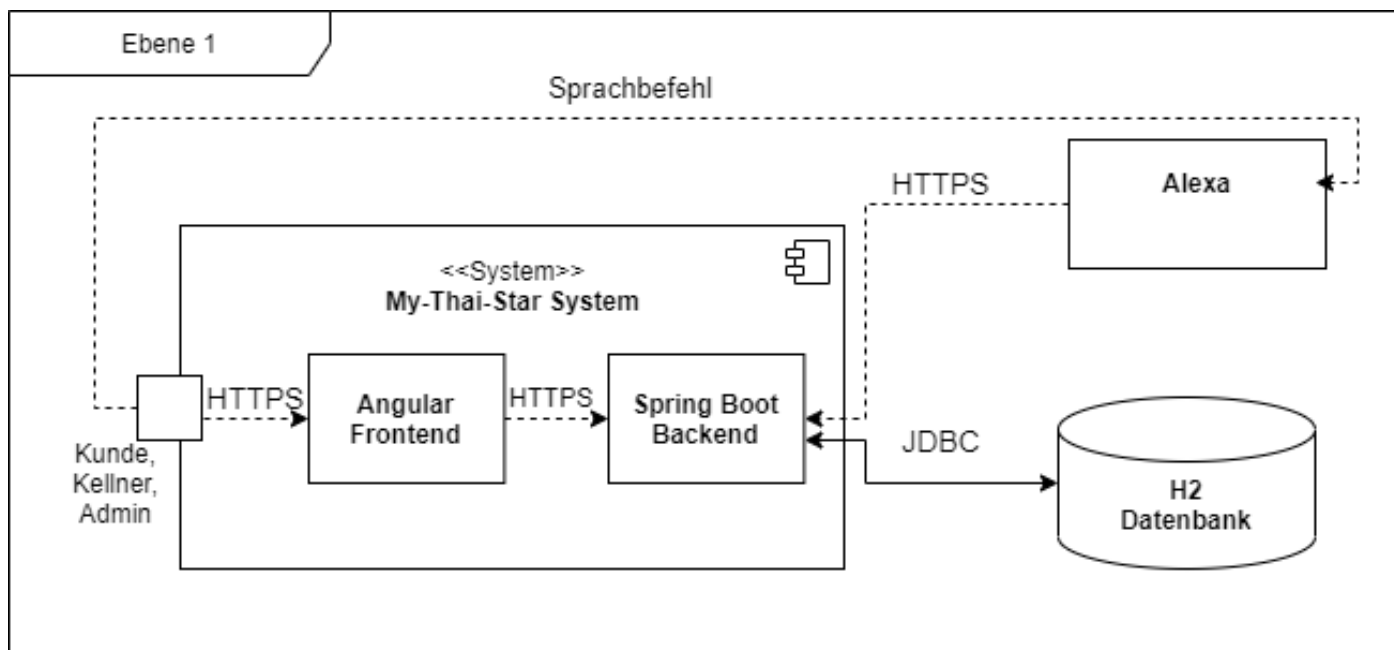


Abbildung 4 Bausteinsicht des Gesamtsystems - Ebene 1

Abbildung 4 zeigt das Gesamtsystem von My-Thai -Star in der ersten Ebene. Das My-Thai-Star System besteht aus dem Angular Frontend und dem Spring Boot Backend. Hier wird die Kommunikation des Gesamtsystems mit externen Komponenten wie dem Webbrowser, Alexa und der H2 Datenbank dargestellt. Der Webbrowser leitet die vom User geforderten http-Requests an das Frontend weiter, das wiederum auf die Daten im Backend zurückgreift, welches die http-Request entgegennimmt und gegebenenfalls mit einer http Response antwortet. Die Datenhaltung geschieht über eine H2 Datenbank.

## 5.2 Ebene 2 - Frontend

Das folgende Diagramm zeigt die statische Darstellung des Frontend mit seinen Komponenten in der zweiten Ebene.

Die Komponente Frontend basiert auf Angular und bietet verschiedenen Komponenten an, die für die Verarbeitung und Darstellung der Daten zuständig sind.

Der Benutzer des Systems navigiert, abhängig von seiner Rolle (Kunde, Kellner oder Administrator), über verschiedene Ansichten in der Webapplikation. Je nach gewünschter Aktion des Benutzers werden die entsprechenden Komponenten angesprochen und über das HTTP Client schließlich eine Anfrage an das Backend gesendet.

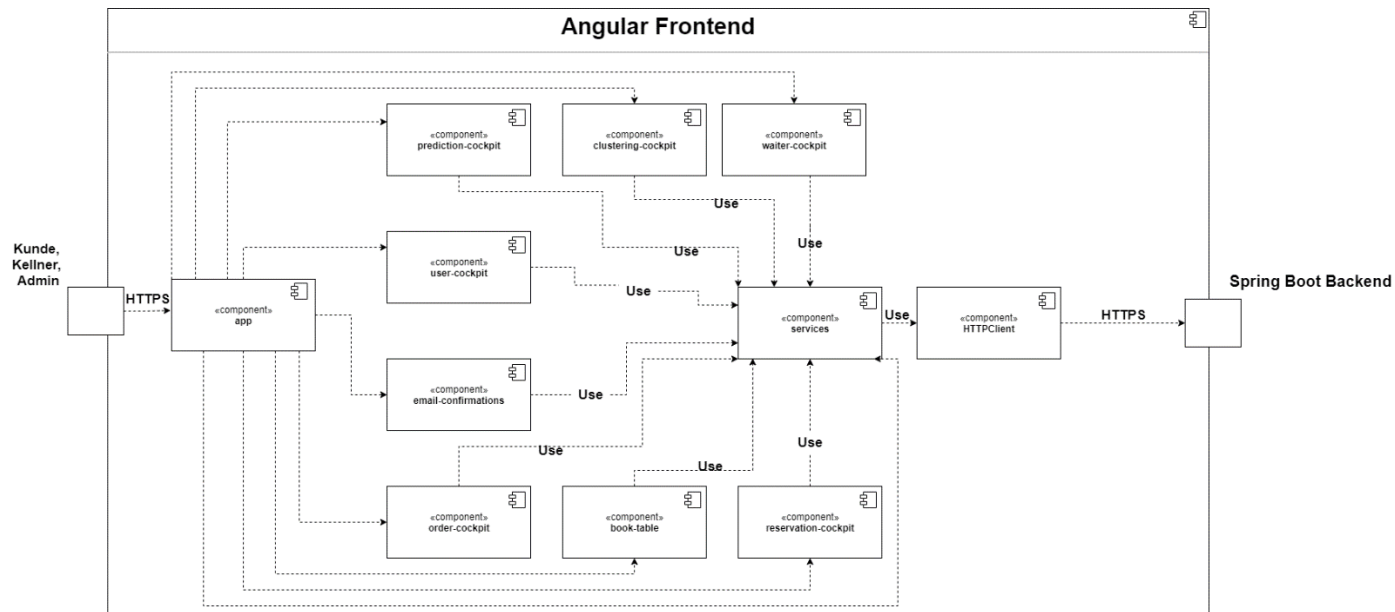


Abbildung 5 Bausteinsicht des Angular Frontend – Ebene 2

Die Verbindung zwischen dem Frontend und Backend verläuft über eine REST-Schnittstelle, die von den Komponenten „*HTTPClient*“, im Frontend, und „*RestService*“, im Backend, implementiert wird. Das Versenden der Daten vom Frontend an das Backend verläuft über eine Auswahl von http-Anfragemethoden.

Die Komponente „*app*“ stellt als Hauptkomponente im Angular dar. Die restlichen Komponenten nutzen die Funktionalitäten der Komponente „*services*“ aus und übernehmen unterschiedliche Aufgaben.

## 5.3 Ebene 2 - Backend

Das folgende Diagramm zeigt die statische Darstellung des Backend mit seinen Komponenten in der zweiten Ebene.

Die Komponente Backend basiert auf Spring Boot. Eingehenden http-Requests vom Frontend oder von Alexa werden an die passende Komponente weitergegeben und verarbeitet. Die Verarbeitung der Daten erfolgt über die einzelnen Komponenten, die anschließend die Daten in der H2 Datenbank speichern.

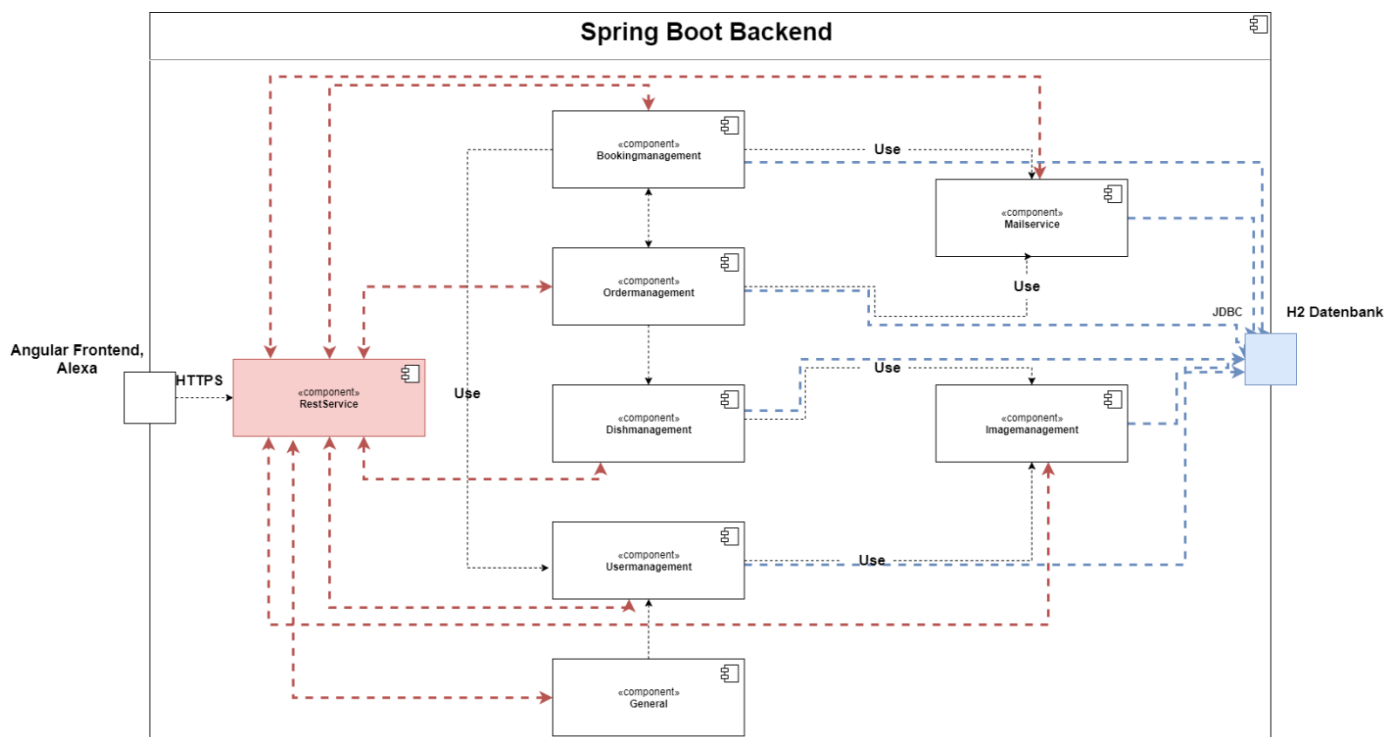


Abbildung 5 Bausteinsicht des Spring Boot Backend – Ebene 2

Die Komponente „RestService“ hat die Aufgabe, eingehende Anfragen (http-Requests), an die jeweilige Komponente weiterzuleiten. Dafür ruft die Komponente die Methoden der zugehörigen Komponente auf, die die Anfrage verarbeitet.

Die Hauptkomponenten in Spring Boot sind „Bookingmanagement“, „Dishmanagement“, „Ordermanagement“ und „Usermanagement“. Diese nutzen untereinander die Funktionalitäten der jeweiligen Komponente aus. Die Komponente „General“ nutzt die Komponente „Usermanagement“, um auf die Benutzerdaten zuzugreifen. Des Weiteren gibt es die gemeinsamen Komponenten, „Mailservice“ und „Imagemanagement“, welche die Funktionalitäten bereitstellen, die von den Hauptkomponenten verwendet werden können.

Zusätzlich gibt es noch die Komponenten „Clustermanagement“ und „Predictionmanagement“, die nicht in der Abbildung dargestellt werden, da sie für das Projekt nicht relevant sind.

## 5.4 Ebene 2 – Alexa

Das folgende Komponentendiagramm zeigt einen statischen Ausschnitt von Alexa.

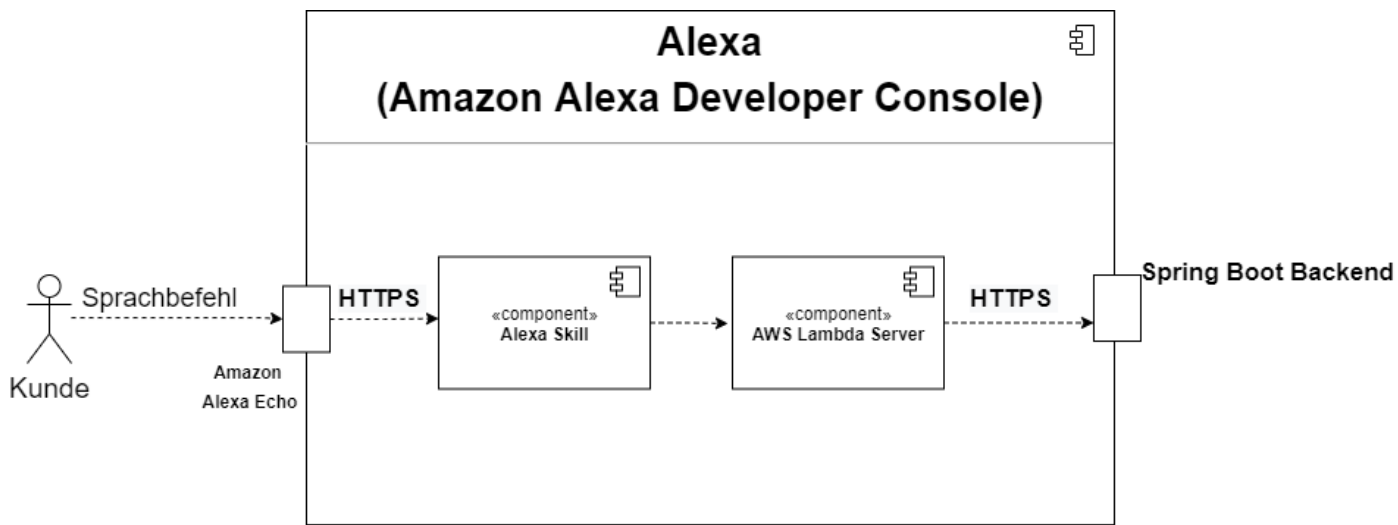


Abbildung 6 Bausteinsicht von Alexa - Ebene 1

Der Kunde tätigt Sprachbefehle, die dann per http-Requests zum Amazon Alexa Echo Gerät erfolgen. Daraufhin aktiviert das Gerät der dazugehörige Alexa Skill. Der Alexa Skill muss eine Programmlogik ausführen, um eine Antwort an den Kunden zu liefern. Dafür nutzt der Alexa Skill den AWS Lambda Server, der den Programmcode ausführt. Der AWS Lambda Server sendet http-Requests, um auf die Daten des Spring Boot Backend zuzugreifen, dessen Antwort wiederum falls gefordert verarbeitet und an dem Kunden weitergegeben wird

## 6. Laufzeitsicht

In diesem Kapitel werden durch zwei Sequenzdiagramme die wichtigsten Abläufe im Zusammenspiel mit den Komponenten in dem System dynamisch dargestellt. Der vorkommende Akteur ist der Kunde.

Nach unserem Ermessen reichen die zwei Sequenzdiagramme aus, um die Abläufe innerhalb zweier Systeme zu beschreiben. Die Nutzer Registration ist dabei Beispiel für einen Vorgang im My Thai Star-System, bei dem sowohl Front- und Backend angesprochen werden. Alle anderen Vorgänge des Systems sind dem Diagramm sehr ähnlich. Das Alexa-Sequenzdiagramm ist Beispiel für einen Alexa-vorgang. Alle anderen Alexa-Vorgänge verlaufen nach diesem Schema.

### 6.1 Sequenzdiagramm 1: Nutzer Registration

Das folgende Sequenzdiagramm zeigt den Ablauf der Nutzerregistration eines Kunden.



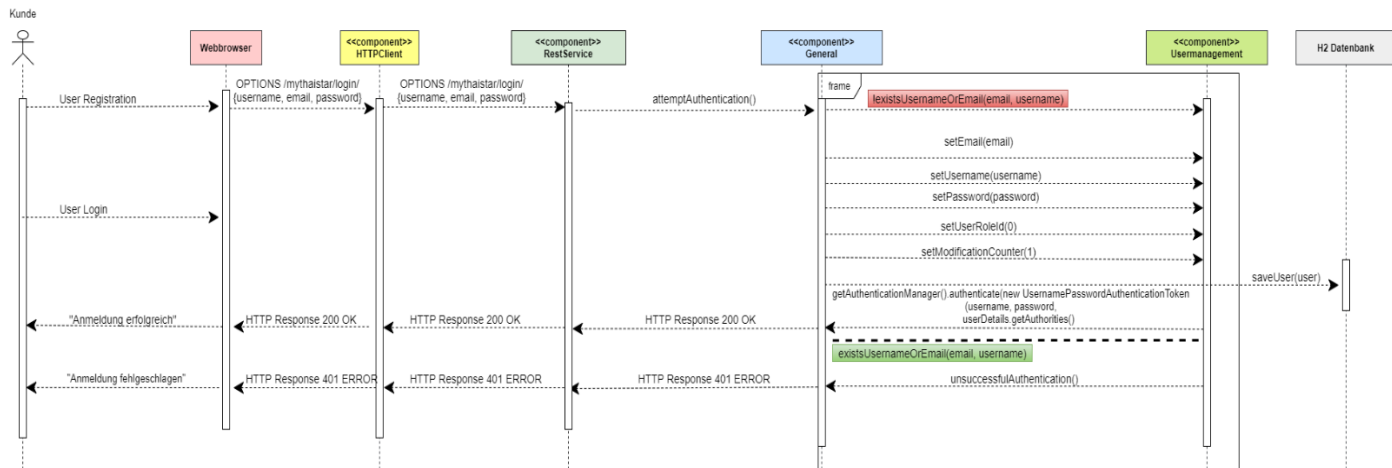


Abbildung 7 Sequenzdiagramm - Nutzer Registration

Der Kunde registriert sich über den Webbrowser und gibt seine Daten ein. Der Webbrowser leitet die Anfrage direkt zur Angular Komponente „HTTPClient“, dieser wiederum die Anfrage an die Komponente „RestService“, im Backend, weiterleitet. Die Komponente „RestService“ leitet die eingetroffenen Daten zunächst an die Komponente „General“, der zunächst die Benutzerdaten authentifiziert. Danach wird in der Komponente „Usermanagement“ geprüft, ob der Username oder die E-Mail in der Datenbank schon gespeichert ist. Falls dies nicht zutrifft, so werden die Daten des Kunden in der Datenbank gespeichert und der Nutzer wird automatisch angemeldet. Ist der Benutzername oder die E-Mail schon vergeben, so schlägt die Registration fehl.

Im Abschnitt [Glossar](#) werden die http Anfragemethoden wie OPTIONS genauer beschrieben.

## 6.2 Sequenzdiagramm 2: Tisch reservieren über Alexa-Home

Im folgenden Diagramm wird der Ablauf dargestellt, wie der Kunde ein Tisch über Alexa-Home reservieren möchte.

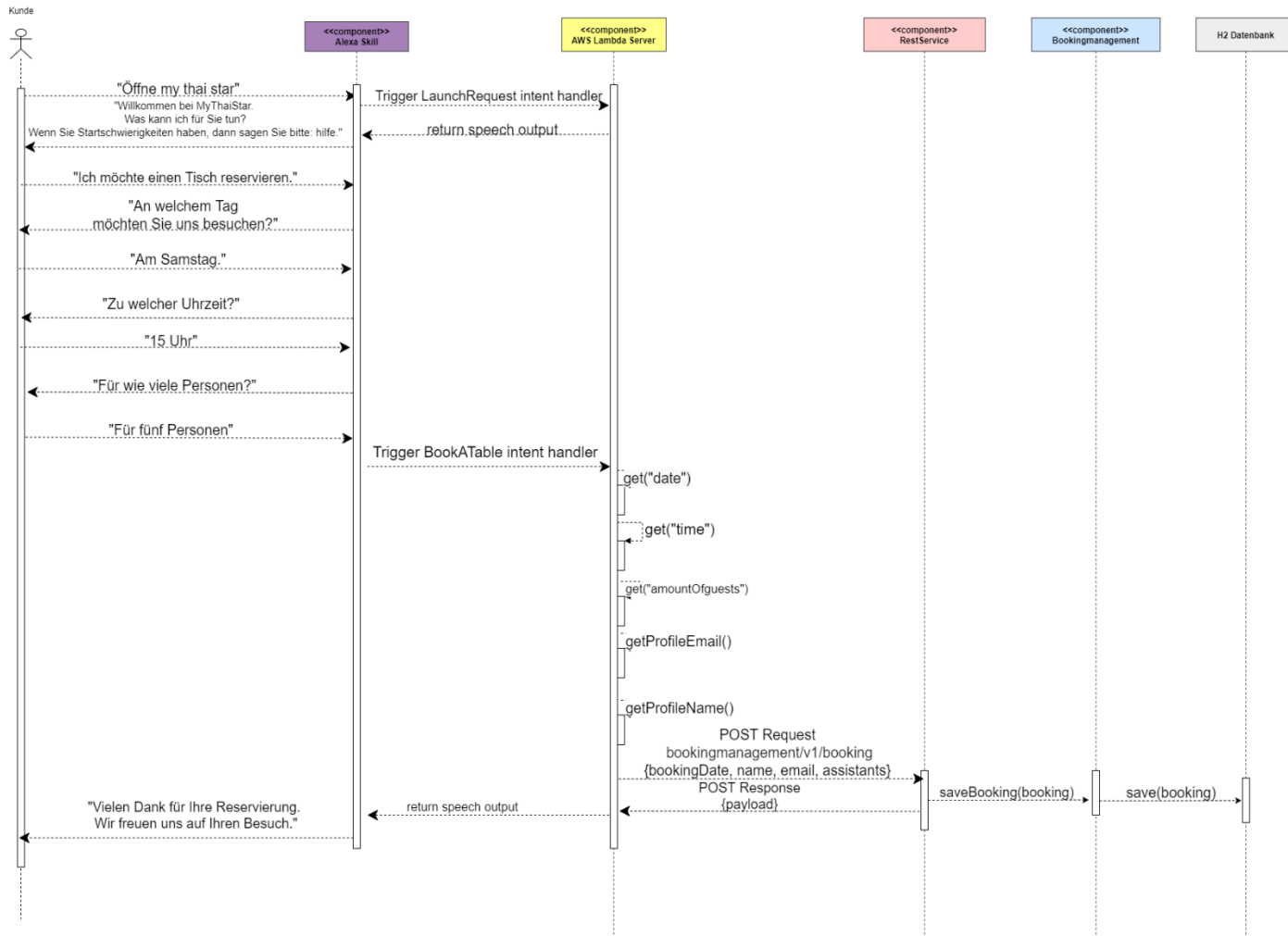


Abbildung 8 Sequenzdiagramm Tisch reservieren über Alexa-Home

Der Kunde gibt die Sprachanweisung „Öffne my thai star“ ein, die das dazugehörige *Alexa Skill* aufruft. Nach Aktivierung des dazugehörigen Skills erwartet der Skill einen bestimmten Intent des Kunden.

Der Intent Handler bewirkt, dass der *Alexa Skill* dem Kunden antwortet und nach den drei Parametern (Tag, Uhrzeit und Anzahl der Gäste) fragt. Diese drei Parameter werden über den BookATable Intent Handler an den *AWS Lambda Server* übergeben. Der *AWS Lambda Server* ruft sich selbst Methoden auf, um die Parameter zu holen und leitet diese über http-Request an das Backend des My-Thai-Star-System weiter. Die Komponente „*RestService*“ empfängt den http-Requests und ruft die saveBooking() Methode der Komponente „*Bookingmanagement*“ auf, damit die Speicherung der Tisch Reservation in der Datenbank erfolgt wird.

Die Parameter werden in der Datenbank aufgenommen und gespeichert. Zum Schluss meldet der *Alexa Skill* dem Kunden, dass die Reservierung erfolgreich aufgenommen wurde.

## 7. Verteilungssicht

Das vorliegende Verteilungsdiagramm zeigt die physische Verteilung des „My-Thai-Star“ Systems.

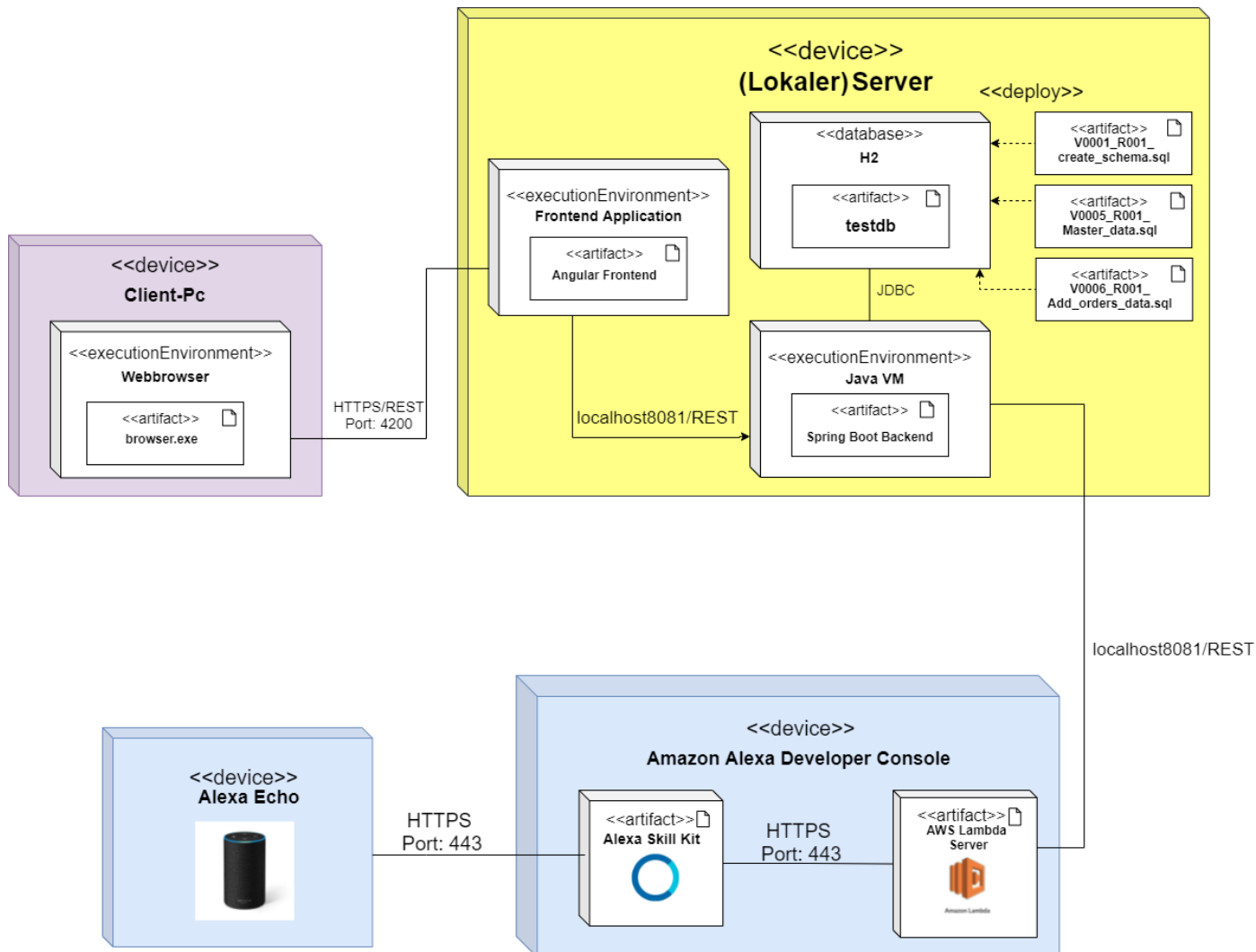


Abbildung 9 Verteilungsdiagramm des My-Thai-Star Systems

Das Gesamtsystem My-Thai-Star umfasst eine Webanwendung, welche die On Premise-Lösung ( [2.1 Technische Randbedingungen](#)) umsetzt. Der Server läuft lokal auf dem PC.

Der Webbrowser greift auf das *Angular Frontend* über die http-Anfragemethode auf Port 4200 über die REST-Schnittstelle, welches die angefragten Ressourcen darstellt. Die Verbindung zwischen dem *Angular Frontend* und *Spring Boot Backend* verläuft über die REST-Schnittstelle (siehe [5.2 Ebene 2 - Frontend](#)). Das Backend des Systems wird durch *Spring Boot Backend* realisiert und kommuniziert mit der Datenbank über die JDBC Schnittstelle. Beim Serverstart werden die SQL-Dateien in der Datenbank deployt.

Im Geräteknoten *Amazon Alexa Developer Console* ist der *Alexa-Skill-Kit* und der *AWS Lambda Server* deployt. Möchte der User eine Interaktion mit Alexa durchführen, so leitet das *Alexa Echo* Gerät die Sprachanfrage über https auf Port 443 an den *Alexa Skill Kit* weiter. Daraufhin wird der *Alexa Skill* identifiziert und aktiviert. Der *AWS Lambda Server* erhält die http-Anfrage auf Port 443 und führt die Programmlogik aus und stellt die Daten für den *Alexa-Skill-Kit* zur Verfügung. Über die REST Schnittstelle hat der *AWS Lambda Server* Zugriff auf die Daten des Spring Boot Backends.

Die durchgezogene Linie zwischen den Komponenten stellt die Kommunikation zwischen zwei Knoten dar und

die gestrichelte Linie mit einem Pfeil am Ende, weist daraufhin, dass ein Knoten oder eine Komponente von einer anderen abhängig ist.

## 8. Entwurfsentscheidungen und Ihre Auswirkungen

Das Kapitel bezieht sich auf die fundamentalen Entscheidungen von hoher Wichtigkeit, die die Architektur und den Aufbau des Systems betreffen.

### 8.1 Wie soll Alexa mit unserem lokalen My-Thai-Star Webserver kommunizieren?

In der ersten Sichtung des „My-Thai-Star“ Projektes wurde festgestellt, dass die Ausführung der Software und dessen Front- sowie Backend ausschließlich über den lokalen Server, „localhost“, ansprechbar ist. Um nun jedoch eine Verbindung mit unseren Alexa-Skill realisieren zu können wurden zwei Alternativen, um dieses Problem zu lösen betrachtet. Die beiden zunächst betrachteten Lösungen waren zum einen eine Docker-Container Lösungen und zum anderen eine „Tunnel“-Lösung unter Verwendung von Ngrok.

#### 8.1.1 Alternative 1: Docker-Container

Docker ist eine Software zur Isolierung von Anwendung mit Hilfe von Container Virtualisierung. Docker vereinfacht die Bereitstellung von Anwendungen, weil sich Container, die alle nötigen Pakete enthalten, leicht als Dateien transportieren und installieren lassen. In Bezug auf unser Projekt also unser Front- und Backend über einen Öffentlich für Alexa zugänglichen Docker Server aufzusetzen.

#### 8.1.2 Alternative 2: Ngrok „Tunnel“

Ngrok ist eine zu installierende Anwendung die einen lokalen Webserver, über den von ngrok gestellten Cloudservice über eine öffentliche Adresse aufrufbar macht und über die lokale Anwendung auf den „localhost forwarden“.

Folgende Abbildung verdeutlicht die Funktionalitäten des ngroks:

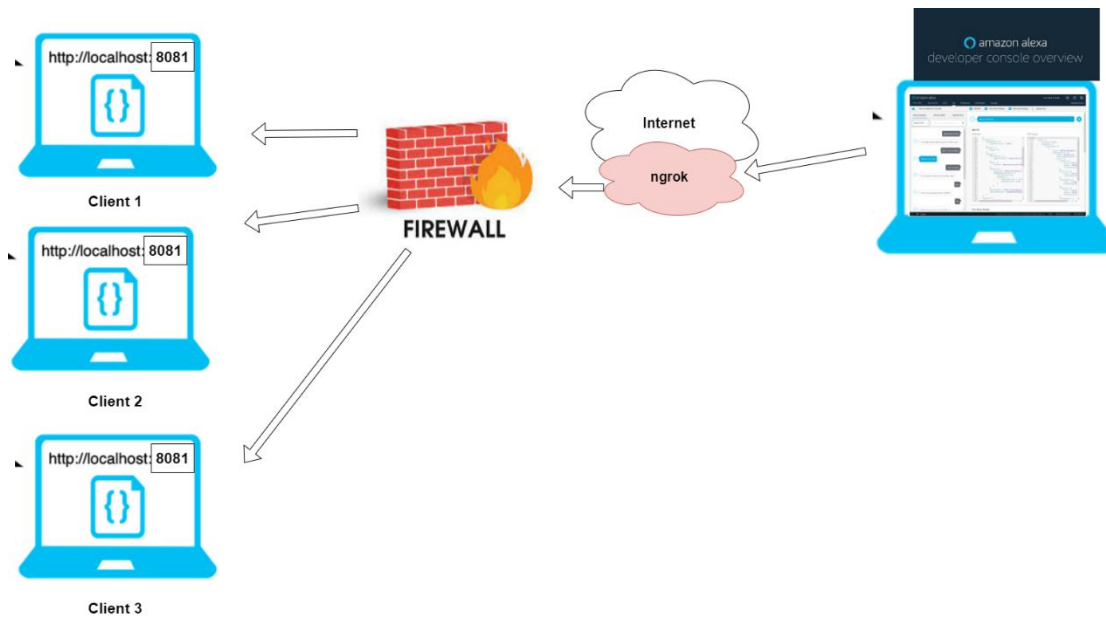


Abbildung 10 Funktionsweise des Ngroks

#### Entscheidung:

In der Recherche der beiden Möglichkeiten wurde die Umsetzbarkeit von Ngrok besser eingeschätzt und sollte auch schneller funktionsfähig sein. Da wir eine Möglichkeit benötigten unseren Vertikalen Durchstich zu realisieren haben wir uns für die zweite Alternative entschieden.

Während des Entwicklungsprozesses wird diese Entscheidung in Absprache mit den Auftraggebern, getroffen. Es wird sichergestellt, dass die Entscheidungen und deren Bedeutung verstanden haben und diese nachvollziehbar getroffen wurden.

Durch eine neue Muss-Anforderung, der am 21. Mai 2021 geäußert wurde, werden wir uns mit einer Docker-Container Lösung auseinandersetzen. Die geplante Umsetzung wird aktuell über Kubernetes-Cluster geplant und nach einem Proof-of-Concept auch auf einen Server realisiert werden können.

## 9. Software Glossar

<b>H2 Datenbank</b>	<b>Datenbankserver, der Datenbanksprachen darunter SQL verwendet.</b>
<b>Eclipse</b>	Eclipse ist ein open-source Programmierwerkzeug zur Entwicklung von Software verschiedener Art.

## 10. Glossar

<b>Alexa</b>	<b>Sprachgesteuerte Software von dem Unternehmen Amazon, welches verschiedene</b>
--------------	---

	<b>Aufgaben erledigen kann oder verschiedene Systeme steuert.</b>
<b>Alexa-Skill</b>	Alexa Skills erweitern Alexa um weitere Funktionen. Dabei handelt es sich um vorgefertigte Szenarien, die auf einen definierten Befehl eine entsprechende Aktion folgen lassen.
<b>Amazon Alexa Developer Console</b>	Die Amazon Alexa Developer Console bietet die Erstellung, Verwaltung und Veröffentlichung der Alexa Skills.
<b>Angular</b>	Ist ein TypeScript-basiertes Front-End-Webapplikationsframework.
<b>Arc42</b>	Arc42 besteht aus einem Template zur Entwicklung, Dokumentation und Kommunikation von Softwarearchitekturen.
<b>Artefakt</b>	In UML-Modellen stellen Artefakte Modellelemente dar, die für die physischen Entitäten in einem Softwaresystem stehen. Artefakte stellen physische Implementierungseinheiten dar, z. B. ausführbare Dateien, Bibliotheken, Softwarekomponenten, Dokumente und Datenbanken.
<b>AWS Lambda Server</b>	AWS steht für Amazon Web Services. AWS Lambda ist ein serverloser Datenverarbeitungsservice, der den Code beim Eintreten bestimmter Ereignisse ausführt.
<b>BookATable Intent Handler</b>	Übergibt Parameter, die für eine Tischbuchung nötig sind, an den AWS Lambda Server.
<b>CRUD</b>	<p>CRUD fasst die Funktionen zusammen, die ein Nutzer benötigt, um Daten anzulegen und zu verwalten.</p> <ul style="list-style-type: none"> <li>▪ <b>Create</b> (Datensatz anlegen)</li> <li>▪ <b>Read bzw. Retrieve</b> (Datensatz lesen)</li> <li>▪ <b>Update</b> (Datensatz aktualisieren)</li> <li>▪ <b>Delete bzw. Destroy</b> (Datensatz löschen)</li> </ul>
<b>Devon-Framework/devonfw</b>	Ist ein Softwareentwicklungsplattform, welche eine Reihe von Technologien und Best-Practises enthält und bereitstellt.
<b>Execution Environment</b>	Der Ausführungsumgebungsknoten ist Computersystem in einem Geräteknoten. Dabei kann es sich um ein Betriebssystem, eine Java Virtual Machine .
<b>Front-/Backend</b>	Das Backend bezieht sich auf Teile einer Computeranwendung oder eines Programmcodes, die den Betrieb des Programms ermöglichen und auf die ein Benutzer nicht zugreifen kann. Die Schicht oberhalb des Backends ist das Frontend und

	umfasst die gesamte Software oder Hardware, die Teil einer Benutzerschnittstelle ist.
<b>Geräteknoten</b>	Geräteknoten sind Computing-Ressourcen, die Prozesse verarbeiten und Programme ausführen können. Gängige Beispiele für solche Knoten sind PCs, Laptops und Mobilgeräte.
<b>H2 Datenbank</b>	Datenbankserver, der Datenbanksprachen darunter SQL verwendet.
<b>HTTPS</b>	Hypertext Transfer Protocol Secure ist ein Kommunikationsprotokoll im World Wide Web, mit dem Daten abhörsicher übertragen werden können.
<b>http Anfragemethoden</b>	Im http-Protokoll gibt es verschiedene Anfragemethoden, die es dem Browser ermöglichen, Informationen, Formulare oder Dateien an den Server zu senden. <b>GET:</b> Ressourcen vom Server anfordern <b>POST:</b> große Datenmengen zur weiteren Verarbeitung zum Server senden. <b>OPTIONS:</b> liefert eine Liste der vom Server unterstützten Methoden und Merkmale
<b>http-Request, http-Response</b>	Im HTTP-Protokoll gibt es verschiedene Anfragemethoden, die es dem Browser ermöglichen, Informationen, Formulare oder Dateien an den Server zu senden. Der <b>http-Request</b> stellt eine Anfrage dar, die ein Nutzer an einen verbundenen Server sendet. Als Antwort auf einen derartigen Request sendet der Server eine <b>http-Response</b> .
<b>Intents</b>	Die Intents stellen dar, was der Nutzer von dem Alexa Skill erwartet. Der Alexa Skill könnte, zb. Eine Reise planen, einen Status abrufen, einen Witz erzählen, sein. Dies sind alle Intents.
<b>JDBC</b>	Java Database Connectivity (JDBC) ist eine treiberbasiert arbeitende Datenbankschnittstelle für Java. Das API erlaubt den Zugriff auf unterschiedliche relationale Datenbanken und ermöglicht die Ausführung von SQL-Anweisungen.
<b>Kellner-Cockpit</b>	Das Kellner-Cockpit ist eine Ansicht, in der Kellner des Restaurants auf alle wichtigen Informationen zu Bestellungen zugreifen können.
<b>Kubernetes</b>	Kubernetes ist ein Open-Source-System zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von Container-Anwendungen.
<b>Kubernetes-Cluster</b>	Kubernetes-Clustern können Container über mehrere Rechner hinweg und in mehreren



	<p>Umgebungen ausgeführt werden: virtuell, physisch, Cloud-basiert und On-Premises.</p> <p>Ein Kubernetes-Cluster ist eine Gruppe von Knoten, auf denen containerbasierte Anwendungen ausgeführt werden. Dabei handelt es sich um Bündelungen einer Anwendung mit ihren Abhängigkeiten und einigen erforderlichen Services in einem Paket.</p>
<b>LaunchRequest Intent Handler</b>	Wenn der Nutzer den Alexa Skill aufruft, aber keinen bestimmten Befehl sagt zB. „Alexa, öffne Tageshoroskop“.
<b>ngrok</b>	Ngrok stellt lokale Server hinter NATs (Netzwerkadressübersetzung) und Firewalls über sichere Tunnel dem öffentlichen Internet zur Verfügung. Der Datenverkehr wird an den auf Ihrem Computer ausgeführten ngrok-Prozess und dann an die von Ihnen angegebene lokale Adresse (zb. My Thai Star Webseite) weitergeleitet.
<b>On-premise</b>	On-Premise beschreibt die Inbetriebnahme von Software auf eigener Hardware.
<b>Proof-of-Concept</b>	Proof of Concept (PoC) ein Meilenstein, an dem die prinzipielle Durchführbarkeit eines Vorhabens belegt ist.
<b>REST-API</b>	REST steht für Representational State Transfer, API für Application Programming Interface. Das Representational State Transfer beschreibt wie verteilte Systeme miteinander kommunizieren können.
<b>Spring</b>	Ist ein Open-Source-Framework für Java und hat das Ziel die Komplexität der Java-Plattform zu reduzieren. Es wird häufig genutzt um Webanwendungen in Java zu programmieren.
<b>Spring Data JPA</b>	Spring Data JPA zielt darauf ab, die Implementierung von Datenzugriffsschichten erheblich zu verbessern, indem der Aufwand auf den tatsächlich benötigten Betrag reduziert wird. Es erleichtert das Erstellen von Feder-basierten Anwendungen, die Datenzugriffstechnologien verwenden.
<b>SQL</b>	Datenbanksprache, mit der Datenbankabfragen möglich sind.
<b>Use Case</b>	Ein Use Case ist ein Anwendungsfall, der alle möglichen Szenarien, die eintreten können, wenn ein Akteur versucht, mit Hilfe des betrachteten Systems ein bestimmtes fachliches Ziel zu erreichen. Er beschreibt, was inhaltlich beim Versuch der Zielerreichung passieren kann und abstrahiert von konkreten technischen

	Lösungen. Das Ergebnis des Anwendungsfalls kann ein Erfolg oder Fehlschlag sein.
<b>Wartung</b>	Unter Wartung werden alle Maßnahmen zur vorbeugenden Instandhaltung verstanden.
<b>Software-Entwicklungsprojekt (SEP)</b>	Das Software – Entwicklungsprojekt ist ein von der Hochschule Mannheim vorgesehenes Semester, in dem Studierende an einem Projekt arbeiten und so ihr Wissen in der Praxis anwenden können.

## 11. Ergänzungen

Hier stehen die Verweisungen für nachträgliche Änderungen im Dokument.

- Kapitel 11 „Ergänzungen“ und 12 „Quellenverzeichnis“ hinzugefügt
- Kapitel 5 Inhalte ergänzt, 5.4 neues Komponentendiagramm für Alexa
- Kapitel 6
- Kapitel 3.1.1
- Kapitel 8.1.1 und 8.1.2

## 12. Quellenverzeichnis

[0] Peter Hruschka, Gernot Starke: arc42 Template Overview

[arc42 Template Overview - arc42](#)

(14.06.2021)

[1] Team Blobfish: Pflichtenheft V.2.0

<https://www.file-upload.net/download-14612964/Pflichtenheft-v2.0-TeamBlobfish.docx.html>

(14.06.2021)

[2] Capgemini S.E.: devonfw, The standard open source software development platform;

<https://devonfw.com/website/pages/welcome/welcome.html>

(14.06.2021)

[3] Capgemini S.E: Projektsemester Sommersemester 2021 HS Mannheim Projekt My Thai Star

<https://www.file-upload.net/download-14613015/2021HSMannheimMyThairStar.pdf.html>

(14.06.2021)