

WBA- Vorgehen / Systemdokumentation

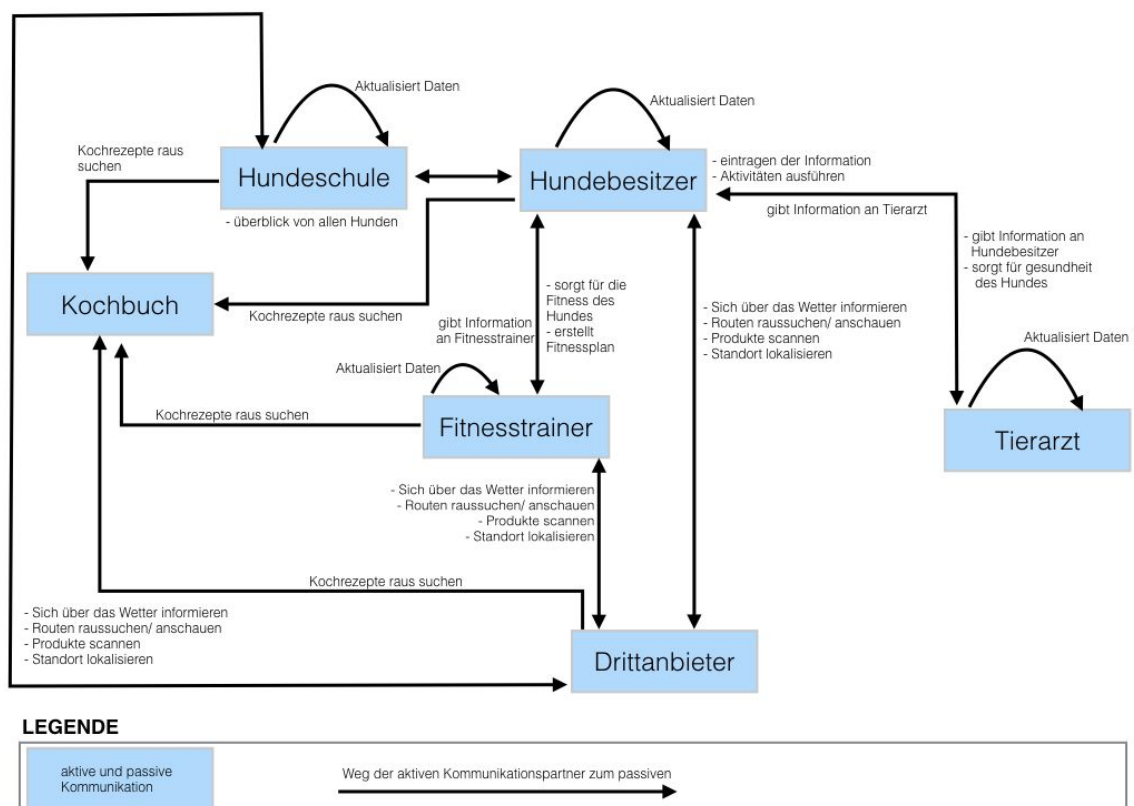
Nun folgt im nächsten Abschnitt eine Abwägung zu den gewählten Technologien. Dies bedeutet, dass Komponenten des System ermittelt und zusammenhänge werden gesucht und Daten und deren Datenmodelle werden analysiert. Ebenso soll die Ist-Situation der Kommunikationspartner ermittelt und eine SoLL- Situation entwickelt, sodass die Alleinstellungsmerkmal noch mehr heraussticht.

Kommunikationsmodell

Mit Hilfe von Grafiken wird der Kommunikationsmodell dargestellt und erläutert. Hierbei muss es deutlich sein, welche Information zwischen Nutzern über das System ausgetauscht werden.

Deskriptives Modell

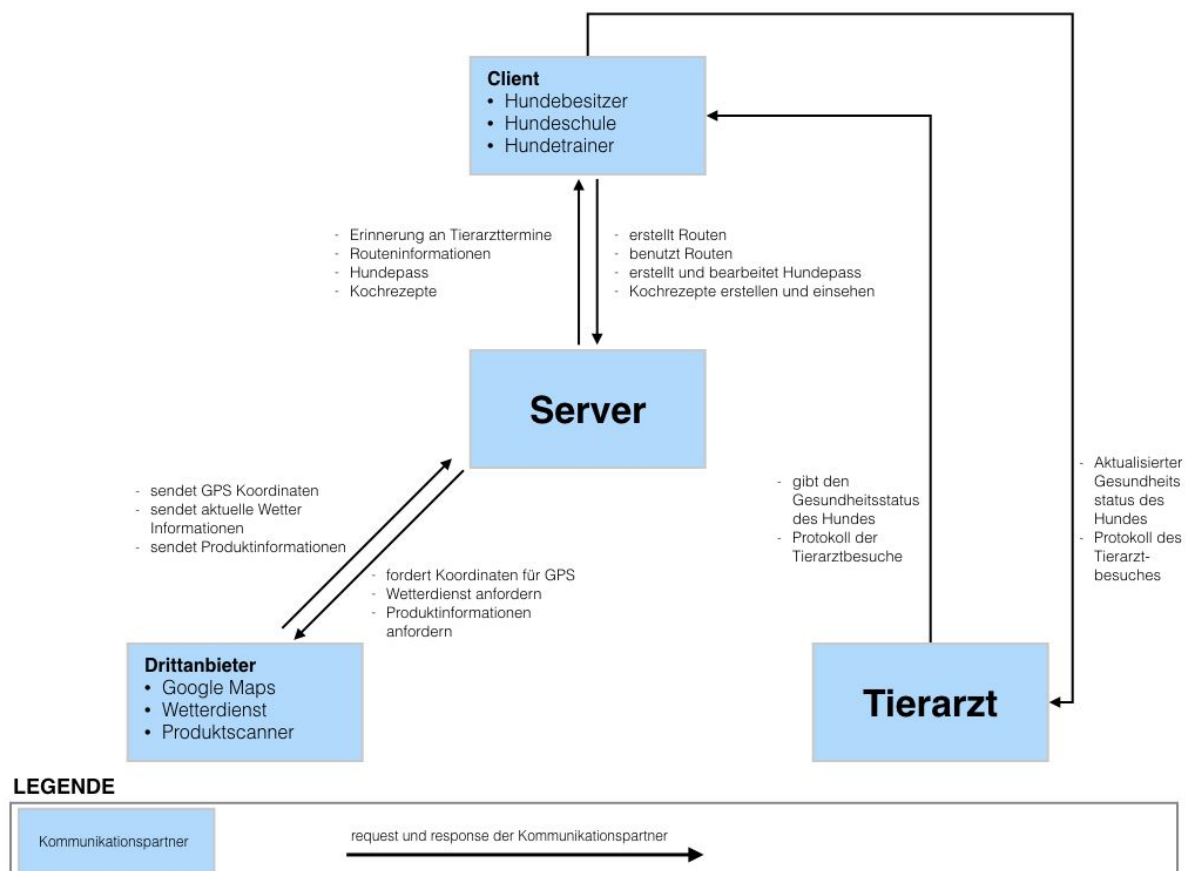
Als erstes wird hier das IST- Situation dargestellt und erklärt. Es wird die Kommunikation zwischen den Stakeholdern dargestellt. Die aktuelle Situation ist wichtig für die danachfolgende Analyse.



Der Hundebesitzer muss sich über das Wohlempfinden des Hundes informieren, wofür sie sich Informationen aus Büchern, Internetquellen, Prospekte, Zeitungen, Hundeschulen, von Fitnesstrainern oder auch von Tierärzten holen. Die Drittanbieter geben ihre Daten an den Kommunikationspartner weiter, die benötigt werden, um eine Planung vorzubereiten. Zum Beispiel, kann sich der Hundebesitzer im Vorfeld über das Wetter informieren und dementsprechend Routen planen, um Aktivitäten im Freien planen zu können.

Im IST-Zustand könnte man einige Kommunikationsvorgänge sparen. Diese werden im Folgenden in der SOLL-Situation skizziert.

Präskriptives Modell



Hundebesitzer, Hundeschulen und Hundetrainer müssen sich selbst Informationen über Routen beschaffen, um Aktivitäten oder Spaziergänge außerhalb des Hauses auszuführen. Die aktuelle Wetterlage ist dafür eine wichtige Informationsquelle. Zum Beispiel würden Unwetterwarnungen vorher informieren, dass man einen Ausgang vermeiden sollte.

Kochrezepte für eine gesunde Ernährung zu finden ist ein wichtiger Bestandteil. Hier wird angezeigt, ob sich Produkte für den Hund eignen oder ob Schadstoffe in diesen Produkten enthalten sind. All dies soll durch die Applikation erleichtert werden.

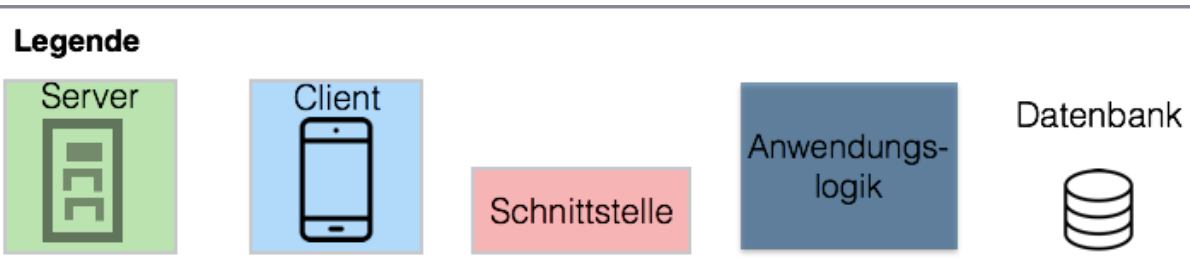
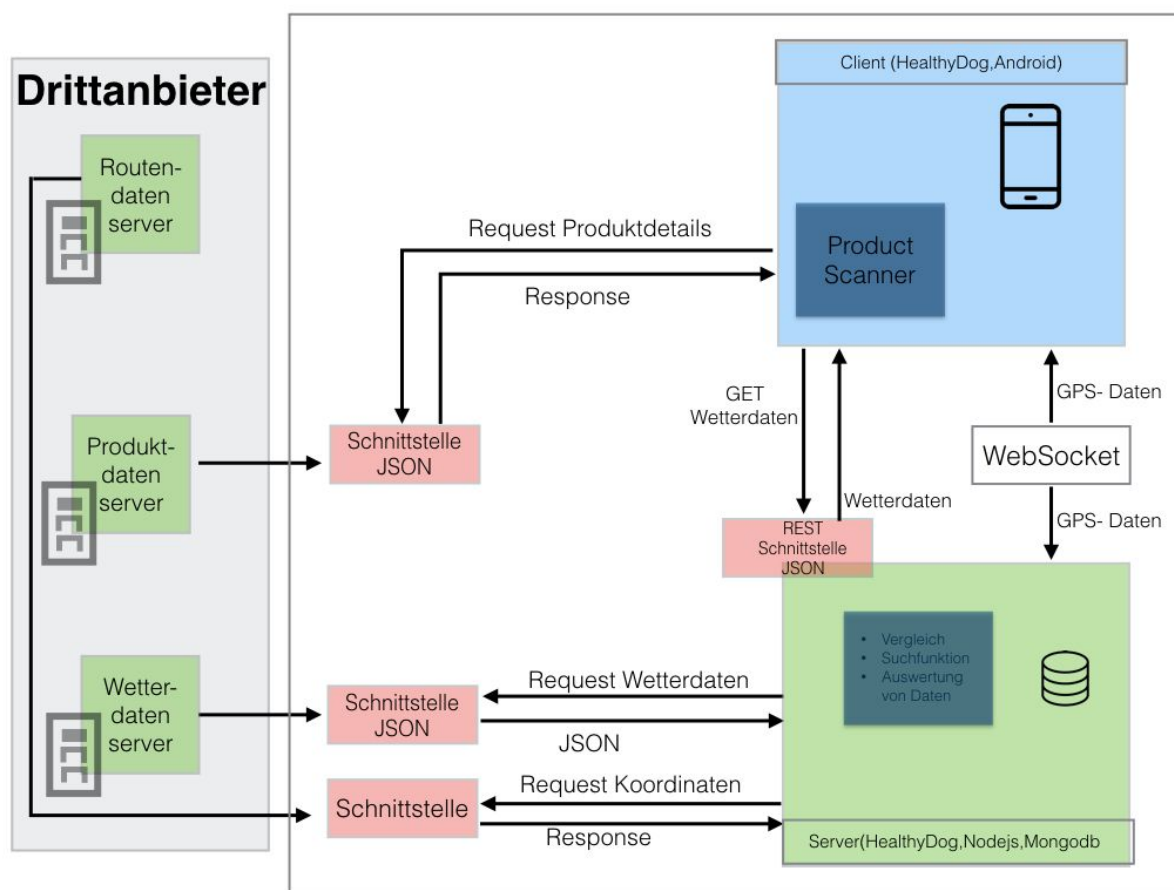
In der Skizze soll verdeutlicht werden, wie die Kommunikationspartner zueinander in Beziehung stehen und wie sie miteinander interagieren.

Durch einen Produktscanner soll die Identifizierung von Produkten für den Nutzer erleichtert werden. Durch die ihm nun bekannten Produkte und den vorgeschlagenen Rezepten, kann der Nutzer auf eine gesunde Ernährung seines Hundes achten.

Nutzer bekommen Informationen über das aktuelle Wetter, sodass sie sich vorbereiten können und evtl. andere Routen auswählen können.

Architekturdiagramm

Zur Entwicklung einer App werden die zur Systemarchitektur zugehörigen grundsätzlichen Aspekte der Struktur untergeordnet. Hierbei werden Systemkomponenten benötigt, um die Schnittstellen und die Kommunikation zwischen diesen zu erstellen. Die wohl wichtigste Komponente ist der Server, da dort wichtige Berechnungen durchgeführt werden. Diese Berechnungen und Funktionen liefern Daten bzw. Informationen, die dann zwischen Client und Server ausgetauscht werden.



Um Produkte identifizieren zu können, wird ein Barcodescanner auf dem Client implementiert.

Für die Funktion der Produktsuche wird entsprechend ein Produktserver implementiert. Produktdaten müssen jederzeit verfügbar und abrufbar sein. Deshalb werden diese Daten auf dem Client in einer Datenbank gespeichert.

Wetterdaten sollen stets aktuell angezeigt werden. Über ein Websocket wird eine dauerhafte Verbindung gewährleistet. GPS Daten sollen ebenfalls immer aktualisiert und gespeichert werden, damit es bei der Routenanzeige zu keinen Komplikationen kommen kann. Dazu wird ein externer Server speziell für den Routenplaner benötigt, welcher die Informationen nach Bedarf des Servers bzw. des Nutzers angibt.

Begründung

Nun werden die auserwählten Technologien ausgewertet. Die Systemarchitektur beinhaltet grundsätzliche Aspekte der Struktur die zur Entwicklung einer App gebraucht werden. Die Systemkompetenten werden daher benötigt um die Schnittstellen und die Kommunikation zwischen ihnen zu ermitteln. Falls der Nutzer in einem Funkloch landet, sollte seine Route trotzdem abgespeichert sein um diese weiter nutzen zukönnen. Das System sollte in Echtzeit gearbeitet werden. Die App soll durch den vielen reichhaltigen Tipps und Informationen, dass Wohlbefinden des Hundes verbessern.

JSON v XML

Es handelt sich um Programme, die den Datenaustausch vereinfachen sollen. Der Nachteil von XML jedoch ist das sehr hohe Datenvolumen im Gegensatz zu JSON.

In unserer App werden viele Daten verarbeitet.

JSON dient als ein sparsames Datenaustauschformat, welches zusätzlich die Ressourcen sparsam einsetzt. JSON ist ein reines JavaScript. Dies ist bei der Interaktion mit dem Wetterdienst von Vorteil, da in JavaScript Objekte gewandelt werden können. Das Programm hat eine einfach Handhabung, sowie eine einfache Implementierung. Daher fiel die Wahl bei uns auf das Programm JSON.

MongoDB v MySQL

Datenbankprogramme sollten in der Lage sein große Datenvolumen zu verarbeiten. MySQL besitzt eine sehr große Datenbank, die dann zum Nachteil wird, wenn bei Anfragen des Nutzers eine längere Antwortzeit auftritt. MySQL ist zwar in der Lage mehrere Datenbanken zeitgleich zu verarbeiten, jedoch ist das bei unserer App nicht notwendig.

MongoDB bietet kleinere Datenbanken an, die besonders gut für mobile Apps empfohlen werden. Zudem ist es sehr flexibel was die Datenverarbeitung angeht. Das Datenbankschema kann dynamisch angepasst werden. Hinzu kommen die Vorteile, dass sie

kostengünstig und eine horizontale Skalierbarkeit besitzen. Ein weiterer wichtiger Faktor, der für die Wahl von MongoDB spricht, ist die einfache Kombination mit JSON. Unsere Wahl fiel aus diesen Gründen auf MongoDB als Datenbank

NodeJS v PHP

PHP ist ein geeignetes Programm für den Einstieg. Der Anwendungsbereich begrenzt sich auf einfache Webseiten mit dynamischen Funktionen.

NodeJS bietet sich uns jedoch mehr an, da wir durch das Modul "Webbasierte Anwendungen 2: Verteilte Systeme" dem Umgang näher sind. NodeJS lässt sich ebenfalls gut mit MongoDB kombinieren, was als enormer Vorteil zu sehen ist. Die Vorteile an sich sind schnelle Ausführung, wenige Arbeitsspeichernutzung und das Verwenden von gemeinsamen Code auf Server und Client. Die Asynchronität verspricht zudem, dass Aufgaben parallel zueinander ausgeführt werden können. In unserem Fall können also mehrere Anfragen der Hundebesitzer zeitgleich verarbeitet werden, ohne dass es zu Antwortverzögerungen kommt. Daher entschieden wir uns für NodeJS.

REST- Architektur

REST hilft uns bei der Synchronisation unserer App. Bei einer möglichen Komplikation, wie z.B. einem Funkloch, muss der Nutzer dennoch Benachrichtigungen über die Route erhalten können. Diese Informationen werden in der Rest-Schnittstelle gespeichert. Die Zusammenarbeit von JSON mit einer REST-Architektur verläuft einwandfrei, weshalb wir uns für REST entschieden haben.

WebSockets

WebSockets sind in sofern wichtig, dass Informationen ständig aktualisiert werden. Wetterdaten oder Änderungen von Rezepten, müssen dem Nutzer angezeigt werden. Es handelt sich um eine bidirektionale Verbindung. Das heißt, eine beidseitige Kommunikation zwischen Server und Client ist möglich. So kann zum Beispiel der Hundebesitzer eine Route wählen und der Server kann zeitgleich Informationen dazu liefern. Dies ist über REST nicht möglich. WebSockets sind also unabdingbar, sodass der Server auf Änderungen reagiert und alle Nutzer informieren kann.

Produktserver

Der Produktserver enthält alle Informationen der Produkte, die bei der Ernährung genannt werden. Dazu gehören auch die Produkte, die bei bestimmten Rezepten empfohlen werden. Eine Bestätigung des Nutzers beweist die Richtigkeit der Informationen. Um zu verhindern, dass der Hauptserver von request-Befehlen überhäuft wird, verwenden wir einen Produktserver, der sich somit auf die Produkte fokussiert.