Hochschule Reutlingen
Reutlingen University

Temporäre Kooperation in mandantenfähigen Systemen

Knowledge Foundation @ Reutlingen University

Professional Software-Engineering (M.Sc.)

Masterarbeit

Pascal Nerger
Matrikelnummer: 807606

31. August 2024

Bearbeitungszeitraum:	01.03.2024 - 31.08.2024
Teilnehmernummer:	807606
Ausbildungsfirma:	IT-Designers GmbH, Esslingen am Neckar
Betreuer:	Prof. Dr. M. Schmollinger, E. Stricker (M.Sc.)

Abstract

Disaster management in Germany is conducted in accordance with established processes and procedures. A multitude of organizations collaborate to facilitate the expedient treatment of patients. This thesis focuses on the *KatApp*, a system designed to enhance logistical operations in the context of disaster management. However, the *KatApp's* multi-tenant architecture necessitates the segregation of data between individual tenants, thereby limiting inter-organizational coordination in disaster scenarios.

In this thesis, a concept is developed that allows organizations to cooperate with each other for the duration of a disaster. To this end, the technical aspects of disaster management are first analyzed, from which the requirements for the new system are derived. In addition to these requirements, related work in the area of multi-tenant systems will be consulted. The requirements are implemented in a Proof of Concept, which is evaluated using test scenarios.

The proposed solution allows for temporary cooperation between tenants through a geo-based authorization that is valid for the duration of a disaster. The evaluation of various test scenarios demonstrates that the developed Proof of Concept is functional and fulfills the specified requirements. In the future, the approach presented in this paper can be extended to include real-time location tracking of participating users to make the system more reliable.

Kurzfassung

Der Katastrophenschutz in Deutschland folgt festgelegten Prozessen und Abläufen. Dabei sind stets mehrere Organisationen beteiligt, welche für eine möglichst schnelle Versorgung aller Patienten kooperieren. Die vorliegende Arbeit befasst sich mit der *KatApp*, einem System zur Optimierung logistischer Prozesse im Kontext des Katastrophenschutzes. Die Mandantenfähigkeit der *KatApp* bedingt jedoch eine Datentrennung zwischen den einzelnen Mandanten, sodass eine Zusammenarbeit der Organisationen untereinander in einer Katastrophe derzeit nicht möglich ist.

Im Rahmen dieser Arbeit wird ein Konzept entwickelt, welches eine Kooperation der Organisationen untereinander für die Dauer einer Katastrophe ermöglicht. Zu diesem Zweck werden zunächst die fachlichen Aspekte des Katastrophenschutzes analysiert, aus welchen die Anforderungen an das neue System abgeleitet werden. Ergänzend zu diesen Anforderungen werden verwandte Arbeiten im Bereich Mandantenfähigkeit herangezogen. Die Anforderungen werden in einem Proof of Concept realisiert, welcher einer Evaluation mittels Testszenarien unterzogen wird.

Eine temporäre Kooperation unter Mandanten wird durch eine geobasierte Autorisierung ermöglicht, welche für die Dauer einer Katastrophe gültig ist. Aus der Evaluation diverser Testszenarien ergibt sich, dass der entwickelte Proof of Concept funktional ist und die Anforderungen erfüllt. In Zukunft kann der in dieser Arbeit vorgestellte Ansatz um die Nachverfolgung von Echtzeitstandorten involvierter Nutzer erweitert werden, um das System zuverlässiger zu gestalten.

Inhaltsverzeichnis

1 Einführung **1**
1.1 Hintergrund . . . 1
1.2 Problemstellung . . . 2
1.3 Ziel der Arbeit . . . 3
1.4 Forschungsfragen . . . 3
1.5 Struktur der Arbeit . . . 3

2 KatApp **5**
2.1 Rollen . . . 5
2.2 Ablauf . . . 6
2.3 Informationsfluss . . . 7
2.4 Probleme und Lösungen . . . 8
2.5 Komponenten . . . 9
2.5.1 App . . . 9
2.5.2 Dashboard . . . 10
2.5.3 Mandantenkonsole . . . 11
2.5.4 Adminkonsole . . . 11
2.5.5 Backend . . . 11
2.6 Funktionsweise . . . 11
2.7 Mandantschaften . . . 13
2.8 Authentifizierung . . . 14
2.9 Autorisierung . . . 16

3 Mandantenfähigkeit **19**
3.1 Architekturen . . . 19
3.1.1 Dediziertes Modell . . . 20
3.1.2 Isoliertes Modell . . . 21
3.1.3 Geteiltes Modell . . . 21
3.1.4 Hybrides Modell . . . 22
3.2 KatApp und Mandantenfähigkeit . . . 23
3.3 Herausforderungen . . . 23

4 Verwandte Arbeiten **27**
4.1 Überblick . . . 27
4.2 Cloud & Katastrophenmanagement . . . 27
4.3 Mandantenfähigkeit & Zugangskontrolle . . . 28
4.4 Forschungslücken . . . 28

5 Architektur **29**
5.1 Qualitätsziele . . . 29
5.2 Umgebungen . . . 30
5.3 Lokale Umgebung . . . 31
5.4 Interaktionen & Schnittstellen . . . 31
5.5 Cloud . . . 33
5.6 Cloud Architektur . . . 34
5.7 Lokale Architektur . . . 36
5.8 Technologien . . . 38

6 Implementierung **39**
6.1 Datenmodell . . . 39
6.2 Authentifizierung & Autorisierung . . . 42
6.3 Erstellung einer Triage . . . 43
6.4 Verbindung via WebSocket . . . 44

7 Anforderungsanalyse **47**
7.1 Kooperation . . . 48
7.2 Einsatzleitung . . . 49
7.3 Leitstelle . . . 50
7.4 Katastrophenende . . . 51
7.5 Datenzugriff . . . 52

8 Konzeption **53**
8.1 Problematik . . . 53
8.2 Administrative Ebenen . . . 53
8.3 Internationaler Einsatz . . . 54
8.4 Leitstellen und Landkreise . . . 55
8.5 Einsatzleitung und standortbasierter Zugriff . . . 55
8.6 Problematiken eines standortbasierten Zugriffs . . . 56
8.7 Geokodierung und Standortermittlung . . . 56
8.8 OpenStreetMap . . . 57
8.9 IP-API . . . 58
8.10 Rollen & Mandanten . . . 58
8.11 Katastrophenmanagement . . . 59
8.12 Datenzugriff & Export . . . 59

9 Proof of Concept **61**
9.1 Datenmodell . . . 61
9.2 Erstellung einer Triage . . . 61
9.3 Standorterfassung via Trust-Key . . . 63
9.4 Verbindungsaufbau via WebSocket . . . 63
9.5 Abfrage von Katastrophendaten . . . 64
9.6 Ende einer Katastrophe . . . 65

10 Evaluierung **67**
10.1 Kriterien & Metriken . . . 67
10.2 Allgemeines . . . 67
10.3 Testszenarien . . . 67
10.4 Testmittel & Ablauf . . . 69
10.5 Fachliche Korrektheit . . . 69
10.6 Zuverlässigkeit . . . 69
10.7 Benutzerfreundlichkeit . . . 70
10.8 Ausführungszeit . . . 71
10.9 Ergebnisanalyse . . . 71

11 Schluss **73**
11.1 Zusammenfassung . . . 73
11.2 Kritische Reflexion . . . 73
11.3 Fazit . . . 74
11.4 Ausblick . . . 74
11.5 Schlusswort . . . 75

1 Einführung

Traditionellerweise werden Triagen bei einer Katastrophe auf Papier dokumentiert. Jeder Patient erhält bei seiner Sichtung eine Triagekarte, worauf seine persönlichen Befunde festgehalten werden. Bei weiteren Triagen im Verlauf der Katastrophe (aufgrund eines Standortwechsels oder einer Verschlechterung bzw. Verbesserung des Zustands eines Patienten) werden zusätzliche Befunde auf derselben Karte ergänzt, so ergibt sich ein individueller Verlauf für jeden Patienten. Bei jeder Triage wird dem Patienten, basierend auf den Befunden sowie des verwendeten Triage-Algorithmus, eine von fünf Kategorien zugeteilt (Grün, Gelb, Rot, Schwarz, Blau). Grün signalisiert, dass der Patient nur leichte Verletzungen erlitten hat und keinesfalls in Lebensgefahr schwebt. Patienten, welche als Grün klassifiziert werden, können zumeist auch selbstständig gehen. Gelb signalisiert schwere Verletzungen, welche umgehend behandelt werden sollten. Patienten, die als Gelb klassifiziert werden, können meist nicht mehr selbstständig gehen. Rot signalisiert einen lebensbedrohlichen Zustand. Patienten, die als Rot klassifiziert werden, müssen umgehend behandelt werden, andernfalls verlieren diese höchstwahrscheinlich ihr Leben. Patienten, welche als Schwarz klassifiziert werden, sind bereits tot. In Ausnahmefällen wird auch die Kategorie Blau vergeben. Dies geschieht nur, falls die von der Katastrophe verursachten Verletzungen so schwer sind, dass der Patient keine Überlebenschance hat. [1] [2]

Zur Klassifizierung eines Patienten wird ein vorher ausgewählter Triage-Algorithmus verwendet. Dieser sieht einige vordefinierte Schritte vor, welche vom Ersthelfer oder behandelnden Arzt durchgeführt werden müssen, um die Klassifizierung (Grün, Gelb, Rot, Schwarz, Blau) des Patienten zu ermitteln. Unterschiedliche Algorithmen können verschiedene Resultate erzielen. Ziel der Algorithmen ist jedoch stets, so schnell wie möglich eine Klassifizierung zu erhalten. [3] [4]

Die Triagierung und Behandlung von Patienten während einer Katastrophe ist nicht nur ein medizinischer, sondern vor allem auch ein logistischer Prozess. Nachdem Patienten von den Ersthelfern gesichtet wurden, werden jene, basierend auf ihrer Erstklassifikation, vom Schadensort entweder zur Patientenablage oder zum Behandlungsplatz gebracht. Je nach Ergebnis der weiteren Triagen, die am Patienten durchgeführt werden, wird der Patient notdürftig vor Ort behandelt oder schnellstmöglich in ein naheliegendes Krankenhaus überführt. Dieser Prozess erfordert eine präzise Planung der zur Verfügung stehenden Rettungsmittel (Rettungswagen, Helikopter, etc.) und deren Transportkapazitäten. Ziel ist es, am Ende einer Katastrophe alle Patienten in einem Krankenhaus untergebracht zu haben.

Dieser Prozess wird von der Einsatzleitung überwacht. Die Einsatzleitung erfasst die in der Katastrophe befindlichen Patienten und deren Lagestatus, Informationen, die an die Leitstelle übertragen werden, welche für die Planung von Rettungsmitteln und Aufnahmekapazitäten verantwortlich ist. Bei größeren Katastrophen ist dieser Vorgang alles andere als trivial. Es müssen nicht nur die verfügbaren Rettungsmittel und vorhandenen Kapazitäten der umliegenden Krankenhäuser ermittelt, sondern zudem auch ein Plan entwickelt werden, welche Patienten mit Priorität abtransportiert werden sollen. Dieser Prozess läuft zumeist telefonisch oder per Funk ab und ist daher oft zeitaufwendig und ineffizient. [5]

1.1 Hintergrund

Die *KatApp* (Katastrophen App) ist ein Projekt zur logistischen Optimierung von Abläufen während einer Katastrophe. Diese Abläufe umfassen, wie oben beschrieben, die Triagierung von Patienten, die Zuordnung von Patienten zu verfügbaren Rettungsmitteln sowie die Überwachung der Katastrophe durch eine Einsatzleitung und Leitstelle. Die *KatApp* ist ein mehrteiliges Projekt und setzt sich primär aus einer App, einem Dashboard und einem Backend zusammen. Genaueres zur *KatApp* wird in Abschnitt 2 beschrieben.

Das Ziel der *KatApp* ist die Optimierung des analogen Triage-Prozesses hin zu einem digitalen Prozess. Vorteile, welche sich daraus ergeben sollen, sind vor allem die Beschleunigung und die Überwachbarkeit des Prozesses. Dabei soll die Triagekarte durch

einen QR-Code ersetzt werden, welcher einen Patienten eindeutig identifiziert. Anstelle der Durchführung des Triage-Algorithmus mit Stift und Papier, soll dieser nun auf der mobilen App durchgeführt werden. Die durchgeführte Triage wird nach Abschluss mit Dashboard und Backend der *KatApp* synchronisiert, sodass sowohl Einsatzleitung als auch Leitstelle jederzeit einen Überblick über die durchgeführten Triagen, deren Standorte und die zugeteilten Klassifikationen haben. Mit Hilfe dieser Informationen können Rettungsmittel sorgfältig geplant sowie Kommunikationswege effizienter gestaltet werden.

1.2 Problemstellung

Bei größeren Katastrophen sowie bei Katastrophen in Grenzgebieten (zwischen Landkreisen, Bundesländern, etc.) ist es üblich, dass mehr als eine Organisation an einer Katastrophe beteiligt ist. Typische beteiligte Organisationen sind naheliegende Kliniken, die Feuerwehr und Rettungsdienste, zu denen zum Beispiel das *Deutsche Rote Kreuz* (DRK), und das *Technische Hilfswerk* (THW), aber auch zum Beispiel der *Arbeiter-Samariter-Bund* (ASB) und die *Johanniter-Unfall-Hilfe* (JUH) sowie der *Malteser Hilfsdienst* (MHD) zählen. Ziel ist es, dass sich alle Organisationen reibungslos am Prozess beteiligen können, sodass alle Patienten möglichst schnell versorgt werden. [6]

Die *KatApp* ist ein Mandantensystem, in dem sämtliche Ressourcen (Katastrophen, Patienten, Triagen) einem Mandanten zugeordnet werden. Die Zuordnung erfolgt dabei automatisch über den Nutzer, welcher die Ressource erstellt. Es ist stets bekannt, zu welchem Mandanten ein Nutzer gehört, daher ist auch stets bekannt, zu welchem Mandanten eine Ressource gehört. Auf fachlicher Ebene wird ein Mandant durch eine individuelle Organisation (zum Beispiel einer Ortsgruppe des DRK) repräsentiert.

Organisationen benötigen eine Mandantschaft, um Zugang zum System zu bekommen. Nutzer wiederum bekommen nur über ihre jeweilige Organisation Zugang. Wie in beinahe jedem Mandantensystem ist es auch in der *KatApp* üblich, dass Ressourcen nur von Nutzern des Mandanten eingesehen werden können, welchem die Ressourcen gehören. Dies mag eine robuste Lösung aus Sicht der Datensicherheit bzw. des Datenschutzes sein, bildet jedoch nicht exakt die Fachlichkeit der Realität ab.

In einer Katastrophe, an der mehrere Organisationen teilnehmen, kommt es vor, dass Triagen von Mitarbeitern unterschiedlichster Organisationen durchgeführt werden. Patienten werden möglicherweise mehrfach von Ersthelfern triagiert. Diese Ersthelfer müssen jedoch nicht derselben Organisation angehören. Mitarbeiter von Einsatzleitung und Leitstelle entspringen ebenfalls unterschiedlichen Organisationen, benötigen jedoch genauso Einsicht in Patientendaten.

Dadurch, dass Mandanten ausschließlich auf ihre eigenen Ressourcen, und nicht die der anderen Mandanten zugreifen können, ist es unterschiedlichen Organisationen nicht möglich zusammenzuarbeiten, da Ressourcen nicht geteilt werden können. Sollten also mehrere Organisationen mit der *KatApp* in einer Katastrophe zusammenarbeiten, würde dies zu chaotischen Umständen führen.

Eine Katastrophe wird stets mit der ersten Triage eröffnet (Initial-Triage). Angenommen, dies geschieht durch Nutzer A, welcher zu Organisation A gehört. Triagiert Nutzer A nun Patient A, werden alle zugehörigen Ressourcen (Katastrophe, Patient, Triage) im System Organisation A zugeordnet. Sollte nun Nutzer B, welcher Organisation B angehört, versuchen eine Folgetriage am gleichen Patienten A durchzuführen, resultiert dies nicht in einer Folgetriage, sondern in einer Ersttriage. Alle Ressourcen (Katastrophe, Patient, Triage) werden neu erstellt und Organisation B zugeordnet, da die Daten von Organisation A nicht für Organisation B sichtbar sind. Sämtliche Ressourcen werden also doppelt im System geführt. Selbes gilt für Einsatzleitung und Leitstelle. Wird zum Beispiel die Einsatzleitung von Organisation A gestellt, werden die Hilfskräfte der Organisation A nicht über Triagen und Patienten benachrichtigt, die von Organisation B ausgehen. Diese Problematik skaliert sich, sollten mehr als zwei Organisationen an einer Katastrophe beteiligt sein.

1.3 Ziel der Arbeit

Ziel dieser Arbeit ist die Angleichung der *KatApp* an die fachlichen Prozesse der Katastrophenhilfe. Es soll ermöglicht werden, dass mehrere Organisationen problemlos in einer Katastrophe zusammenarbeiten können. Um dies zu realisieren, werden die fachlichen Anforderungen für eine Kooperation mehrerer Organisationen mit Hilfe von Experten aus dem medizinischen und medizintechnischen Bereich erhoben. Aus diesen Anforderungen soll ein Konzept erarbeitet werden, mit welchem eine Kooperation unter Organisationen ermöglicht werden kann. Die Funktionsfähigkeit dieses Konzeptes soll mittels eines *Proof of Concept* (PoC) nachgewiesen werden. Hierfür soll der entwickelte PoC mittels selektierter Testverfahren evaluiert werden.

Dabei beschränkt sich der Umfang dieser Arbeit auf das Backend der *KatApp*. Zugehörige Anwendungen des *KatApp* Ökosystems wie App oder Dashboard fallen nicht in den Umfang dieser Arbeit. Die im Rahmen dieser Arbeit durchgeführten Änderungen des *KatApp* Backends führen zu Inkompatibilitäten zwischen Backend, App und Dashboard. Auch diese werden nicht in dieser Arbeit behoben.

1.4 Forschungsfragen

Im Rahmen dieser Arbeit sollen die folgenden Forschungsfragen beantwortet werden:

RQ1 Wie kann der bisherige mandantenfähige Ansatz der *KatApp*-Architektur erweitert werden, um die Zusammenarbeit mehrerer Mandanten in einer Katastrophe zu ermöglichen?

- **RQ1.1** Bestehen bereits Architekturen, die Kooperationsfähigkeit unter Mandanten implementieren, und falls ja, wie können diese bestehenden Ansätze in der *KatApp* angewendet werden?
- **RQ1.2** Ist eine fachliche Trennung der sensiblen medizinischen Daten ausreichend oder ist eine Trennung auf technischer Ebene, zum Beispiel durch eine getrennte Infrastruktur, erforderlich?

RQ2 Wie muss die Verwaltung von Katastrophen in der *KatApp* angepasst werden, damit mehrere Organisationen im Falle einer Katastrophe kooperieren können?

- **RQ2.1** Ab wann gilt eine Katastrophe als gestartet, ab wann gilt sie als beendet?
- **RQ2.2** Falls mehrere Organisationen vor Ort sind, welche dieser Organisationen übernimmt die Rolle der Einsatzleitung oder die der Leitstelle?
- **RQ2.3** Was passiert mit sensiblen Patientendaten, wenn eine Katastrophe beendet ist?

RQ3 Wie kann im Falle einer Kooperation mehrerer Organisationen sichergestellt werden, dass die entstandenen Nutzungsdaten dem jeweiligen Mandanten zugeordnet werden, sodass der Betreiber des Systems auf die verursachten Kosten rückschließen kann?

1.5 Struktur der Arbeit

Zu Beginn der Arbeit wird die *KatApp* genauer vorgestellt. Hierbei wird speziell die Fachlichkeit in den Vordergrund gestellt. Es wird auf involvierte Rollen, den Ablauf einer Katastrophe sowie den Informationsfluss im Katastrophenmanagement eingegangen. Darüber hinaus werden Probleme und Herausforderungen des aktuellen Ablaufs aufgezeigt und es wird erläutert, wie die *KatApp* diese zu beheben versucht. Des Weiteren werden die

individuellen Komponenten der *KatApp* dargelegt. Das Kapitel wird mit einem Überblick über die Funktionsweise der *KatApp* abgeschlossen.

Nachdem die *KatApp* vorgestellt wurde, werden generelle Architekturen von mandantenfähigen Systemen inspiziert. Es wird untersucht, inwiefern sich diese von der *KatApp* unterscheiden und eruiert, weshalb gewisse Designentscheidungen getroffen wurden. Anschließend werden verwandte Arbeiten analysiert. Da es sich bei der *KatApp* um ein recht spezifisches Projekt handelt, werden hierbei Arbeiten selektiert, welche Konzepte unterbreiten, die auf die *KatApp* anwendbar sind.

Bevor auf die Implementierung der zentralen Prozesse der *KatApp* eingegangen wird, wird die Architektur des Backends vorgestellt. Dabei werden zunächst die Qualitätsziele der Architektur betrachtet, bevor auf die verschiedenen Ausführungsumgebungen der *KatApp* eingegangen wird. Es wird diskutiert, warum sowohl der Betrieb in der Cloud als auch der lokale Betrieb notwendig sind und wie diese architektonische Herausforderung gelöst wird.

Im Rahmen der Implementierung erfolgt zunächst eine Beschreibung des Datenmodells der *KatApp*. Dabei wird auch auf die Gründe für die Wahl des Datenmodells eingegangen. Im Anschluss werden zentrale Prozesse und Abläufe der *KatApp* erörtert, wodurch die Grundlage für die konzeptionelle Arbeit gelegt wird. Diese baut ebenfalls auf der vorangehenden Anforderungsanalyse auf.

Im Anschluss an die Anforderungsanalyse, welche unter Einbezug ausgewählter Experten durchgeführt wird, erfolgt eine Erörterung der Konzeption. Dabei werden Lösungsansätze für die aus den erhobenen Anforderungen resultierenden Herausforderungen präsentiert. Die Konzeption bildet die Grundlage für die Implementierung des PoC, welcher im folgenden Abschnitt erläutert wird.

Im Rahmen der Evaluierung des implementierten PoC erfolgt eine Bewertung des Lösungsmodells anhand diverser Kriterien und Metriken. Die Arbeit schließt mit einem Fazit und einem Ausblick.

2 KatApp

Die *KatApp* ist ein Innovations- und Forschungsprojekt, welches bereits seit mehreren Jahren aktiv von Studenten der Hochschule Reutlingen[1] in Kooperation mit dem Universitätsklinikum Tübingen[2] entwickelt wird.

Im Rahmen der letzten am Projekt getätigten Arbeiten wurde das Backend der *KatApp* von Grund auf neu entwickelt. Es wurde eine neue, cloudbasierte und mandantenfähige Architektur eingeführt. In Abschnitt 1 werden bereits die Gründe für die Einführung dieser Architektur angeschnitten. Zudem wird auf die Problematik eingegangen, die aus der Einführung der mandantenfähigen Architektur resultiert. Es existiert eine Diskrepanz zwischen der aktuellen Umsetzung und den fachlichen Abläufen des Katastrophenschutzes, welche in dieser Arbeit behoben werden soll.

Diese Arbeit baut auf der bestehenden Implementierung der *KatApp* auf. Im Folgenden wird diese Version der *KatApp* auch als erste Version bzw. *KatApp 1.0* referenziert. Die folgenden Kapitel befassen sich unter anderem mit den fachlichen Abläufen des Katastrophenschutzes. Da die *KatApp* diese Abläufe digital nachbildet und optimiert, ist es wichtig diese Abläufe im Voraus zu klären. Durch die Digitalisierung des Katastrophenschutzes können zudem andere Problematiken des analogen Prozesses eliminiert werden. Die Funktionsweise der *KatApp* sowie deren Komponenten werden im Anschluss an die Fachlichkeit diskutiert. Darüber hinaus werden die Grundlagen der Mandantenfähigkeit erläutert.

2.1 Rollen

Im Falle einer Katastrophe sind stets diverse Einsatzkräfte vor Ort. Für einen organisierten und effizienten Ablauf übernimmt jede Person oder Gruppe von Personen eine bestimmte Rolle. Im Folgenden werden die an einer Katastrophe beteiligten Entitäten sowie deren Rollen beschrieben:

Patient Ein Patient ist eine Person, die aufgrund einer Katastrophe zu Schaden gekommen ist. Er ist verletzt und braucht zumeist medizinische Versorgung. Zudem ist er häufig nicht in der Lage, das Katastrophengebiet selbständig zu verlassen. Dem Patienten wird mittels Triage eine der folgenden Kategorien zugeordnet: Grün, Gelb, Rot, Schwarz oder Blau (in Ausnahmefällen).

Ersthelfer Die Ersthelfer treffen meist zuerst im Katastrophengebiet ein. Ersthelfer sind medizinisch ausgebildete Fachkräfte. Sie suchen das Katastrophengebiet nach Patienten ab. Sie führen die Triagen an den Patienten durch und nehmen die Klassifikationen in eine der obigen Kategorien (Grün, Gelb, Rot, Schwarz, Blau) vor. Zudem leisten sie, falls nötig, erste Hilfe, sollte ein akutes, lebensbedrohliches Problem bestehen.

Notarzt Notärzte sind bei einer Katastrophe vor Ort, um akute Behandlungen an kritisch verletzten Patienten durchzuführen. Oftmals müssen die Patienten vorsorglich behandelt werden, da es, abhängig von den Katastrophenbedingungen, einige Zeit dauern kann, bis Patienten in eine medizinische Einrichtung abtransportiert werden können.

Einsatzleitung Die medizinische Einsatzleitung setzt sich dem organisatorischen Einsatzleiter (auch *OrgL* genannt) sowie dem leitenden Notarzt (auch *LNA)* zusammen. Sie übernimmt die Führung der Katastrophe vor Ort. Während der OrgL die taktische Führung und Koordination materieller und personeller Ressourcen übernimmt, leitet der LNA alle medizinischen Maßnahmen vor Ort.

[1] Hochschule Reutlingen: `https://www.reutlingen-university.de/`
[2] Universitätsklinikum Tübingen: `https://www.medizin.uni-tuebingen.de/de/`

Leitstelle Die Einsatzleitstelle (kurz *Leitstelle*) leitet und überwacht den Einsatz aller beteiligten Organisationen. Ihre primäre Aufgabe ist die Entgegennahme von Informationen und die Koordinierung der teilnehmenden Organisationen. Eine ihrer wichtigsten Rollen ist die Organisation von Rettungsmitteln (Rettungswagen, Hubschrauber, etc.) und der Abtransport von Patienten. In Deutschland werden Leitstellen je Landkreis vergeben. Eine Leitstelle ist für alle Katastrophen innerhalb ihres Landkreises verantwortlich.

Organisation Im Rahmen einer Katastrophe sind stets diverse Organisationen beteiligt. Eine teilnehmende Organisation könnte eine Klinik sein, welche Patienten entgegennimmt, oder eine behelfende Rolle übernehmen wie zum Beispiel das DRK oder das THW.

2.2 Ablauf

In diesem Unterkapitel wird der Ablauf einer Katastrophe beschrieben. Dies dient dazu, den Einsatzzweck der *KatApp* besser nachvollziehen zu können. Es ist anzumerken, dass sich der Ablauf der Katastrophenhilfe je nach Bedingungen und Katastrophe im Einzelfall unterscheiden kann. Grundsätzlich bleiben jedoch die Grundprinzipien der Katastrophenhilfe intakt. Diese Prinzipien werden in Abbildung 2.1 veranschaulicht.

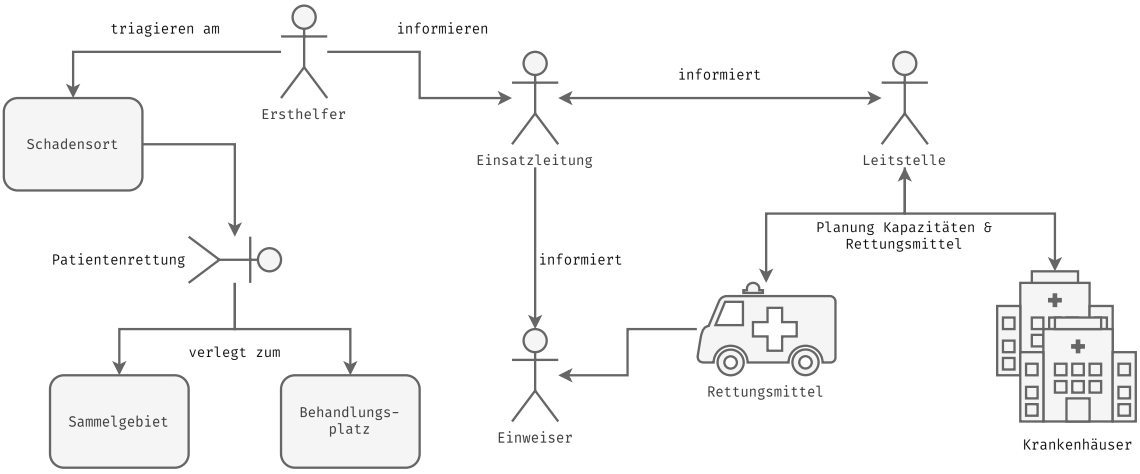

Abbildung 2.1: Ablauf einer Katastrophe [5]

Im Katastrophengebiet angekommen, suchen die Ersthelfer am Schadensort nach den ersten Patienten und starten mit der Durchführung der ersten Triagen. Ungefähr zur gleichen Zeit sorgen weitere Einsatzkräfte für den Aufbau des Sammelgebiets bzw. Behandlungsplatzes. Zumeist wird dafür ein größeres, aufblasbares Zelt aufgebaut, welches Platz für einige Patienten bietet und mit Liegen sowie medizinischen Geräten ausgestattet ist. [5]

Die Ersthelfer sind stets dazu angehalten, die Einsatzleitung vor Ort über die aufgefundenen Patienten und deren Lagestatus zu informieren. Diese Informationen sind wichtig für die Einsatzleitung, da diese das Ausmaß der Katastrophe ungefähr abschätzen muss. Die Einsatzleitung ist zudem in ständigem Kontakt mit der Leitstelle, welche basierend auf den Informationen der Einsatzleitung die Kapazitäten und Rettungsmittel der umliegenden Krankenhäuser erfragt. Darüber hinaus entsendet die Leitstelle die verfügbaren Rettungsmittel. Die von der Leitstelle geplanten Rettungsmittel werden dann im Katastrophengebiet vom Einweiser empfangen. Dieser bildet den Kern der Transportorganisation vor Ort. Die Patienten werden vom Behandlungsplatz, basierend auf der Dringlichkeit und der Schwere ihrer Verletzungen, auf die verfügbaren Rettungsmittel verteilt. Eine Katastrophe gilt als beendet, sobald alle Patienten abtransportiert und in einer der umliegenden medizinischen Einrichtungen untergebracht wurden. [5]

Patienten werden im Laufe einer Katastrophe mehrfach triagiert. Die Ersttriage findet dabei stets am Schadensort der Katastrophe statt. Im Sammelgebiet, am Behandlungsplatz oder sogar im Rettungswagen können erneute Triagen des Patienten stattfinden. Dieser Prozess dient einer erneuten Ermittlung des Patientenzustands. Der Zustand eines Patienten kann sich im Verlauf der Katastrophe verbessern oder verschlechtern. Je nachdem, welcher Fall eintritt, besteht entweder mehr oder weniger Handlungsbedarf. Durch die mehrfache Triagierung von Patienten ergibt sich eine Art individuelle Patientenhistorie. Aus dieser können weitere Erkenntnisse gewonnen werden. Verschlechtert sich beispielsweise der Zustand des Patienten im Verlauf der Katastrophe immer weiter, kann diesem Patienten eine höhere Priorität beim Abtransport zugewiesen werden. [5]

2.3 Informationsfluss

Als digitale Lösung soll die *KatApp* den Informationsfluss zwischen den einzelnen, beteiligten Instanzen erleichtern und beschleunigen. Dafür müssen die wichtigsten Kommunikationswege im Prozess der Katastrophenhilfe herausgearbeitet und analysiert werden. Um die Kommunikationswege zu bestimmen, müssen zunächst die beteiligten Akteure inspiziert werden:

Ersthelfer Die Ersthelfer kommunizieren vor allem Informationen über Patienten. Dazu gehören zum Beispiel Lagestatus, Klassifizierung und Anzahl. Diese Informationen sind wichtig, um das Ausmaß einer Katastrophe festzustellen. Abnehmer dieser Informationen ist die Einsatzleitung.

Einsatzleitung Die Einsatzleitung bildet einen Dreh- und Angelpunkt für Informationsaustausch. Sie nimmt vorwiegend Informationen von Ersthelfern entgegen und verteilt diese an Leitstelle und Einweiser. Dabei werden die Informationen für den jeweiligen Abnehmer gefiltert.

Leitstelle Die Leitstelle nimmt Informationen von der Einsatzleitung entgegen. Dabei interessiert sich die Leitstelle vor allem für die Anzahl und die Klassifizierungen der Patienten. Zudem kontaktiert und erfasst sie die Kapazitäten und verfügbaren Rettungsmittel umliegender Kliniken. Mit diesen Informationen wird die Verteilung der Patienten auf die entsprechenden Kliniken geplant. So muss zum Beispiel beurteilt werden, ob in einer Klinik noch Betten für Aufnahme von Patienten zur Verfügung stehen. Unter anderem kann auch festgestellt werden, ob eine Klinik noch über bestimmte Geräte verfügt, die zum Beispiel für die Behandlung von Schwerverletzten benötigt werden.

Einweiser Der Einweiser empfängt sowohl Informationen von der Einsatzleitung als auch von der Leitstelle. Er erhält Informationen über schwer verletzte Patienten sowie die zur Verfügung stehenden Rettungsmittel. Mit diesen Informationen und in Kooperation mit der Einsatzleitung koordiniert er den Abtransport der Patienten.

Krankenhäuser Die Krankenhäuser benötigen vor allem Information über die Anzahl der Patienten und deren Klassifikationen. Basierend auf diesen Informationen können sie die verfügbaren Kapazitäten an Betten und Rettungsmitteln an die Leitstelle durchgeben.

Zwischen den genannten Akteuren fließen Informationen. Im Optimalfall erhält jeder Akteur genau die Infoformationen, die er benötigt. Zudem sollte der Informationsaustausch in Echtzeit stattfinden. Genau an dieser Stelle setzt die *KatApp* an. In Zukunft sollen also die folgenden Kommunikationswege durch die *KatApp* gesteuert werden: Ersthelfer zu Einsatzleitung, Einsatzleitung zu Einweiser, Einsatzleitung zu Leitstelle, Leitstelle zu Einweiser. Aktuell sieht die *KatApp* nicht vor, in die Kommunikation zwischen Leitstelle und

Kliniken einzugreifen, wobei dies eine potenzielle Erweiterung in der Zukunft darstellen könnte.

Eine Analyse des in Abbildung 2.1 dargestellten Informationsflusses offenbart, dass es durchaus redundanten Informationsaustausch zwischen den involvierten Parteien gibt. Vor allem die Einsatzleitung fungiert als Dreh- und Angelpunkt des Informationsaustausches. Informationen, die von Ersthelfern oder der Leitstelle stammen und an den Einweiser gerichtet sind, nehmen stets einen Umweg über die Einsatzleitung. An dieser Stelle kann man nicht nur den Informationsaustausch kompakter gestalten, man kann ebenso redundanten Informationsaustausch kürzen. Unterabschnitt 2.6 geht genauer auf die Funktionsweise der *KatApp* ein. Hier wird dargestellt, inwiefern diese Kommunikationswege optimiert werden.

2.4 Probleme und Lösungen

In diesem Unterabschnitt werden die Problematiken des aktuellen Ablaufs der Katastrophenhilfe, sowie mögliche Lösungen, welche die *KatApp* für exakt diese Problematiken bereitstellen soll, angeführt. Der bisherige, analoge Prozess im Katastrophenmanagement ist zwar funktional, weist aber an vielen Stellen Optimierungspotential auf. Durch passende Optimierungen kann Zeit gespart und somit Leben gerettet werden.

Zu Beginn dieser Arbeit wurde bereits erwähnt, wie traditionellerweise Triage-Karten zum Einsatz kommen, um den Verlauf der Triagen zu dokumentieren. Diese Papierkarten werden handschriftlich ausgefüllt und dem Patienten typischerweise um den Hals gehängt. Folgetriagen werden ebenso auf diesen Karten vermerkt. Es wird also der gesamte gesundheitliche Verlauf eines Patienten über Triage-Karten abgebildet. Das manuelle Ausfüllen dieser Karten kann durchaus zeitaufwendig und vor allem auch fehleranfällig sein, vor allem wenn man bedenkt, dass diese Karten zumeist unter höchst stressigen Bedingungen ausgefüllt werden. Die Wahrscheinlichkeit, dass sich in diesen Prozess menschliche Fehler einschleichen, ist relativ hoch. Zudem kann es vorkommen, dass Karten im Verlauf der Katastrophe verloren gehen, oder dass Informationen auf den Karten durch widrige Umstände (zum Beispiel Regen, verschmiertes Blut, etc.) unleserlich werden.

Neben der Problematik analoger Triage-Karten ist die Koordination der Katastrophenhilfe oft sehr herausfordernd. Im Falle einer Katastrophe herrscht meist Chaos, und auf traditionellem Wege müssen Standorte, Zustände und Klassifikationen von Patienten entweder über Funk oder auf mündlichem Wege übertragen werden. Zudem müssen die Einsatzleitung und die Leitstelle stets mit Informationen zur Katastrophe versorgt werden, was weitere, aufwendige Kommunikation erfordert. Kommunikation auf diese Art ist zumeist unkoordiniert und zeitaufwendig. Informationen fließen evtl. mehrfach oder wichtige Informationen gehen in der Flut unter. Andere Informationen müssen womöglich aktiv erfragt oder aufgrund widriger Bedingungen (Lärm, Funklöcher, etc.) mehrfach wiederholt werden.

Eine weitere Problematik ist die Flüchtigkeit der Informationen, die während einer Katastrophe erfasst werden. Patientendaten, die auf Triage-Karten geführt werden, werden nicht über eine Katastrophe hinaus persistiert. Dies gilt ebenso für Triagen oder allgemeine Katastrophendaten. Unter Voraussetzung einer digitalen Repräsentation der Daten können Analysen durchgeführt und somit Prozesse potenziell optimiert werden. Folglich könnte in zukünftigen Katastrophen effizienter gearbeitet werden.

Die *KatApp* wird mit dem Ziel entwickelt, die logistischen Abläufe einer Katastrophe zu optimieren, sodass Patienten schneller versorgt werden können. Sie löst durch die Digitalisierung des Prozesses aber auch weitere der obigen Problematiken.

Durch die *KatApp* müssen Triagen am Schadensort ab sofort nicht mehr analog mit Stift und Papier durchgeführt werden. Stattdessen wird die zugehörige App für die Durchführung der Triagen verwendet. Dies löst nicht nur die Problematiken wie Datenverlust oder Fehleranfälligkeit des analogen Prozesses, sondern verkürzt durch die digitale Erfassung der Daten zudem auch die Kommunikationswege und ermöglicht die Persistierung und Analyse der eingehenden Daten. Wird zum Beispiel eine Triage über die App durchge-

führt und somit Patient und Triage im System erfasst, werden die von der Einsatzleitung und Leitstelle benötigten Informationen über das Dashboard der *KatApp* umgehend zur Verfügung gestellt. Dadurch können alle beteiligten Parteien effizienter arbeiten und die mündliche Kommunikation kann deutlich reduziert werden. Die Fehleranfälligkeit und der Zeitaufwand des Gesamtprozesses wird dadurch ebenfalls reduziert.

Durch die Erfassung von Echtzeit-Daten während der Katastrophe können zudem weitere Prozesse des Katastrophenschutzes optimiert werden. In größeren Katastrophengebieten können Rettungsmittel entsprechend der vorliegenden Daten direkt zu den wichtigsten Lageorten entsandt werden. Dabei handelt es sich um die Orte mit dem größten Patientenaufkommen.

Darüber hinaus bietet die *KatApp* den Vorteil der digitalen Datenerfassung. Im Gegensatz zum analogen Prozess werden alle in der Katastrophe erfassten Daten persistiert. Durch die erfassten Daten sollen in Zukunft wichtige Erkenntnisse zu Abläufen, zeitlichen Engpässen oder anderen Prozessen, die während einer Katastrophe ablaufen, gewonnen werden. Zudem können diverse Triage-Algorithmen auf ihre zeitliche Effizienz untersucht werden.

2.5 Komponenten

Um die Funktionsweise der *KatApp* besser zu verstehen, werden in den folgenden Unterkapiteln Verantwortlichkeiten und Funktionsumfang der individuellen Komponenten der *KatApp* dargelegt. Das *KatApp* Ökosystem umfasst mehrere Anwendungen. Diese Anwendungen lassen sich weiterhin in zwei Kategorien aufteilen, Anwendungen für den Einsatz und Anwendungen für Organisation und Verwaltung. Die App, das Dashboard sowie das Backend werden für den Einsatz verwendet. Die Mandanten- sowie die Adminkonsole werden im Verbund mit dem Backend für die Verwaltung der Organisationen bzw. Mandanten genutzt.

2.5.1 App

Die Smartphone-App der *KatApp* wird im Feld eingesetzt. Sie ist sowohl auf Android als auch auf iOS verfügbar. Mit Hilfe der App werden die Triagen an den Patienten direkt am Schadensort durchgeführt. In der App ist aktuell genau ein spezifischer Triage-Algorithmus implementiert, welcher in Abbildung 2.2 abgebildet ist.

Die App leitet den Nutzer anhand des abgebildeten Algorithmus durch den Triage-Prozess. Der typische Ablauf einer Triage mittels der App sieht dabei folgendermaßen aus: am Einsatzort angekommen wählt der Ersthelfer eines der bereits authentifizierten Smartphones aus, welche im Rettungsfahrzeug für den Einsatz bereitliegen. Zudem nimmt er einige laminierte Papierkarten aus dem Einsatzfahrzeug mit, auf denen QR-Codes abgebildet sind. Hinter jedem dieser QR-Codes verbirgt sich eine einzigartige ID, mit welcher ein späterer Patient eindeutig identifiziert werden kann. Danach begibt sich der Ersthelfer an den Schadensort der Katastrophe und sucht den ersten Patienten auf. Hier weist er dem Patienten einen der QR-Codes zu. Die QR-Codes werden den Patienten, analog zu den Triage-Karten, um den Hals gehängt. Nun greift der Ersthelfer zu seinem Smartphone und scannt den soeben zugewiesenen QR-Code mit Hilfe der App. Der Ersthelfer durchläuft nun den Triage-Algorithmus und erhält dadurch eine von fünf Klassifizierungen (Grün, Gelb, Rot, Schwarz, Blau). Hat der Ersthelfer den Algorithmus durchlaufen, werden die von der App erhobenen Daten mit dem Backend der *KatApp* synchronisiert. Die Triage des Patienten ist nun abgeschlossen und der Ersthelfer begibt sich zum nächsten Patienten. Bei Folgetriagen wird der gleiche Prozess durchlaufen, mit dem Unterschied, dass kein neuer QR-Code vergeben, sondern der bestehende QR-Code des Patienten gescannt wird.

Die App bietet zudem eine Übersicht über alle Triagen, die auf diesem Gerät durchgeführt wurden. So kann später nachvollzogen werden, welcher Ersthelfer welche Triagen durchgeführt hat. Zudem implementiert die App einen Mechanismus, der durchgeführte

Triagen auf dem Ersthelfer-Gerät temporär zwischenspeichert, falls das Backend gerade nicht erreichbar sein sollte. Ein solcher Fall mag zum Beispiel auftreten, wenn Triagen im Keller eines Gebäudes durchgeführt werden, und keine Internetverbindung besteht. Diese temporär gespeicherten Triagen werden dann bei Wiederaufbau der Verbindung mit dem Backend synchronisiert.

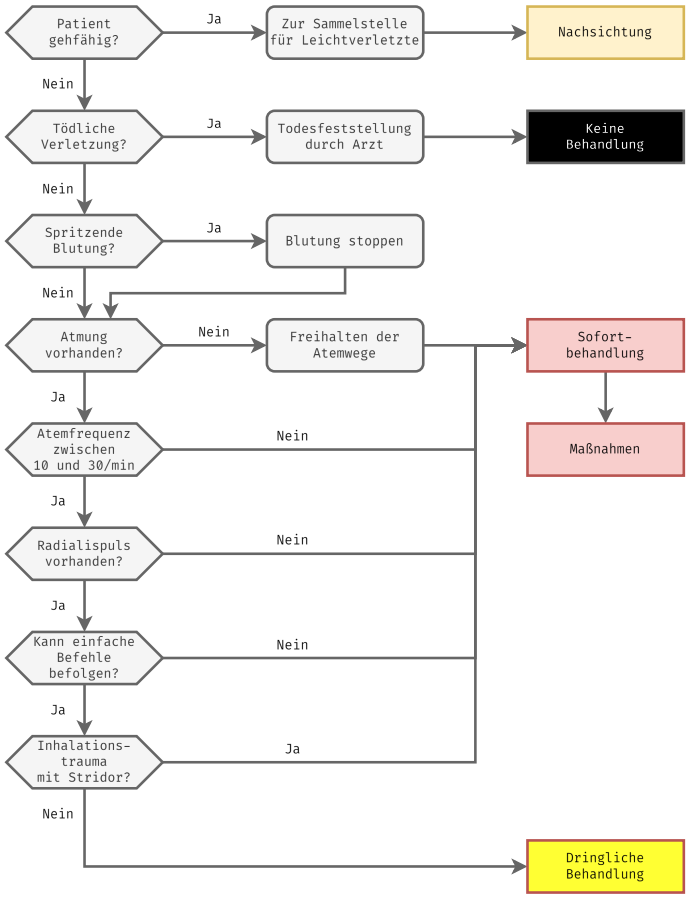

Abbildung 2.2: Vereinfachte Darstellung des mSTaRT Triage-Algorithmus [4]

Der Triage-Algorithmus der *KatApp* ist vollständig aufseiten des Klienten implementiert. Das Backend ist nicht in diesen Prozess involviert, und interessiert sich ausschließlich für die resultierende Klassifikation des Patienten.

Der mSTaRT (modified Simple Triage and Rapid Treatment) Algorithmus ist ein häufiger in Deutschland verwendeter Algorithmus [4]. Die in Abbildung 2.2 abgebildete Variante gilt für erwachsene Personen. Es gibt zudem weitere Varianten des Algorithmus speziell für Kinder oder Nachsichtungen, die leicht von der hier gezeigten Variante abweichen. Auch die App der *KatApp* separiert zwischen erwachsenen Personen und Kindern. Unabhängig von der Zielgruppe soll jedoch die Schwere der Verletzungen möglichst schnell bestimmt werden können. [4] [7]

2.5.2 Dashboard

Das Dashboard ist eine Web-Applikation und wird ausschließlich von der Einsatzleitung und der Leitstelle verwendet. Es bietet einen Überblick über die aktuell laufenden Katastrophen anhand einer Karte.

Der typische Einsatz des Dashboards sieht folgendermaßen aus: die Einsatzleitung kommt am Einsatzort an und öffnet das Dashboard. Anschließend wählt der Einsatzleiter die aktuelle Katastrophe auf der Karte des Dashboards aus und wählt sich in diese ein. Darauffolgend erhält der Einsatzleiter einen Überblick über die zu diesem Zeitpunkt

bereits gescannten Patienten, deren Standorte und deren Klassifikationen. Zukünftig gescannte Patienten werden in Echtzeit mit dem Dashboard synchronisiert. Mit Hilfe dieser Daten kann sich der Einsatzleiter einen Überblick über die Katastrophe und deren Ausmaß verschaffen. Zudem kann der Einsatzleiter die Entwicklung der Klassifikationen und den Verlauf des Standorts individueller Patienten einsehen. Die Leitstelle kann indessen die Daten des Dashboards nutzen, um Rettungsmittel (Rettungswägen, Helikopter, etc.) zu planen. Anhand dieser Daten können zudem die Aufnahmekapazitäten umliegender Krankenhäuser geprüft werden.

2.5.3 Mandantenkonsole

Die Mandantenkonsole ist eine Webanwendung zur Verwaltung der eigenen Organisation innerhalb der *KatApp*. Hier können ein oder mehrere Administratoren einer Organisation Nutzer verwalten. Benutzer, die sich innerhalb der Organisation für die Nutzung des Dashboards registriert haben, können hier explizit durch Administratoren freigeschaltet werden. Zudem wird die Mandantenkonsole für die Generierung von Zugangstokens für Ersthelfer-Geräte verwendet (siehe Unterabschnitt 2.8).

Rollen und Berechtigungen können direkt in der Mandantenkonsole vergeben werden. Diese werden in Zukunft auch konfigurierbar sein, sodass Organisationen ihre eigenen Rollen erstellen und bestehende Rollen anpassen können. Darüber hinaus werden Administratoren hier auch Nutzungsdaten ihrer Organisation einsehen können. Dazu zählen zum Beispiel die durch ihre Nutzung verursachten Kosten.

2.5.4 Adminkonsole

Die Adminkonsole ist eine weitere Webanwendung, welche ausschließlich vom Betreiber (siehe Unterabschnitt 2.7) der *KatApp* verwendet wird. Der Betreiber der *KatApp* stellt die Infrastruktur (entweder selbst oder über einen Cloud-Anbieter) bereit, auf der die *KatApp* ausgeführt wird. Alle Webanwendungen (Dashboard, Mandantenkonsole, Adminkonsole) sowie das Backend werden auf dieser Infrastruktur ausgeführt.

Über die Adminkonsole, können Organisationen für die Nutzung der *KatApp* freigeschaltet werden. Darüber hinaus sind hier Nutzungsdaten und Metriken sämtlicher Organisationen einsehbar. Mit Hilfe dieser Daten kann der Systemadministrator zum Beispiel Rechnungen für individuelle Organisationen ausschreiben. Die Adminkonsole ist in der aktuellen Version der *KatApp* noch nicht verfügbar, repräsentiert aber einen essenziellen Teil des *KatApp* Authentifizierungs- und Autorisierungsablaufes.

2.5.5 Backend

Das Backend der *KatApp* ist verantwortlich für die Datenhaltung, Datenabfrage, Organisationsverwaltung sowie Authentifizierung und Autorisierung. Im Grunde stellt das Backend die gesamte Anwendungslogik der *KatApp* bereit. Das Backend kommuniziert mit jeder einzelnen Komponente im *KatApp* Ökosystem: der App, dem Dashboard, der Mandantenkonsole und der Adminkonsole. Daher werden die Funktionen des Backends unterteilt in Einsatz- und Verwaltungsfunktionen. Genaueres zum Backend wird in Abschnitt 5 beschrieben.

2.6 Funktionsweise

Nachdem in den vorangegangenen Kapiteln die einzelnen Bestandteile der *KatApp* vorgestellt wurden, wird in diesem Kapitel der Ablauf einer Katastrophe mit Hilfe der *KatApp* beschrieben. In Unterabschnitt 2.2 wurde bereits auf den traditionellen Ablauf der Katastrophenhilfe ohne externe Hilfsmittel eingegangen. Nun soll der neue, angestrebte Ablauf des Katastrophenschutzes dargestellt werden.

Im Gegensatz zum traditionellen, analogen Ablauf der Katastrophenhilfe müssen einige Vorbereitungen für den digitalen Ablauf getroffen werden. Um sich überhaupt an einer Katastrophe beteiligen zu können, müssen sich zunächst alle beteiligten Organisationen in der *KatApp* als Mandanten registrieren. Nachdem diese vom Betreiber freigeschaltet wurden, müssen zunächst einige organisatorische Aufgaben erledigt werden. Mitglieder der Einsatzleitung und der Leitstelle müssen sich beim Dashboard registrieren. Der jeweilige Mandantenadministrator kann diese dann über die Mandantenkonsole freischalten und ihnen entsprechende Rollen und Berechtigungen zuweisen, sodass sie zur Nutzung des Dashboards berechtigt sind.

Für die Ersthelfer werden Smartphones und QR-Codes mit einzigartigen Kennungen präpariert, die in den Rettungsfahrzeugen deponiert werden. Die dort hinterlegten Geräte müssen jedoch vorher authentifiziert werden, sodass damit Triagen durchgeführt werden können. Dieser Vorgang läuft über den Mandantenadministrator ab, welcher Zugangstokens für diese Geräte über die Mandantenkonsole generieren kann. Diese Zugangstokens müssen dann manuell in die Geräte eingepflegt werden. Danach sind die Geräte bereit zum Einsatz. Ausgetauscht werden müssen diese Zugangstokens sehr selten bis nie. In der aktuellen Version der *KatApp* sind diese Tokens unbegrenzt gültig. In Zukunft wird die Dauer dieser Tokens durch einen Mandantenadministrator konfigurierbar sein.

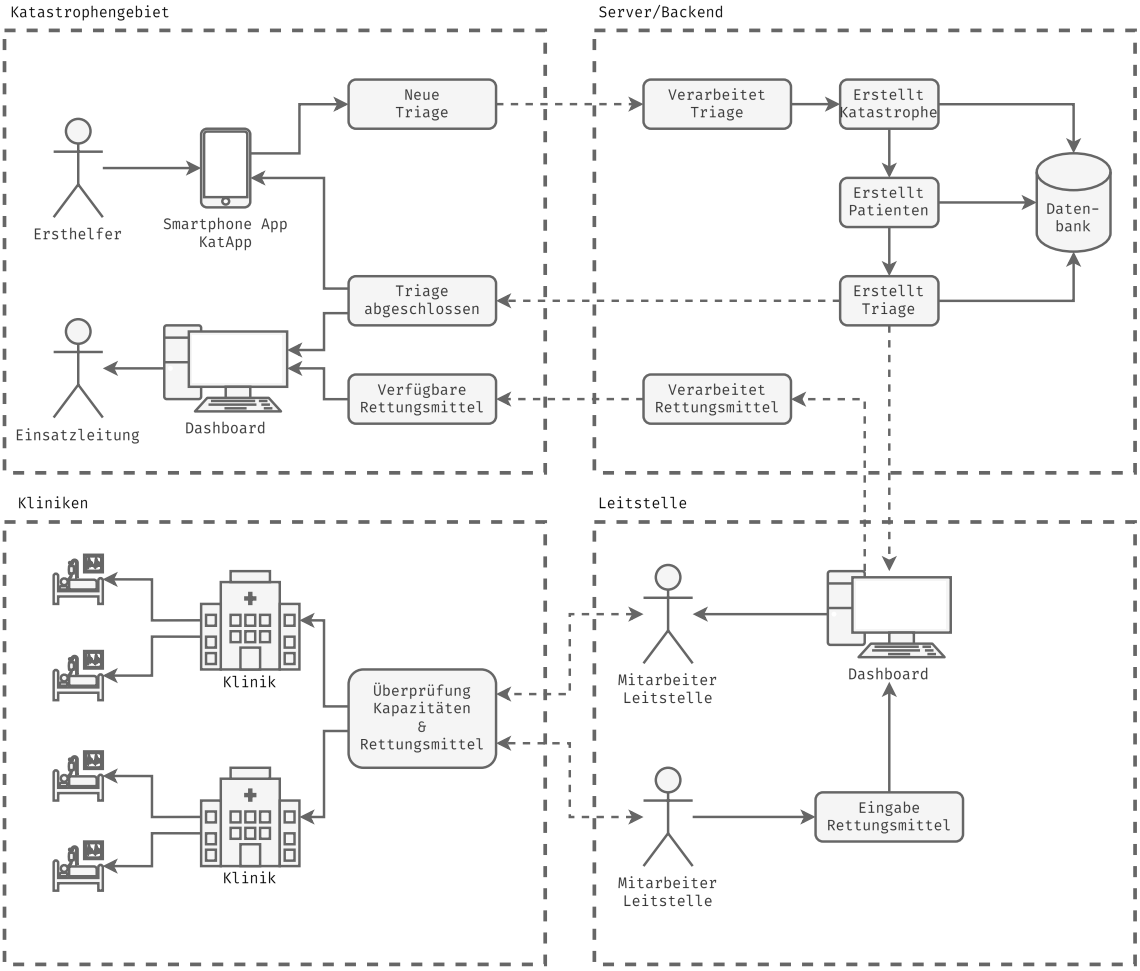

Abbildung 2.3: Funktionsweise der *KatApp*

Abbildung 2.3 gibt einen umfassenden Überblick über die Prozesse, welche im Rahmen des Katastrophenschutzes durch die *KatApp* ausgeführt werden. Im Kontext des Katastrophenschutzes sind vier verschiedene Entitäten an vier verschiedenen Orten beteiligt: die Katastrophenhilfe vor Ort, das Backend, welches in der Cloud bzw. einem externen Server betrieben wird, die Leitstelle, welche den Einsatz koordiniert, und die umliegenden Klini-

ken, welche für die Aufnahme von Notfallpatienten verantwortlich sind. Jede dieser vier Entitäten ist für die Aufrechterhaltung der Katastrophenhilfe von essenzieller Bedeutung.

Die in einer Katastrophe ausgetauschten Informationen umfassen Triagen, Patienten, Lageorte und Ausmaße, verfügbare Rettungsmittel und Weiteres. Wie die *KatApp* diese Informationen abbildet, wird in Unterabschnitt 6.1 genauer beschrieben. Anhand von Abbildung 2.3 lässt sich ableiten, inwiefern die *KatApp* Kommunikationswege des Katastrophenschutzes optimiert. Durch den Einsatz des Systems und der Digitalisierung des Prozesses können Kommunikationswege nun automatisiert werden.

Durch die Triage mittels App kann das System nicht nur die entsprechenden Daten verarbeiten, es können auch relevante Personen über den Neueingang solcher Daten automatisiert informiert werden. Während eines Einsatzes sind Einsatzleitung, Leitstelle und Einweiser permanent mit dem Dashboard der *KatApp* verbunden. Sobald ein Ersthelfer die Triage eines Patienten beendet hat, wird diese automatisch mit den verbundenen Nutzern des Dashboards synchronisiert. Auf diese Art und Weise erhalten sowohl Einsatzleitung als auch Leitstelle Triage- und Patientendaten in Echtzeit. Dadurch wird der Prozess nicht nur übersichtlicher für alle Beteiligten, sondern auch effizienter. Die Leitstelle wiederum kann verfügbare Rettungsmittel umliegender Kliniken in das System einpflegen, sodass Einsatzleitung und Einweiser in Echtzeit über Rettungsmittel benachrichtigt werden können. Folglich kann der Abtransport von Patienten effizienter geplant werden.

2.7 Mandantschaften

Eine Mandantschaft ist der zentrale Bindepunkt zum System. Ohne Mandantschaft gibt es keinen Zugang zur *KatApp*, daher sind sie einer der zentralen Bestandteile des Systems. Mandantschaften wurden aus diversen Gründen eingeführt, welche im Folgenden diskutiert werden.

Betrachtet man die Fachlichkeit der Katastrophenhilfe und die damit einhergehende Rollenverteilung, legt dies bereits die Verwendung eines Mandantensystems nahe. Allein dir Strukturen, welche durch die Teilnahme verschiedenster Organisationen gegeben sind, weisen in diese Richtung. Es gibt durchaus Gedankengänge, ein solches System ohne die Einführung von Mandantschaften zu betreiben. Dadurch wäre nicht nur die Gestaltung der Software-Architektur einfacher, sondern auch der Verwaltungsaufwand der Administratoren würde sich deutlich reduzieren. Geht man jedoch von einem System aus, welches alle möglichen Teilnehmer unter einer „Mandantschaft“ gruppiert, ergeben sich diverse Problemstellungen. Organisationen wie Leitstellen zum Beispiel, welche als Organisation eine spezifische Rolle erfüllen, sind dadurch schwer abzugrenzen und benötigen ein ausgeklügeltes Rollen- und Rechtesystem. Zudem muss beachtet werden, dass die *KatApp* hauptsächlich mit sensiblen medizinischen Daten arbeitet. Wenn es keine Trennung zwischen Organisationen gibt, wie wird dann festgelegt, wer welche Daten einsehen darf? Eine Abgrenzung wäre damit extrem herausfordernd.

Die größere Problematik eines mandantenunabhängigen Systems ist die Bezahlung des Betreibers. Auf technischer Ebene muss es einen Systembetreiber geben, eine Organisation, welche die *KatApp* (in der Cloud) betreibt und somit für den Einsatz zur Verfügung stellt. Im Falle eines mandantenunabhängigen Systems stellt sich also die Frage, wer für die Kosten des Betriebs aufkommt. Unabhängig davon, ob die *KatApp* als Mandantensystem betrieben wird oder nicht, muss es auf jeden Fall eine zentrale Instanz der *KatApp* geben. Ein Eigenbetrieb der *KatApp* für Organisationen ist nicht möglich, da zum einen das technisch ausgebildete Personal fehlt, und zum anderen eine Kooperation unter Mandanten nicht umsetzbar ist, wenn jeder Mandant seine eigene *KatApp* betreibt.

Für den Betreiber der *KatApp* entstehen nicht unerhebliche Kosten für den Betrieb des Systems. Dies hat zur Folge, dass Anwender der *KatApp* einen Preis für die Nutzung zahlen müssen. Ein endnutzerbasiertes Abonnement-Modell kommt hierbei nicht infrage. Das System sollte nicht von beliebigen Personen verwendet werden können. Zudem ist hierdurch eine klare Rollenverteilung während einer Katastrophe schwierig. Wer übernimmt

die Rolle der Ersthelfer, wer übernimmt die Rolle der Einsatzleitung oder der Leitstelle? Gerade da zum Beispiel die Rolle der Leitstelle immer fest vergeben ist, und einer Katastrophe automatisch zugewiesen werden kann, ist die Anwendung eines solchen Modells nicht möglich. Durch den Ausschluss eines endnutzerbasiertes Lizenzmodells wird zudem der Gedanke eine mandantenunabhängigen System mitigiert. Für die Kosten des Systems kommen die Organisationen auf, nicht die Endnutzer.

Generell sind Abonnement-Modelle eher unüblich im medizinischen Bereich. Flexible Modelle werden in diesem Sektor aus diversen Gründen nicht angewendet. Derartige Organisationen betreiben ihre Kostenkalkulationen und Ausgaben typischerweise auf Jahres-Basis. Organisationen benötigen daher einen Gesamtpreis im Voraus. Der Betreiber benötigt daher eine Schätzung der Gesamtkosten für ein Jahr. Diese Beträge können nur aus Erfahrungswerten oder groben Schätzungen gewonnen werden. Der Betreiber erhält die Betriebskosten des Systems vom entsprechenden Cloud-Anbieter, dennoch hilft ihm diese Information nur begrenzt weiter. Verschiedene Organisationen verwenden das System unterschiedlich oft, und verursachen dadurch unterschiedlich hohe Kosten. Daher müssen die Kosten auch für jede Organisation individuell berechnet werden. Der Betreiber kann aus den entstandenen Gesamtkosten, welche er den Rechnungen des Cloud-Anbieters entnehmen kann, unmöglich auf die Einzelkosten für jede Organisation rückschließen, daher müssen die verursachten Kosten je Organisation durch das System selbst berechnet werden.

Die Abrechnung von Organisationen ist also einer der primären Treiber für die Nutzung eines Mandantensystems. Durch die Zuordnung von Mandantschaften zu Organisationen ist stets bekannt, von welcher Organisation ein Systemzugriff ausgeht. Diese Zugriffe können für jeden Mandanten individuell nachverfolgt werden. Daraus lassen sich am Ende die Kosten für jede Organisation berechnen.

Darüber hinaus ist auf fachlicher Ebene zu beachten, dass sensible Informationen wie Patientendaten typischerweise nur von der Organisation einsehbar sein sollten, welche diese erhoben hat. Ein Mechanismus, welcher erfordert, dass bekannt sein muss, welche Organisation die Daten eines Patienten erhoben hat.

Neben der Problematik des Lizenzmodells und der Kostenabrechnung muss sichergestellt werden, dass die *KatApp* nur von autorisiertem, geschultem Personal verwendet wird. In einem mandantenunabhängigen System könnten solche Kompetenzen nur schwer verifiziert werden. Daher ist der Zugang zum System nur über Organisationen möglich, welche für die Katastrophenhilfe zugelassen sind. Das Personal dieser Organisationen ist stets ausgebildet und somit befähigt, an der Katastrophenhilfe teilzunehmen. Dies gilt sowohl für Ersthelfer als auch Einsatzleitung sowie Leitstelle.

Betrachtet man all diese Gegebenheiten, sowohl auf technischer als auch auf fachlicher Ebene, muss eine Korrelation zwischen Patientendaten und Organisation bestehen. Zudem muss stets eine Beziehung zwischen einem Nutzer des Systems und seiner Organisation bestehen, um zu ermitteln, ob ein Nutzer authentifiziert und autorisiert ist. Aus diesen Rahmenbedingungen lässt sich auf ein Mandantenmodell schließen.

2.8 Authentifizierung

Da die *KatApp* aufgrund der Patientendaten vor allem mit sensiblen medizinischen Informationen arbeitet, darf das System nicht öffentlich zugänglich sein. Authentifizierung und Autorisierung sind daher unumgängliche Mechanismen, welche die *KatApp* implementieren muss. Dies hat zur Folge, dass sämtliche Mitarbeiter beteiligter Organisationen authentifiziert und autorisiert sein müssen, um an einer Katastrophe teilzunehmen, unabhängig davon, ob es sich um einen Ersthelfer, einen Einsatzleiter oder einen Mitarbeiter der Leitstelle handelt.

Eine Authentifizierung bzw. Autorisierung ist nur über eine zugelassene Organisation möglich. Durch dieses Vorgehen ist sichergestellt, dass nur Personal mit entsprechenden Kompetenzen Zugang zum System hat. Eine dritte Person ohne medizinische Kenntnisse,

die außerhalb einer zugelassenen Organisation arbeitet, sollte sich nicht als Ersthelfer oder Einsatzleitung in einer Katastrophe beteiligen können. Ersthelfer, welche entsprechende Kompetenzen vorweisen und keiner zugelassenen Organisation angehören (zum Beispiel spontane Hilfe engagierter Bürger), können durchaus behelfende Tätigkeiten übernehmen, aber nicht mit dem System interagieren. Darüber hinaus ist sichergestellt, dass ein Zugriff auf das System vom Betreiber über die Organisation abgerechnet werden kann. Bei authentifizierten Zugriffen ist der Mandant eines Nutzers stets bekannt.

Die *KatApp* verfolgt ein hierarchisches Authentifizierungsverfahren. Dieses Verfahren stellt sicher, dass nur autorisiertes, qualifiziertes Personal Zugang zum System hat. Abbildung 2.4 visualisiert diese Hierarchie.

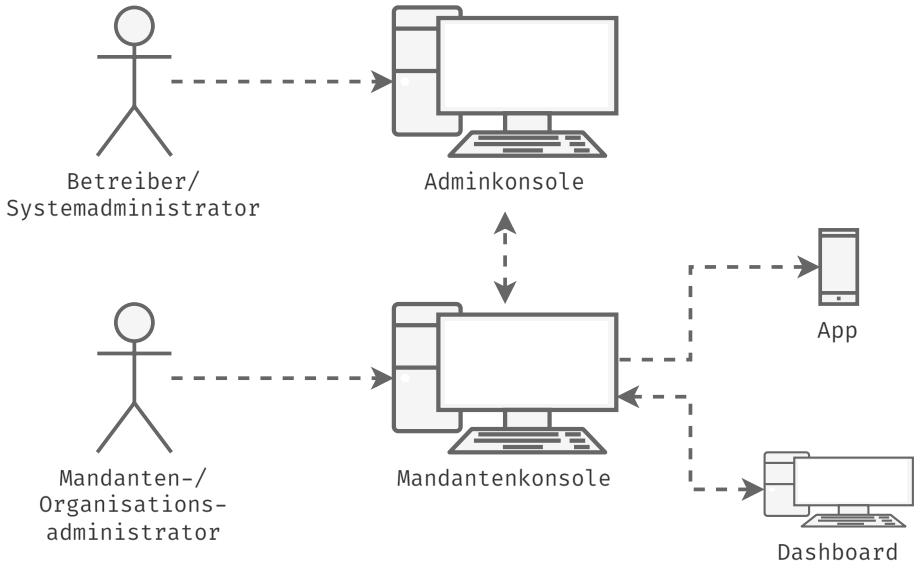

Abbildung 2.4: *KatApp* Authentifizierungs-Hierarchie

Zugelassen werden Organisationen vom Systemadministrator, welcher gleichzeitig der Betreiber des Systems ist. Eine Organisation erhält nur über den Betreiber der *KatApp* Zugang. Organisationen können sich zwar selbstständig für die *KatApp* registrieren, erhalten aber keinen Zugang zum System, bis dieser explizit vom Systemadministrator freigegeben wird. Sobald der Zugang freigegeben wurde, hat der Organisationsadministrator Zugang zur Mandantenkonsole.

Nutzer des Dashboards wie Einsatzleitung oder Leitstelle erhalten ausschließlich über ihre Organisation Zugang zum System. Sie registrieren sich selbständig über das *User Interface* (UI) des Dashboards für ihren Zugang, müssen aber auch hier aktiv vom Administrator der Organisation über die Mandantenkonsole freigeschaltet werden. Mithilfe dieser Vorgehensweise können Zugänge explizit durch einen Menschen verifiziert werden, bevor diese freigegeben werden. In Zukunft wäre es durchaus denkbar, eine automatische Verifizierung über die E-Mail-Adresse eines Nutzers durchzuführen. Da die Mehrheit der Organisationen eigene E-Mail-Domänen nutzt, wäre dieses Verfahren ein effizienterer Weg, um automatisiert verifizierte Zugänge für das Dashboard zu vergeben.

Im Gegensatz zum Dashboard ist die Authentifizierung eines Ersthelfer-Smartphones nicht nutzer-, sondern gerätebasiert. Hinter einem Dashboardnutzer steht immer eine spezifische Person mit individuellen Rechten und Zugängen. Dies kann zum Beispiel ein Mitarbeiter der Einsatzleitung oder der Leitstelle sein. Hinter den Ersthelfer-Geräten, welche für den Einsatz und die Durchführung von Triagen verwendet werden, steht jedoch keine spezifische Person, sondern ein zufälliger Ersthelfer.

Die Geräte (Smartphones) für die Triage-Durchführung sind keine persönlichen Geräte, sondern werden durch die Organisation für den Einsatz zur Verfügung gestellt. In jedem Rettungsfahrzeug wird eine Reihe dieser Geräte platziert, sodass diese im Fall eines

Einsatzes bereit sind. Diese Geräte müssen im Voraus durch die Organisation authentifiziert werden, sodass dies nicht während des Einsatzes gemacht werden muss. Die mobile App der *KatApp* ist auf diesen Geräten vorinstalliert.

Durch die fehlende Verbindung von konkreter Person zu Smartphone werden diese Geräte personenunabhängig authentifiziert. Daher findet für die Freischaltung der Geräte keine traditionelle Authentifizierung statt, sondern die Zugänge werden durch einen Organisationsadministrator über die Mandantenkonsole der *KatApp* generiert. Die Zugangstokens, welche der Administrator dabei erhält, sind extrem lange gültig, damit diese nicht vor jedem Einsatz neu eingepflegt werden müssen.

Auf diese Art und Weise ist zudem sichergestellt, dass jeder authentifizierte Zugriff auf das System einer bestimmten Organisation zugehörig ist. Diese Information ist essenziell, um jede angelegte Ressource einem bestimmten Mandanten zuordnen zu können.

2.9 Autorisierung

Neben der Authentifizierung von Anfragen muss jede im Backend durchführbare Aktion autorisiert werden. Hierfür implementiert das Backend ein klassisches Rollen- und Rechte-System. *Role-Based Access Control* (RBAC) ist ein häufig genutztes Schema für die Nutzerverwaltung in mandantenfähigen Systemen und kommt auch in der *KatApp* zum Einsatz.

In einem RBAC basierten System, gibt es stets eine Menge von Rechten, mit denen jede einzelne Aktion im System abgebildet werden kann. Die Aufteilung dieser Rechte kann je nach System unterschiedlich feingranular sein. Mehrere Aktionen können zum Beispiel unter einem Recht gruppiert werden.

Die *KatApp* verwendet ein äußerst feingranulares System, bei dem jedes Recht mit genau einer Aktion assoziiert wird. Dies ermöglicht die Zuordnung einer Vielzahl von Aktionen zu einer Rolle. Die feine Aufteilung von Rechten in der *KatApp* erlaubt eine hohe Konfigurierbarkeit von Rollen, wodurch sich die Applikation an die individuellen Bedürfnisse ihrer Nutzer anpassen lässt.

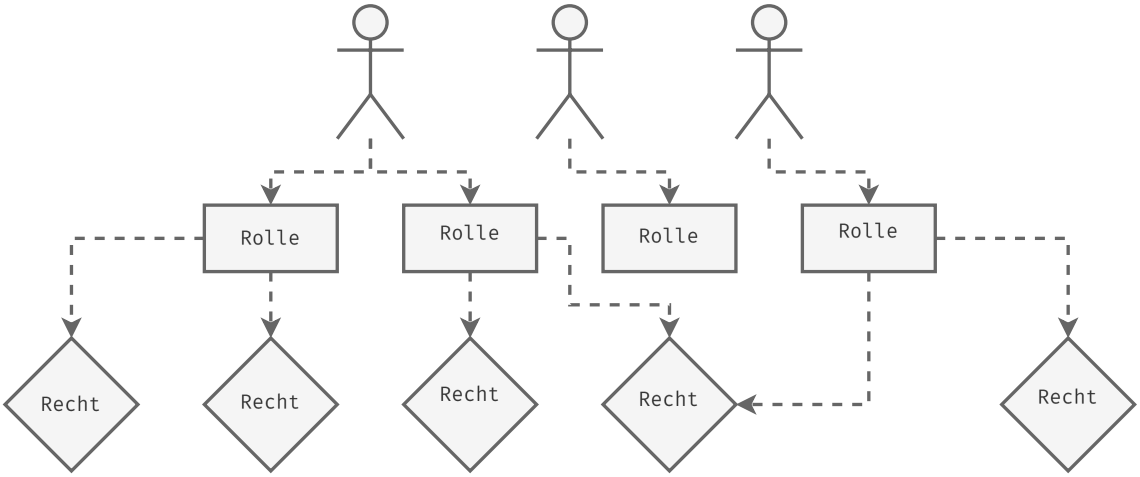

Abbildung 2.5: *KatApp* Rollen & Rechte

Abbildung 2.5 bildet beispielhaft eine Verteilung von Rollen und Rechten ab. Jedem Nutzer können eine oder mehrere Rollen zugewiesen werden. Jede dieser Rollen kann wiederum mehrere Rechte beherbergen. Ein Recht repräsentiert dabei Zugang zu einer feingranular spezifizierten Aktion, welche mit Hilfe dieses Rechts durchgeführt werden darf (zum Beispiel „Triage erstellen“). Rollen und Rechte können über die Mandantenkonsole individuell je Organisation verwaltet werden.

Um den Einstieg in die *KatApp* etwas einfacher zu gestalten, definiert das System einige Standard-Rollen vor. Diese werden bei der initialen Registrierung eines Mandanten automatisch angelegt.

Operator Die *Operator* Rolle bildet den Systemadministrator (Betreiber) der *KatApp* ab. Dieser gewährt Mandanten Zugang zum System und stellt zudem Rechnungen für dessen Nutzung aus. Die Systemadministration wird wie alle anderen Organisationen ebenfalls durch eine Mandantschaft abgebildet. Die *Operator* Rolle wird nur für den administrierenden Mandanten angelegt. Für reguläre Mandanten existiert diese Rolle nicht. Der *Operator* ist die einzige Entität mit Zugang zur Adminkonsole der *KatApp*.

Tenant Admin Die Rolle *Tenant Admin* bildet den Administrator einer Organisation ab. Dieser verwaltet die Zugänge von Organisationsmitarbeitern und stellt Zugangstokens für Ersthelfer-Geräte bereit. Der *Tenant Admin* hat Zugang zur Mandantenkonsole der *KatApp*.

Dashboard User Die Rolle *Dashboard User* ist für die Einsatzleitung und Mitarbeiter einer Leitstelle vorgesehen. Ein *Dashboard User* hat Zugang zum Dashboard der *KatApp*.

Device User Die Rolle *Device User* ist für Ersthelfer vorgesehen. Diese Rolle kann den Zugangstokens zugewiesen werden, welche für die Ersthelfer-Geräte ausgestellt werden. Mit Hilfe dieser Rolle lassen sich Triagen erstellen und Patientendaten abfragen.

Momentan ist die *KatApp* auf ihre vordefinierten Rollen beschränkt. In Zukunft sollen Rollen und Rechte jedoch individuell je nach Organisation angepasst werden können.

3 Mandantenfähigkeit

Nach Bezemer et al., welche über eintausend wissenschaftliche Arbeiten und industrielle Artikel analysiert haben, um eine einheitliche Definition für Mandantenfähigkeit abzuleiten, ist Mandantenfähigkeit folgendermaßen definiert: *Die Mandantenfähigkeit ist eine Systemeigenschaft, bei der mehrere Kunden, sogenannte Mandanten, die Ressourcen eines Systems, wie beispielsweise Dienste, Anwendungen, Datenbanken oder Hardware, gemeinsam nutzen. Ziel dieser Vorgehensweise ist es, die Kosten zu senken und gleichzeitig das System exklusiv auf die Bedürfnisse des jeweiligen Mandanten anzupassen.* [8]

Generell versteht man also unter Mandantenfähigkeit, dass ein System mehrere Kunden bedienen kann, ohne dass diese gegenseitig ihre Daten einsehen können. Gerade Cloud-Plattformen wie AWS oder Azure sind hervorragende Beispiele für mandantenfähige Systeme. Auch die *KatApp* hat in ihrer Entwicklung Inspiration von solchen Systemen bezogen. Es treten aber auch einige Unterschiede zu konventionellen, mandantenfähigen Systemen auf. Dieses Kapitel befasst sich mit üblichen, mandantenfähigen Architekturen, deren Eigenheiten und deren Gemeinsamkeiten sowie Unterschieden zur *KatApp* Architektur.

3.1 Architekturen

Bezemer et al. haben in ihrer umfangreichen Studie unter anderem beobachtet, dass Mandantenfähigkeit hauptsächlich als ein Thema des Bereichs Software-Architekturen betrachtet wird. Aus ihrer Analyse geht zudem hervor, dass Mandantenfähigkeit auf verschiedenen Ebenen eines Systems etabliert werden kann. Mandantenfähigkeit kann so zum Beispiel auf Infrastruktur-Ebene, Plattform-Ebene oder Applikationsebene angewandt werden. [8]

Eine Trennung auf Infrastruktur-Ebene resultiert dabei in zwei Oberkategorien von Systemen, Einzel- und Mehrmandanten-Systemen. Bei Einzelmandanten-Systemen mietet ein Mandant stets seine eigene Infrastruktur. Das heißt, er muss die gemietete Hardware nicht mit anderen Mandanten teilen. Daraus resultiert eine strikte, physische Trennung der Daten. Bei Mehrmandanten-Systemen ist das Gegenteil der Fall. Hier teilen sich mehrere Mandanten dieselbe Infrastruktur. Das heißt auch, dass die Daten der individuellen Mandanten nicht physisch, sondern logisch getrennt sind.

Diese verschiedenen Ebenen können als Isolationsgrade angesehen werden, und werden so auch in der Arbeit von Abdul et al. diskutiert. Die Autoren identifizieren in ihrer Arbeit insgesamt vier unterschiedliche Modelle für mandantenfähige Architekturen, welche Einzel- und Mehrmandantensysteme weiter spezifizieren: das dedizierte Modell, das isolierte Modell, das geteilte Modell sowie das hybride Modell. Dabei beschreiben die Modelle unterschiedliche Grade der Isolation. Alle Modelle garantieren jedoch, dass kein Mandant auf die Daten eines anderen zugreifen kann. Die Modelle unterscheiden sich durch die Ebene, auf der separiert wird. [9]

Um die Modelle für mandantenfähige Systeme entsprechend zu bewerten, werden diverse Kriterien etabliert, mit denen die Vor- und Nachteile der verschiedenen Modelle besser identifiziert werden können. Hierfür muss erarbeitet werden, mit welchen Kriterien eine tatsächliche Bewertung der Modelle stattfinden kann.

Eines der primären Attribute von mandantenfähigen Systemen ist die effizientere Nutzung von zur Verfügung stehenden Ressourcen. Diese Eigenschaft ist überhaupt einer der initialen Treiber, weswegen mandantenfähige Systeme existieren. Verschiedene Isolationsgerade resultieren in unterschiedlich effizienter Nutzung von Ressourcen [9].

Die Gewährleistung der Sicherheit stellt insbesondere für Mandanten, die sich dieselben Ressourcen mit anderen Mandanten teilen, eine essenzielle Herausforderung dar. Dabei ist die Sicherheit der Daten von besonderer Relevanz. Das System sollte dabei durch logische Separation verhindern, dass Mandanten die Daten anderer Mandanten einsehen können. Zudem sollten Aktionen anderer Mandanten nicht zu Datenverlusten auf eigener Seite führen. Ob die Daten eines Mandanten als sicher zu einzustufen sind, wenn ein anderer Mandant kompromittiert wurde, ist in diesem Kontext von entscheidender Bedeutung.

Je nachdem, wie stark oder schwach die Ressourcen der Mandanten voneinander isoliert sind, kann zudem die Skalierbarkeit, Verfügbarkeit oder Stabilität des Systems durch einzelne Mandanten beeinflusst werden. Eine hohe Auslastung des Systems durch einen einzelnen Mandanten kann dazu führen, dass für andere Mandanten weniger Ressourcen zur Verfügung stehen.

Darüber hinaus kann die Konfigurierbarkeit eines Systems für einen Mandanten von großer Bedeutung sein. Die Konfigurierbarkeit mandantenfähiger Modelle ist jedoch stark vom Isolationsgrad abhängig.

3.1.1 Dediziertes Modell

Beim dedizierten Modell (Abbildung 3.1) handelt es sich um ein Einzelmandantensystem. Jedem Mandanten wird seine eigene, dedizierte Infrastruktur zugeteilt. Auf jeder Infrastruktur wird genau eine Instanz der Applikation betrieben, es wird also auf Instanz-Ebene getrennt. In anderen Worten, Mandanten teilen sich weder Applikations- noch Datenbankinstanzen. Infolgedessen erreicht das dedizierte Modell den höchsten Isolationsgrad unter den mandantenfähigen Architekturen. [9]

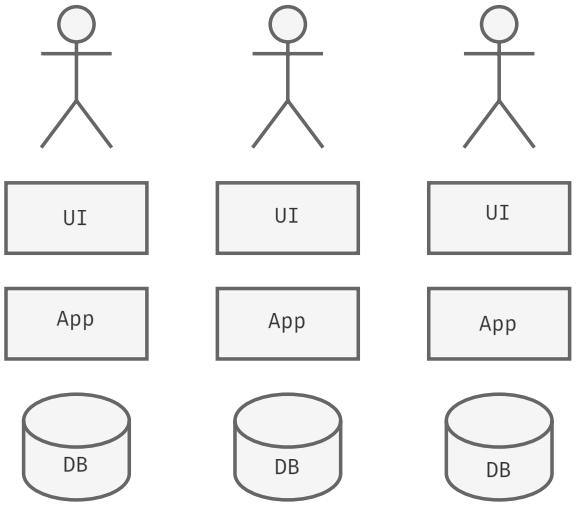

Abbildung 3.1: Mandantenfähigkeit: Dediziertes Modell [9]

Durch seinen hohen Isolationsgrad eignet sich das dedizierte Modell vor allem für Applikationen, bei denen Sicherheit und Datenschutz essenzielle Faktoren sind. Durch Isolation auf Infrastruktur-Ebene ist es äußerst unwahrscheinlich, dass Angriffe und potenzielle Datenlecks andere Mandanten betreffen. Zudem wird es oft als sehr stabiles Modell angesehen, da ein Systemausfall nie mehr als einen Mandanten betrifft. So haben zum Beispiel Ausfälle oder Wartungsarbeiten keine Auswirkungen auf andere Mandanten. Darüber hinaus wird die Performance des Systems nicht durch Aktivitäten anderer Mandanten beeinflusst. Im Gegensatz zu Mehrmandantensystemen haben Mandanten im dedizierten Modell mehr individuelle Freiheiten, sowohl was Hardware als auch Software betrifft. Die Umgebung kann somit auf jeden Mandanten individuell zugeschnitten werden. [9]

Dadurch, dass Applikationen und Datenbanken auf Instanz- und Infrastruktur-Ebene isoliert werden, muss aber auch wesentlich mehr Hardware betrieben werden, als es bei Mehrmandantensystemen der Fall ist. Dies treibt vor allem die Kosten für Mandanten in die Höhe. Ein weiteres Problem dieses Modells ist die ineffiziente Nutzung von Ressourcen. Es ist unwahrscheinlich, dass jeder Mandant seine verfügbaren Ressourcen zu hundert Prozent ausnutzt. Ein Großteil der Ressourcen befindet sich daher oft im Leerlauf. [9]

3.1.2 Isoliertes Modell

Beim isolierten Modell (Abbildung 3.2) teilen sich die Mandanten dieselbe Infrastruktur, weswegen dieses Modell zu den Mehrmandantensystemen zählt. Zudem wird neben der Applikations-Instanz auch dieselbe Datenbank verwendet. Die Daten der Mandanten werden jedoch pro Mandant und nach Schema voneinander getrennt. Dies resultiert in der Nutzung von getrennten Tabellen. [9]

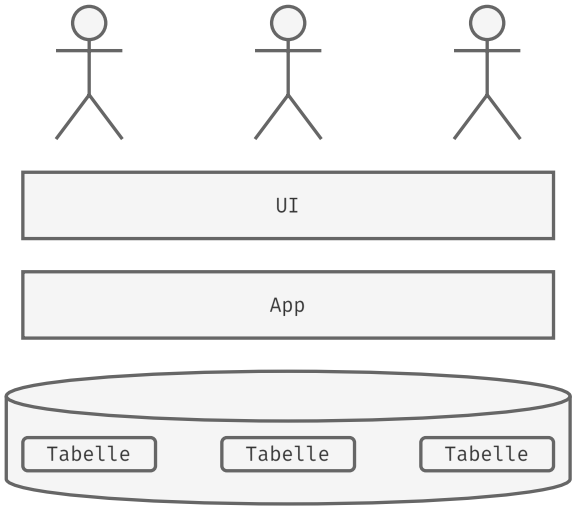

Abbildung 3.2: Mandantenfähigkeit: Isoliertes Modell

Im Vergleich zum dedizierten Modell erweist sich dieser Ansatz als kosteneffizienter für die Mandanten. Die verfügbaren Ressourcen können hier wesentlich besser ausgelastet werden, was zu einer Reduktion der Gesamtkosten für den Betreiber und folglich auch für die individuellen Mandanten führt. Das Teilen von Ressourcen kann jedoch auch ein mögliches Problem dieses Modells sowie aller anderen Mehrmandantensysteme darstellen. Nutzt ein einzelner Mandant übermäßig Ressourcen, so stehen den übrigen Mandanten entsprechend weniger Ressourcen zur Verfügung. Dies kann Auswirkungen auf Verfügbarkeit und Skalierbarkeit des Systems haben. [9]

Des Weiteren ist zu berücksichtigen, dass das isolierte Modell nicht den gleichen Grad an Sicherheit gewährleistet wie das dedizierte Modell. Ein potenzieller Angriff auf einen Mandanten könnte potenzielle Konsequenzen für andere Mandanten nach sich ziehen. Dies betrifft insbesondere die Daten eines Mandanten. Bei einem potenziellen Datenleck könnten folglich auch die Daten anderer Mandanten betroffen sein. [9]

Für den Betreiber der Infrastruktur besteht zudem das Risiko eines erhöhten finanziellen Schadens. Ein Ausfall der Infrastruktur, beispielsweise durch einen Angriff oder einen Hardware-Schaden, führt zu einem größeren finanziellen Schaden als beim dedizierten Modell, da mehr als ein Mandant von dem Ausfall betroffen ist.

Im Vergleich zum dedizierten Modell ist zudem eine eingeschränkte Konfigurierbarkeit auf individueller Ebene zu verzeichnen. Dies betrifft besonders die Infrastruktur. Bei Mehrmandantensystemen ist diese für gewöhnlich nicht durch den Mandanten wählbar, sondern wird durch den Betreiber bestimmt. Mandanten nehmen dies aber oft zugunsten eines geringeren Preises in kauf.

3.1.3 Geteiltes Modell

Das Konzept des geteilten Modells (Abbildung 3.3) geht einen Schritt weiter. Im Rahmen dessen werden die Daten sämtlicher Mandanten in identischen Tabellen gespeichert, wobei ausschließlich die Einträge je nach Mandant getrennt werden. Dies impliziert, dass jeder Datenbankeintrag einem Mandanten zugewiesen werden kann. In der Konsequenz lassen sich Datenzugriffe auf einer logischen Ebene separieren. [9]

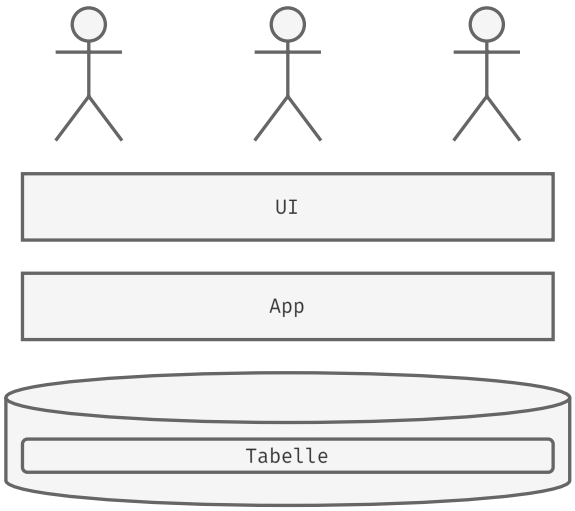

Abbildung 3.3: Mandantenfähigkeit: Geteiltes Modell

Im Vergleich zu den bisherigen Modellen stellt diese Variante die kostengünstigste dar. Das geteilte Modell teilt sich die meisten Risiken mit dem isolierten Modell. Auf Datenebene besteht allerdings zusätzlich die Gefahr, dass Daten anderer Mandanten aufgrund eines einzelnen korrumpiert werden, da alle auf dieselben Datenbank-Tabellen zugreifen. Zudem gestalten sich Backups und Wiederherstellungsprozesse wesentlich komplexer, da alle Mandanten auf diesem Server durch mögliche Ausfälle oder Wartungsaufgaben betroffen sind. [9]

3.1.4 Hybrides Modell

In der Arbeit von Abdul et al. wird zudem ein weiteres, hybrides Modell vorgestellt (Abbildung 3.4). Dieses kombiniert die Aspekte des isolierten und des geteilten Modells, indem es eine allgemeine Instanz mit geteilter Datenbank bereitstellt, was einer Umsetzung des geteilten Modells mit geteilten Tabellen entspricht. Zusätzlich wird eine mandantenbezogene Instanz mit eigener Datenbank bereitgestellt. [9]

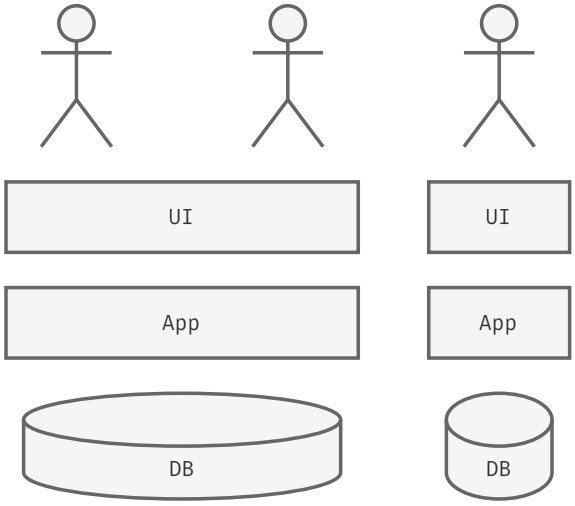

Abbildung 3.4: Mandantenfähigkeit: Hybrides Modell

Hier teilen sich sowohl Vorteile als auch Risiken basierend auf den ausgelagerten Applikationen. Dieses Modell hat den Vorteil, dass Systeme je nach Mandant individuell optimiert werden können. Dadurch erhöhen sich aber auch die Kosten für den jeweiligen Mandanten. [9]

3.2 KatApp und Mandantenfähigkeit

Typischerweise treten mandantenfähige Systeme in einer der vier Architekturen auf, die in der Arbeit von Abdul et al. beschrieben werden. Der entscheidende Unterschied liegt dabei im Isolationsgrad. In diesem Zusammenhang stellt sich die Frage, welche der vier Architekturen für die *KatApp* geeignet scheint.

Aufgrund der Tatsache, dass die *KatApp* vor allem mit medizinischen Daten umgeht, und Datenschutz daher höchste Priorität einräumt, mag das dedizierte Modell zunächst sinnvoll erscheinen. Physisch getrennte Infrastruktur und separate *KatApp* Instanzen würden ein hohes Maß an Datensicherheit garantieren. Aufgrund des hohen Isolationsgrades wäre jedoch eine Kooperation zwischen mehreren Mandanten praktisch unmöglich. Katastrophendaten könnten, bedingt durch getrennte Datenbank-Instanzen, nur doch komplexe Logik oder überhaupt nicht zwischen Mandanten geteilt werden. Das Konzept von Einzelmandantensystemen scheint also für die *KatApp* ausgeschlossen. Benötigt wird ein Mehrmandantensystem, welches das Teilen von Daten im Rahmen einer Katastrophe ermöglicht.

Die Arbeit von Abdul et al. geht in vielerlei Hinsicht davon aus, dass Daten zwischen Mandanten stets getrennt werden, sei es eine Trennung auf physischer oder logischer Basis. Die *KatApp* negiert dieses Konzept jedoch in einigen Aspekten aufgrund der nötigen Kooperationsfähigkeit. Betrachtet man die verbleibenden mandantenfähigen Architekturen unter diesem Aspekt, kommt entweder das geteilte oder ein hybrides Modell für die *KatApp* infrage.

Eine Umsetzung der geteilten Architektur würde sich in diversen Aspekten einfacher gestalten. Die Daten aller Mandanten werden in derselben Tabelle gehalten und logisch durch das Backend getrennt. Diese logische Barriere kann nach Bedarf durch das Backend aufgehoben werden, zum Beispiel bei einer Katastrophe, an der mehrere Mandanten beteiligt sind. Diese Aufhebung kann dann zeitlich auf die Katastrophe sowie nur auf relevante Daten begrenzt werden.

Ein hybrides Modell hätte den Vorteil, dass nur relevante Daten zwischen Mandanten teilbar sind. Dafür muss die Vorlage der Architektur des hybriden Modells aus der Arbeit von Abdul et al. leicht angepasst werden. Anstelle von getrennten UI- und Applikations-Instanzen verwendet man jeweils eine gemeinsame Instanz, nur die Datenbank-Instanzen werden getrennt. Infolgedessen hat jeder Mandant eine eigene Datenbank für mandantenspezifische Daten (zum Beispiel Rollen- und Rechteverwaltung über die Mandantenkonsole). Für Katastrophendaten (Patienten, Triagen, etc.) hingegen verwendet man eine geteilte Datenbank, welche wiederum durch das Backend logisch abgeschirmt wird.

Dadurch, dass das Backend im Falle eines hybriden Modells jedoch, basierend auf der jeweiligen Situation, auf unterschiedliche Datenbanken zugreifen müsste, hätte dies komplexe Anwendungslogik zur Folge, was eine intensive Belastung für zeitliche und monetäre Ressourcen bedeuten würde. Zudem müssten zusätzliche Ressourcen (Datenbanken, Blockspeicher, etc.) betrieben werden, um das hybride Modell abzudecken. Daher hat man sich für ein geteiltes Modell entschieden.

3.3 Herausforderungen

Trotz der Mehrmandantenfähigkeit der *KatApp* bestehen einige Herausforderungen im Umgang mit Katastrophen. Sollen mehrere Mandanten an einer Katastrophe zusammenarbeiten, muss die logische Trennung für selektierte Daten (Katastrophen, Patienten, Triagen) zumindest temporär aufgehoben werden. Daraus ergeben sich jedoch weitere Fragen, wie:

- Wird die logische Trennung der Daten für Katastrophen permanent oder nur die Dauer einer Katastrophe aufgehoben, und wenn ja, wann wird die logische Trennung wieder eingesetzt?

- Wird die logische Trennung der Daten für alle oder nur für selektierte Mandanten aufgehoben?

Zunächst muss man bedenken, dass die Nutzbarkeit der *KatApp* an erster Stelle steht. Das System sollte so einfach und so schnell wie möglich bedienbar sein, sodass es weder die Einsatzleitung noch die Ersthelfer in ihrer Arbeit behindert. Aus Kundenperspektive gilt also, je einfacher, desto besser. Gleichermaßen muss man aber beachten, dass es sich hierbei um sensible medizinische sowie persönliche Daten handelt.

Aus Kundenperspektive betrachtet scheint es sinnvoll, zu einer Katastrophe hinzustoßende Mandanten ohne explizite Freigabe an der Katastrophe teilnehmen zu lassen. Diese Lösung ist einfach und erfordert kein zusätzliches Handeln der Mandanten. Im Grunde hätte dies zur Folge, dass Katastrophen für alle Ersthelfer (Nutzer der App) bedingungslos zugänglich wären (sofern sie über einen Mandanten authentifiziert und autorisiert sind). Aus technischer Sicht wäre diese Lösung ebenfalls leicht umsetzbar, da die mandantenbezogene Beschränkung für die Erstellung von Triagen und die Abfrage von Patientendaten einfach aufgehoben werden kann. Aus Sicht der Datensicherheit jedoch muss man beachten, dass jeder App-Nutzer (von jedem Mandanten) Zugriff auf Patientendaten hat, sofern er dessen eindeutige Identifikationsnummer kennt. Da die Identifikationsnummern aber erst während und nur für die Dauer einer Katastrophe vergeben werden, scheint dies ein zu vernachlässigendes Problem zu sein. Eine Dritte Person kennt die vergebenen Identifikationsnummern nicht und kann somit auch nicht unvorhergesehen auf Patientendaten zugreifen.

Eine Alternative zu diesem Ansatz wäre die explizite Freigabe von Katastrophen für andere Mandanten. Der Mandant, der bei einer Katastrophe als Erstes vor Ort ist und den ersten Patienten einscannt, wäre demnach der „Inhaber“ der Katastrophe. Andere Mandanten könnten der Katastrophe also nicht beitreten, es sei denn der Zugang zu dieser Katastrophe wird vom Inhaber explizit freigeschaltet. Mit diesem Ansatz hätte man mehr Kontrolle über den Zugang zu Triage- und Patientendaten, man hätte jedoch auch einen wesentlich höheren organisatorischen Aufwand während einer Katastrophe. Alle teilnehmenden Organisationen müssten sich beim Inhaber-Mandanten melden, sodass dieser die Organisationen über deren eindeutige Kennung freischalten kann. Auf technischer Seite hätte dies zudem den Nachteil, dass bei jeder Triage zuerst überprüft werden muss, ob die Organisation des Ersthelfers für die Katastrophe freigegeben wurde. Zudem müsste die Zugangsfreigabe selbst implementiert werden. Die gewonnene Zugangskontrolle käme also auf Kosten von Zeit und Laufzeit, und macht das System zudem umständlicher zu verwenden.

Im Gegensatz zu App-Nutzern (Ersthelfer) muss der Zugang für Dashboard-Nutzer (Einsatzleitung oder Leitstelle) auf anderem Wege reguliert werden. In einer Katastrophe gibt es stets mehrere Ersthelfer (von unterschiedlichen Mandanten), aber immer nur eine Einsatzleitung (gestellt von einem Mandanten) und eine Leitstelle (gestellt von einem weiteren Mandanten). Welcher Mandant übernimmt also die Einsatzleitung, und welcher die Leitstelle?

Generell erscheint es sinnvoll, dem Mandanten, der als Erstes vor Ort ist, die Einsatzleitung zuzuweisen. Dadurch hätten nur Dashboard-Nutzer des „ersten“ Mandanten Zugang zur Katastrophe. Auf technischer Ebene wäre dies leicht umsetzbar. Ist jedoch immer garantiert, dass der Mandant des Ersthelfers, welcher den ersten Patienten eingescannt, auch die Einsatzleitung stellt?

Grundsätzlich können auch hier die zwei Ansätze für App-Nutzer aufgegriffen werden. Entweder, der Inhaber-Mandant muss andere Organisationen für die Verwendung des Dashboards bei dieser Katastrophe freischalten, oder man gewährt jeder teilnehmenden Organisation Zugang zur Dashboard-Übersicht. Eine Organisation gilt dann als „teilnehmend“, wenn mindestens eine Triage eines Organisationsmitarbeiters in der aktuell laufenden Katastrophe durchgeführt wurde. Da das Dashboard aktuell ausschließlich Daten liest und darstellt und keine Daten schreibt, wäre es durchaus eine akzeptable Variante, allen teil-

nehmenden Organisationen Zugriff auf die Katastrophe zu gewähren, zumal diese Lösung technisch einfacher umsetzbar wäre.

Dieser Ansatz deckt jedoch nicht den Zugang der Leitstelle ab. Leitstellen werden deutschlandweit nach Landkreis vergeben. Eine Leitstelle wird nach diesen Bedingungen daher nie Teilnehmer einer Katastrophe sein. Der Zugang für Leitstellen müsste daher entweder manuell durch Katastrophenteilnehmer vergeben oder generell für Leitstellen freigeschaltet werden. Eine generelle Freigabe für Leitstellen ist jedoch auch nicht zielführend, da Leitstellen außerhalb eines relevanten Landkreises keinen Zugang zu diesen sensiblen Daten haben sollten. Eine automatische Vergabe der Leitstelle wäre somit nur über die Verwendung externer Dienste für Geokodierung möglich.

Die logische Trennung sollte aufgrund von Datenschutz nicht permanent aufgehoben werden, sondern auf die Dauer einer Katastrophe begrenzt sein. Die erste Triage markiert dabei stets den Beginn einer Katastrophe. Aufgrund der festgelegten Abläufe des Katastrophenschutzes kann die letzte Triage jedoch nicht das Ende markieren. Eine Katastrophe ist erst dann beendet, wenn alle Patienten sicher in den örtlichen Kliniken untergebracht wurden. In der *KatApp* gibt es aber keinen Auslöser, welches dieses Ereignis markieren würde. Das Ende einer Katastrophe automatisiert festzustellen, ist daher alles andere als trivial.

Auf manuellen Wege würde sich das Problem durch ein vom Nutzer ausgelöstes Ereignis lösen lassen. Die Einsatzleitung könnte eine Katastrophe über das Dashboard manuell beenden. Damit liegt das Ende der Katastrophe und der Wiedereinsatz der logischen Trennung der Daten in menschlicher Hand und somit außer Verantwortung des Systems.

Eine weitere Frage, die sich daraus ergibt, ist, wer hat nach Beendigung einer Katastrophe noch Zugriff auf die Daten? Da die *KatApp* aufgrund der digitalen Erfassung der Triage- und Patientendaten auch analytische Zwecke erfüllen soll, müssen die Daten einer Katastrophe auch im Nachhinein entweder zugreifbar oder exportierbar sein. Haben alle Organisationen, welche an der Katastrophe teilgenommen haben, Zugriff auf die Daten? Hat nur der Inhaber-Mandant Zugriff auf die Daten? All diese Herausforderungen müssen im Rahmen eines Lösungsmodells in Betracht gezogen werden.

Diverse dieser hier angeführten Aspekte müssen zudem mit Experten aus dem medizinischen Bereich sowie den Kunden der *KatApp* diskutiert werden. All diese Überlegungen müssen daher über Kunden- und Expertengespräche reevaluiert werden. Diese Aspekte werden genauer in Abschnitt 7 diskutiert.

4 Verwandte Arbeiten

In diesem Kapitel werden verwandte Arbeiten im Bereich Gesundheitswesen, Cloud und Mandantenfähigkeit analysiert. Keine dieser Arbeiten steht in direkter Korrelation zum Funktionsumfang der *KatApp*, dennoch greifen die Arbeiten Aspekte auf, die für die *KatApp* relevant sind. Es wird auf Thematiken wie Datensicherheit in mandantenfähigen Systemen und den Einsatz von Cloud-Computing für Katastrophenmanagement eingegangen.

4.1 Überblick

Singhal et al. diskutieren in ihrer Arbeit diverse Aspekte des Cloud-Computings, welche speziell für das Management von Katastrophen relevant sind, darunter Echtzeit-Datenverarbeitung, Skalierbarkeit und Kommunikation [10]. Velev et al. diskutieren weitere Punkte in den Themenbereichen Cloud und Notfall-Management, wobei vorwiegend auf Anforderungen an die Cloud-Anbieter eingegangen wird [11]. Während sich die Autoren dieser Arbeiten auf die nützlichen Aspekte des Cloud-Computings für das Katastrophenmanagement konzentrieren, diskutieren Anwar et al. kritische Sicherheitsaspekte, die in mandantenfähigen Cloud-Anwendungen, bezogen auf Gesundheitsdaten, auftreten können [12]. Brown et al. gehen des Weiteren auf generelle Sicherheitsaspekte von mandantenfähigen Systemen ein [13].

4.2 Cloud & Katastrophenmanagement

In diesem Unterkapitel werden Arbeiten beleuchtet, die auf die Nutzung der Cloud für Katastrophen- bzw. Notfallmanagement eingehen. Hier werden nützliche Aspekte der Cloud sowie potenzielle Risiken für diesen Anwendungsfall hervorgehoben.

Singhal et al. gehen auf vier verschiedene Aspekte des Cloud-Computings ein, welche für das Katastrophenmanagement von Bedeutung sind: Echtzeit-Datenerfassung und Analyse, Skalierbarkeit und Ressourcenbereitstellung, Kollaboration und Kommunikation sowie Simulation und Modellierung [10]. Die *KatApp* macht sich genau drei dieser Aspekte zunutze. Echtzeit-Datenverarbeitung wird über die Triagierung von Patienten mithilfe der App erreicht. Hierbei werden Patienten- und Geodaten erfasst. Zudem werden die erfassten Daten über das Dashboard der *KatApp* - ebenfalls in Echtzeit - visualisiert. Infolgedessen können diverse Aspekte einer Katastrophe in Echtzeit überwacht werden. Für maximale, dynamische, horizontale Skalierung nutzt die *KatApp* AWS *Lambda*. Durch den funktionalen Aufbau des Backends und die dadurch bestehende Modularität lassen sich individuelle Funktionen des Backends nach Bedarf skalieren (siehe Abschnitt 5). Über das Dashboard werden außerdem nicht nur Kommunikationswege komplett eingespart, sondern zudem die Kommunikationswege zwischen Ärzten, Einsatzleitung und Leitstelle vereinfacht und verkürzt. Eine Simulation bzw. Modellierung von Katastrophen gehört nicht zum geplanten Umfang der *KatApp*, wäre aber mit den durch die *KatApp* erfassten Daten durchaus möglich. Die *KatApp* etabliert also drei der vier von Singhal et al. aufgeführten Punkte im Bereich Cloud für Katastrophenmanagement [10].

Während Singhal et al. vorwiegend auf neue Möglichkeiten, die durch die Cloud im Bereich Katastrophenhilfe entstehen, eingehen, beleuchten Velev et al. unter anderem Risiken, die damit einhergehen. Dafür stellen die Autoren gewisse Anforderungen bereit, die ein Cloud-Anbieter erfüllen sollte, um für einen Einsatz im Bereich Katastrophenhilfe in Betracht gezogen werden zu können [11]. Cloud-Anbieter sind unter anderem verantwortlich für Datensicherheit, -vertraulichkeit und Verschlüsselung [11], Aspekte, die auch von Singhal et al. in ihrer Arbeit angeführt werden [10]. Da verschiedene Cloud-Anbieter unterschiedliche Garantien hinsichtlich dieser Aspekte geben, sollten die Mindestanforderungen für diese Aspekte vor Auswahl des Cloud-Anbieters herausgearbeitet werden, sodass eine fundierte Entscheidung getroffen werden kann. AWS ist nicht nur konform mit der *Datenschutz-Grundverordnung* (DSGVO), sondern auch die in der *KatApp* verwendete *Dy-*

namoDB ist standardmäßig verschlüsselt [14] [15]. AWS bietet also eine stabile Grundlage in puncto Datensicherheit. Velev et al. betonen zudem, dass ein Cloud-Anbieter über einen Krisenbewältigungsprozess verfügen sollte, um seine Infrastruktur auch unter widrigen Bedingungen bereitstellen zu können [11]. AWS verfügt über diverse Risiko-Management Prozesse, um eine stabile Infrastruktur bereitzustellen [16].

4.3 Mandantenfähigkeit & Zugangskontrolle

Anwar et al. erörtern in ihrer Arbeit weitere Sicherheitsaspekte für Software aus dem Gesundheitsbereich, speziell für mandantenfähige Systeme. Dabei ist die Zugangskontrolle einer der wichtigsten Kritikpunkte, besonders im Angesicht eines geteilten, mandantenfähigen Modells [12]. Auch Brown et al. betonen diesen Sicherheitsaspekt, wobei die Autoren Isolation auf physischer Ebene empfehlen [13]. In einem geteilten Modell werden die Daten aller Mandanten in derselben Datenbank gespeichert und müssen daher durch Anwendungslogik getrennt werden. Bei den gespeicherten Informationen handelt es sich um persönliche und medizinische Patientendaten, weswegen besonders achtsam mit diesen Daten umgegangen werden muss. Zwei Kriterien müssen daher von der *KatApp* unbedingt erfüllt werden, Authentifizierung bzw. Autorisierung und Mandantenzugehörigkeit. Mit Hilfe dieser Mittel kann eine solide Zugangskontrolle implementiert werden. Anwar et al. empfehlen für diesen Zweck eine Implementierung von RBAC, ein Konzept, welches auch von der *KatApp* verwendet wird (siehe Unterabschnitt 2.9).

4.4 Forschungslücken

Während die in diesem Kapitel angeführten Arbeiten recht allgemeine Prinzipien erörtern, die im Zusammenhang mit der *KatApp* stehen, ist die *KatApp* selbst doch eher ein Nischenprodukt, zu dem bisher keine (öffentlichen) Forschungen bestehen. Diverse Fragen, im speziellen Implementierungsdetails, bleiben daher von der öffentlichen Domäne unbeantwortet oder unerforscht. Dazu gehören unter anderem Lösungen wie die Kooperation unter Mandanten (speziell im medizinischen Sektor) oder spezifische Probleme wie die automatisierte Vergabe einer Leitstelle sowie die Feststellung eines Katastrophenendes. Lösungen für derartige Herausforderungen müssen daher in Zusammenarbeit mit Experten des jeweiligen Bereichs erarbeitet werden. Gefundene Lösungsansätze müssen implementiert und im Feld getestet werden, um deren Validität zu prüfen.

5 Architektur

Dieses Kapitel befasst sich mit der Architektur des *KatApp* Backends. Hier werden Designentscheidungen, Architekturmuster, Komponenten und Technologien herangeführt. Die Architektur der App bzw. des Dashboards sowie der Admin- und Mandantenkonsole ist für diese Arbeit nicht direkt von Bedeutung und wird nur dort angeführt, wo Relevanz für das Backend besteht.

5.1 Qualitätsziele

Die bestehende Architektur der *KatApp* leitet sich aus den folgenden Qualitätszielen ab. Diese Qualitätsziele ergeben sich aus den fachlichen, nicht-funktionalen Anforderungen, die vom System erfüllt sein sollten.

Verfügbarkeit Die Verfügbarkeit der *KatApp* sollte allzeit (höher als 99 %) garantiert sein. Tritt eine Katastrophe auf und die *KatApp* ist nicht verfügbar, können weder das Dashboard noch das Backend effektiv genutzt werden. Zudem sollten Ausfälle während einer Katastrophe vermieden werden, da durchgeführte Triagen sonst nicht mit dem Backend synchronisiert werden können. Ein Rückfall auf die Stift- und Papier-Methode zur Durchführung von Triagen wäre die Folge.

Skalierbarkeit Die *KatApp* sollte sehr flexibel in ihrer Skalierbarkeit sein. Die meiste Zeit über befindet sich das System im Leerlauf, da Katastrophen eher selten auftreten. Bei kleineren Katastrophen wird das System kaum belastet. Das System wird also im Normalfall nur wenig bis gar nicht belastet. Tritt jedoch plötzlich eine Katastrophe größeren Ausmaßes auf, so kann es sein, dass mehrere hundert oder tausend Personen auf einmal auf die *KatApp* zugreifen. Das System muss in der Lage sein, mit derartigen, plötzlichen Lastspitzen umzugehen. Zudem sollte das System in Zukunft in der Lage sein, mit großen Lasten umzugehen. Es ist vorgesehen, die *KatApp* deutschlandweit zu betreiben.

Kosten Aus ökonomischen Gründen sollten die Kosten des Systems so gering wie möglich gehalten werden. Aufgrund der stark schwankenden Belastung sollten die Kosten gerade während eines Leerlaufs so niedrig wie möglich sein. Finden keine Katastrophen statt, sollten wenige bis keine Kosten für den Betrieb anfallen (mit Ausnahme der Verwaltungstätigkeiten von Mandanten).

Portabilität Das System sollte ein Mindestmaß an Portabilität aufweisen, um einen Vendor Lock-in zu verhindern. Zudem sollte das System auf einer eigenständig bereitgestellten Infrastruktur lauffähig sein (siehe Unterabschnitt 5.2).

Performance Die *KatApp* ist ein Echtzeit-System. Anfragen an das System sollen beinahe in Echtzeit (weniger als eine Sekunde pro Anfrage) bearbeitet werden, sodass Ersthelfer, Notärzte, die Einsatzleitung und die Leitstelle Entscheidungen möglichst schnell treffen können.

Wartbarkeit Der Code sollte strukturiert und dokumentiert sein. Außerdem sollte das System modular aufgebaut sein, sodass es einfach erweitert werden kann.

Datenschutz Die *KatApp* operiert hauptsächlich mit persönlichen und medizinischen Daten, weswegen Datenschutz-Konformität keinesfalls vernachlässigt werden sollte. Dabei sollen die technischen und organisatorischen Maßnahmen (TOMs) dokumentiert und erfüllt werden.

5.2 Umgebungen

Um die angestrebten Qualitätsziele zu erreichen, ist die Wahl einer geeigneten Ausführungsumgebung für das System von entscheidender Bedeutung. Die gewählte Umgebung sollte daher über eine hohe Verfügbarkeit, Skalierbarkeit und Kosteneffizienz sowie eine geringe Latenz verfügen.

Grundsätzlich existieren zahlreiche Möglichkeiten, die *KatApp* zu betreiben. Es können diverse Ansätze verfolgt werden, um die zuvor genannten Anforderungen zu erfüllen. Dabei ist jedoch zu beachten, dass eine ausgewogene Balance zwischen den spezifizierten Qualitätszielen gefunden wird. Diese schließen sich nicht gegenseitig aus, gehen jedoch oft zulasten anderer Aspekte. Eine hohe Verfügbarkeit des Systems korreliert beispielsweise oft mit höheren Kosten. Ähnliches gilt für die Skalierbarkeit. Eine höhere Portabilität wirkt sich in vielen Fällen negativ auf die Wartbarkeit aus. [17]

Aus der Perspektive einer medizinischen Anwendung, die hauptsächlich mit Patientendaten interagiert, erscheint das Betreiben eigener Infrastruktur als die naheliegende Lösung. Im medizinischen Sektor ist das Betreiben eigener Infrastruktur ein weit verbreitetes Phänomen. Eine Ursache dafür ist meist das hohe Alter medizinischer Applikationen. Oft kommen Technologien zum Einsatz, die gegenwärtig nicht mehr unterstützt werden oder nicht mehr weit verbreitet sind. Zudem hatten alternative Lösungen wie die Cloud zu damaliger Zeit oft mit Datenschutzkonformität zu kämpfen oder waren schlichtweg noch nicht verfügbar. [18] [19] [20]

Die Problematik des Datenschutzes stellt sich heutzutage für die meisten Cloud-Nutzer nicht mehr. Dies ist vor allem auf die Bemühungen großer Cloud-Anbieter wie AWS, Azure oder Google Cloud zurückzuführen, die den Datenschutz zunehmend in den Fokus rücken [21] [22]. Dies ist insbesondere im europäischen Raum der Fall, wo die Einhaltung von Datenschutzrichtlinien zunehmend verpflichtend ist [23]. Im Zeitalter der Cloud wird das Betreiben eigener Infrastruktur zunehmend seltener, und dies aus gutem Grund [24]. Das Mieten von Hardware-Infrastrukturen im Cloud-Bereich ist in der Regel sehr kostengünstig. Zudem ist die Einstiegshürde oft äußerst gering. Systeme können nach Bedarf skaliert werden, und dazu ist meist wenig bis kein Aufwand durch die Kunden der Cloud-Anbieter notwendig.

Im Falle des Katastrophenschutzes besteht jedoch eine weitere Herausforderung, die für Software-Systeme eher untypisch ist: mögliche Schäden an der Infrastruktur durch die Katastrophe. Bei stärkeren Erdbeben oder Stürmen zum Beispiel ist es durchaus möglich, dass das mobile Netz vor Ort ausfällt und somit keine Internetverbindung besteht. Dies erschwert die Nutzung von Cloud-Infrastruktur, aber auch eigener Infrastruktur. In diesem Zusammenhang können alternative Lösungen wie Satelliten-Internet (beispielsweise via *Starlink*[3]) in Erwägung gezogen werden.

Als alternative Möglichkeit zum Satelliten-Internet kann die *KatApp* auch direkt vor Ort, in einem lokalen Server und Netzwerk, betrieben werden. Für eine derartige Infrastruktur wäre es unter anderem denkbar, diese in einem Einsatzfahrzeug vorzuinstallieren. Die Herausforderung besteht darin, die *KatApp* möglichst portabel zu gestalten. Entwickelt man Anwendungen direkt für die Cloud, ergibt sich oft das Problem eines Vendor Lock-Ins. Bestimmte Cloud-Services erfordern meist sehr spezifische Technologien oder Abhängigkeiten, was dafür sorgt, dass Anwendungen nur in dieser Umgebung ausführbar sind.

Aufgrund der Effizienz, Simplizität sowie Kosten hat man sich entschieden, die *KatApp* primär in der Cloud zu betreiben. Man möchte jedoch weiterhin nicht auf die Möglichkeit verzichten, das System in einer lokalen Infrastruktur (zum Beispiel in einem Einsatzfahrzeug vor Ort) zu betreiben. Tatsächlich bleibt der lokale Betrieb der *KatApp* weiterhin eine Anforderung an das System.

[3] Starlink: `https://www.starlink.com/`

5.3 Lokale Umgebung

Die *KatApp* ist primär für den Betrieb in der Cloud konzipiert. Über diese Art der Bereitstellung können voraussichtlich 99% der Einsatzfälle abgedeckt werden. In dem einen Prozent der Fälle, wo jedoch die Netzwerkverbindung aufgrund der Katastrophe ausfällt, und somit keine Verbindung zur *KatApp* in der Cloud besteht, muss eine lokal bereitgestellte Version der *KatApp* verwendet werden.

Während sich die Architektur des Systems für unterschiedliche Bereitstellungsumgebungen anpassen lässt, ist dies nicht so einfach für die Anwendungslogik. Die *KatApp* ist ein mandantenfähiges System, sowohl in der Cloud, als auch lokal. Das hat zur Folge, dass eine lokale Bereitstellung nicht einfach trivial erfolgen kann. Genau wie bei der Cloud Variante, muss es mindestens einen zugelassenen Mandanten geben, über den die Zugriffe auf das System autorisiert werden können. Diese organisatorischen Aufgaben können unmöglich während einer Katastrophe vorgenommen werden. Daher muss im Falle einer lokalen Bereitstellung der *KatApp* mit einer bestehenden Datenbasis gearbeitet werden.

Um eine lokale Bereitstellung zu ermöglichen, die sofort einsatzbereit ist, müssen daher einige Dinge gegeben sein. Es muss ein Mandant existieren, über den Zugriffe autorisiert werden können. Es müssen Nutzer existieren, über die Zugriff auf das Dashboard erlangt werden kann, und es müssen Zugangstokens für Ersthelfer-Geräte existieren, die direkt am Einsatzort in den Geräten hinterlegt werden können.

Durch den Betrieb der *KatApp* in einem lokalen Netzwerk hat zudem die Leitstelle keinen Zugriff auf das lokale System und somit die Katastrophe. In diesem Fall wird die Leitstelle also vom System ausgeschlossen, und es muss zu traditionellen Methoden zurückgekehrt werden.

Für einen effizienten Einsatz müssen lokale Bereitstellungen also entsprechend präpariert werden. Im Optimalfall, könnte man die *KatApp* während einer lokalen Bereitstellung dazu bringen, anders zu agieren. Mechanismen wie Mandantschaften würden komplett wegfallen. Andere Prozesse wie Authentifizierung und Autorisierung wären nicht länger notwendig, da diese Sicherheit durch die Lokalität des Netzwerks gegeben wäre.

Die Unterscheidungen, welche in der Anwendungslogik basierend auf der aktuellen Bereitstellungsumgebung gemacht werden müssten, sind jedoch erheblich. Qualitätsziele wie Wartbarkeit und Performance würden stark unter solchen Anpassungen leiden. Im Grunde, würden diese Unterscheidungen zu einer komplett neuen Anwendung führen.

Aufgrund der durchaus geringen Wahrscheinlichkeit von Netzwerkausfällen in einer Katastrophe kann man daher vorerst mit der Möglichkeit einer präparierten lokalen Bereitstellung der *KatApp* leben.

5.4 Interaktionen & Schnittstellen

Die Ableitung der Architektur des *KatApp* Backends erfordert zunächst die Betrachtung des Systementwurfs. Dabei ist zu eruieren, welche Anwendungen involviert und welche Akteure beteiligt sind. Zudem muss eruiert werden, auf welchem Wege beteiligte Akteure und Anwendungen interagieren sowie welche Kommunikationskanäle bestehen.

Um die Akteure des Systems zu definieren, kann ein Blick auf die im Katastrophenschutz beteiligten Rollen (Unterabschnitt 2.1) geworfen werden. Es gibt drei Rollen, die direkten Kontakt mit dem System haben: die Ersthelfer, die Einsatzleitung und die Leitstelle. Diese Akteure interagieren jedoch mit unterschiedlichen Komponenten. Der Ersthelfer beispielsweise interagiert ausschließlich mit der mobilen App, während Einsatzleitung und Leitstelle ausschließlich mit dem Dashboard interagieren. Abbildung 5.1 veranschaulicht diese Interaktionen.

Darüber hinaus muss betrachtet werden, welche Anwendungen untereinander interagieren. Sowohl die App als auch das Dashboard interagieren mit dem Backend, jedoch nicht untereinander. Die Interaktion der App mit dem Backend ist unidirektional, während die Interaktion von Backend und Dashboard bidirektional ist. Das bedeutet, eine

Interaktion zwischen mobiler App und Backend kommt nur durch die App zustande, während eine Interaktion zwischen Backend und Dashboard durch beide der Seiten ausgelöst werden kann.

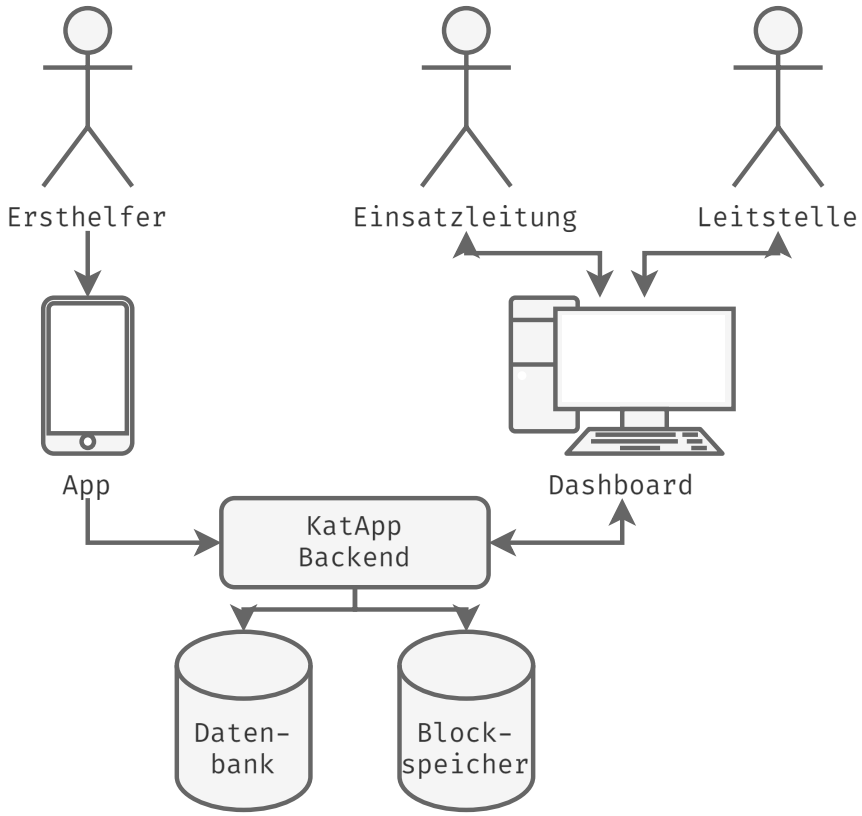

Abbildung 5.1: *KatApp* Systementwurf

Die *KatApp* nutzt zwei verschiedene Arten von Kommunikationskanälen: synchrone und asynchrone Kanäle. Bei der synchronen Kommunikation besteht eine unmittelbare Kausalität zwischen Anfrage und Antwort. Das bedeutet, dass jede Anfrage eine konkrete Antwort zur Folge hat. Bei der asynchronen Kommunikation hingegen ist diese Kausalität nicht gegeben. Eine Anfrage muss nicht direkt beantwortet werden, und eine „Antwort“ kann versendet werden, ohne dass eine Anfrage eingeht.

Die mobile App der *KatApp* kommuniziert ausschließlich über einen synchronen Kanal mit dem Backend. Zwischen Backend und Dashboard hingegen bestehen zwei Kommunikationskanäle, ein synchroner, und ein asynchroner. Operationen wie Authentifizierung oder die Abfrage initialer Daten werden über den synchronen Kanal durchgeführt, während die Aktualisierung von Echtzeit-Daten über den asynchronen Kanal stattfindet.

Für die Kommunikation mit App und Dashboard muss das *KatApp* Backend diverse Schnittstellen bereitstellen. Synchrone Kommunikation führt das Backend stets über REST durch. Dabei ist die API des Backends versioniert. Das heißt, für jeden REST Endpunkt kann es mehrere Versionen geben. Durch diesen Mechanismus können sowohl Verbesserungen an Endpunkten durchgeführt als auch die Kompatibilität mit älteren Versionen der App oder des Dashboards aufrechterhalten werden. Das Dashboard der *KatApp* nutzt ebenfalls REST für das initiale Laden von Katastrophendaten sowie zum Beispiel die Authentifizierung.

Die Echtzeit-Daten, die das Dashboard während einer Katastrophe vom Backend erhält, werden stets asynchron kommuniziert. Für den asynchronen Kommunikationsweg wird das *WebSocket* Protokoll verwendet. Mit der Anmeldung eines Nutzers beim Dashboard wird eine *WebSocket* Verbindung mit dem Backend aufgebaut, welche besteht, bis sich der Nutzer aus dem Dashboard abmeldet. Das Backend benachrichtigt verbundene Klienten über neue Triagen, Patienten und Katastrophen, wodurch das Dashboard nicht

zyklisch nach neuen Information fragen muss. Die beschriebene Vorgehensweise führt zu einer Reduktion der Belastung sowohl des Backends als auch des Dashboards.

Aus dem Kernentwurf der *KatApp* ergeben sich zudem die externen Dienste, welche zum Betrieb des Systems benötigt werden. Das ist zum einen die Datenbank, zum anderen der Blockspeicher. Der Blockspeicher wird zur Persistierung unstrukturierter, binärer Daten verwendet (zum Beispiel für Fotos der Patienten und ihren Verletzungen).

Da die *KatApp* sowohl in der Cloud als auch lokal (auf eigener Infrastruktur) betrieben werden soll, muss der Entwurf aus Abbildung 5.1 auch in beiden diesen Umgebungen abgebildet werden. Dadurch entstehen zusätzliche Herausforderungen. Cloud-Dienste wie Datenbanken oder Blockspeicher sind zumeist auf den Betrieb bei ihrem jeweiligen Cloud-Anbieter beschränkt. Dies impliziert, dass für den lokalen Betrieb andere Dienste für Datenbank und Blockspeicher verwendet werden müssen, als es in der Cloud der Fall ist. Die *KatApp* muss also trotz der Verwendung unterschiedlicher Dienste in unterschiedlichen Umgebungen dasselbe Verhalten abbilden. Zudem muss das Backend unabhängig von seiner Umgebung dieselben Schnittstellen bereitstellen, damit sowohl App als auch Dashboard unabhängig von der Umgebung auf einheitliche Weise mit dem Backend interagieren können.

5.5 Cloud

Dass die Cloud ein Medium der Zukunft ist, erkennt man an den Jahr für Jahr steigenden Umsätzen in diesem Sektor [25]. Im Vergleich zum Eigenbetrieb von Serverinfrastruktur bietet die Cloud erhebliche Vorteile. Hardware muss nicht selbstständig gewartet oder aktualisiert werden, es braucht kein separates Personal für die Betreuung, neueste Sicherheitsstandards sind bereits gegeben oder werden umgehend automatisch bereitgestellt. Darüber hinaus kann man die benötigte Hardware praktisch in Sekundenschnelle an die aktuellen Anforderungen anpassen, wenn ein System hoch- oder herunterskaliert werden muss. [17]

Trotz der durch die Cloud gegebene Flexibilität ist die Wahl der Infrastruktur für ein System nicht unbedingt einfacher. Cloud-Anbieter stellen einen großen Umfang an Services bereit. Welche dieser Services für eine Anwendung passend sind, lässt sich aus den Anforderungen der Anwendung ableiten. Auch hier ist zu beachten, dass oftmals ein Kompromiss zwischen ausgewählten Anforderungen eingegangen werden muss. Die Nutzung der Cloud zur Bereitstellung hochverfügbarer oder hochskalierbarer Systeme ist grundsätzlich eine vielversprechende Strategie. Allerdings ist zu berücksichtigen, dass die damit verbundenen Kosten signifikant ansteigen. [17]

Aus den Qualitätsanforderungen der *KatApp* lässt sich auf ein hochverfügbares, hoch skalierbares, aber dennoch kosteneffizientes System schließen. Die *KatApp* scheint also ein typischer Kandidat für einen Kompromiss aus Verfügbarkeit, Skalierbarkeit und Kosten zu sein. Im Gegensatz zu den meisten Cloud-Anwendungen hat die *KatApp* jedoch eine Besonderheit, die extrem schwankende Belastung. Katastrophen treten typischerweise nicht allzu häufig auf, was bedeutet, die *KatApp* befindet sich die meiste Zeit über im Leerlauf. Natürlich entsteht eine minimale Belastung durch Verwaltungsaufgaben der Mandanten, ist aber in den meisten Fällen zu vernachlässigen. Folglich bleibt das System die meiste Zeit über ungenutzt. Setzt man nun eine hochverfügbare Infrastruktur für den Betrieb der *KatApp* auf, muss man trotz geringer Nutzung mit außerordentlich hohen Kosten rechnen. Im Idealfall nutzt man also ein System, welches im Leerlauf wenige bis keine Kosten verursacht, im Ernstfall aber hochverfügbar und -skalierbar ist.

Die Bewältigung von Herausforderungen führt in der Regel zu neuen Lösungsansätzen. So hat sich in den vergangenen Jahren das Modell *Serverless* etabliert [26]. Ein modernes Modell, welches die Balance zwischen Kosten, hoher Verfügbarkeit sowie Skalierbarkeit in den Griff bekommen zu scheint. Serverlose Modelle sind eine attraktive Lösung für diverse Anwendungsfälle, so auch für die *KatApp*.

Serverlose Modelle überzeugen insbesondere durch ihre Simplizität. Sie abstrahieren den Server von der Applikation und dem Nutzer und vereinfachen somit die Bereitstellung von Anwendungen [27]. Entwickler sind nicht länger gezwungen, sich Gedanken über Infrastruktur und Hardware zu machen, da diese automatisch vom Cloud-Anbieter verwaltet werden. Serverlose Modelle werden in der Regel auf Basis des Pay-Per-Use-Prinzips abgerechnet. Dies bedeutet, dass der Kunde nicht dauerhaft für den Betrieb der Hardware zahlt, sondern lediglich für deren Nutzung. Im Gegensatz zu „regulärer“ Infrastruktur wie virtuellen Maschinen, wird ein serverloses System bei Nicht-Benutzung automatisch abgeschaltet. Dies hat den Vorteil, dass nur geringe bis keine Kosten für den Kunden anfallen, sollte die Hardware gerade nicht genutzt werden (minimale Kosten werden potenziell durch Speicher verursacht).

Ein serverloses Modell bietet somit die perfekte Grundlage für die *KatApp*. Im Leerlauf, wenn das System gerade nicht verwendet wird, fallen keine Kosten für den Betrieb an, und unter Last wird das System je nach Bedarf vom Cloud-Anbieter hochskaliert. Eine stabile Balance zwischen Verfügbarkeit, Skalierbarkeit und Kosteneffizienz ist die Folge.

5.6 Cloud Architektur

Dieses Unterkapitel widmet sich der Cloud Architektur der *KatApp*. Die *KatApp* wird für AWS entwickelt und optimiert und basiert auf einer serverlosen, *Function as a Service* (FaaS) Architektur. AWS bietet ein breites Spektrum an serverlosen Services und stellt mit *Lambda*[4] und *DynamoDB*[5] den Kern der *KatApp*. Darüber hinaus werden die Frontend-Komponenten (Dashboard, Mandantenkonsole, Adminkonsole) über einen *S3*[6] Blockspeicher zur Verfügung gestellt. Die gesamte Architektur ist auf preisliche Flexibilität ausgelegt. Wird das System nicht verwendet, fallen minimale bis keine Kosten an.

Abbildung 5.2 bildet die Cloud-Architektur des Backends ab. Diese Architektur wird in vier verschiedenen Umgebungen bereitgestellt:

Prod *Prod* repräsentiert die Produktivumgebung der *KatApp*. Alle auf der Produktivumgebung bereitgestellten Dienste werden aktiv durch Kunden genutzt. Die in dieser Umgebung hinterlegten Daten gehören zu echten Patienten und Katastrophen, welche in Vergangenheit aufgetreten sind. Neue Funktionen oder Fehlerbehebungen sollten zuvor auf einer Entwicklungs- oder Testumgebung getestet werden, bevor sie auf der Produktivumgebung ausgerollt werden.

Dev Die *Dev* Umgebung ist die primäre Umgebung für Entwicklungsarbeiten und Tests. Auf dieser Umgebung können neue Funktionen getestet oder Fehlerbehebungen durchgeführt werden. Zudem können Übungen für den Katastrophenschutz auf dieser Umgebung ausgeführt werden. Ist die Entwicklungsumgebung ausreichend auf Fehlverhalten getestet, so kann derselbe Stand auf die Produktivumgebung ausgerollt werden.

Test Neben Produktiv- und Entwicklungsumgebung werden zudem zwei weitere Testumgebungen *Test1* und *Test2* bereitgestellt. Diese Umgebungen dienen Entwicklern für flüchtige Tests oder Anpassungen des Bereitstellungsprozesses.

[4] Lambda: `https://aws.amazon.com/de/lambda/`
[5] DynamoDB: `https://aws.amazon.com/de/dynamodb/`
[6] S3: `https://aws.amazon.com/de/s3/`

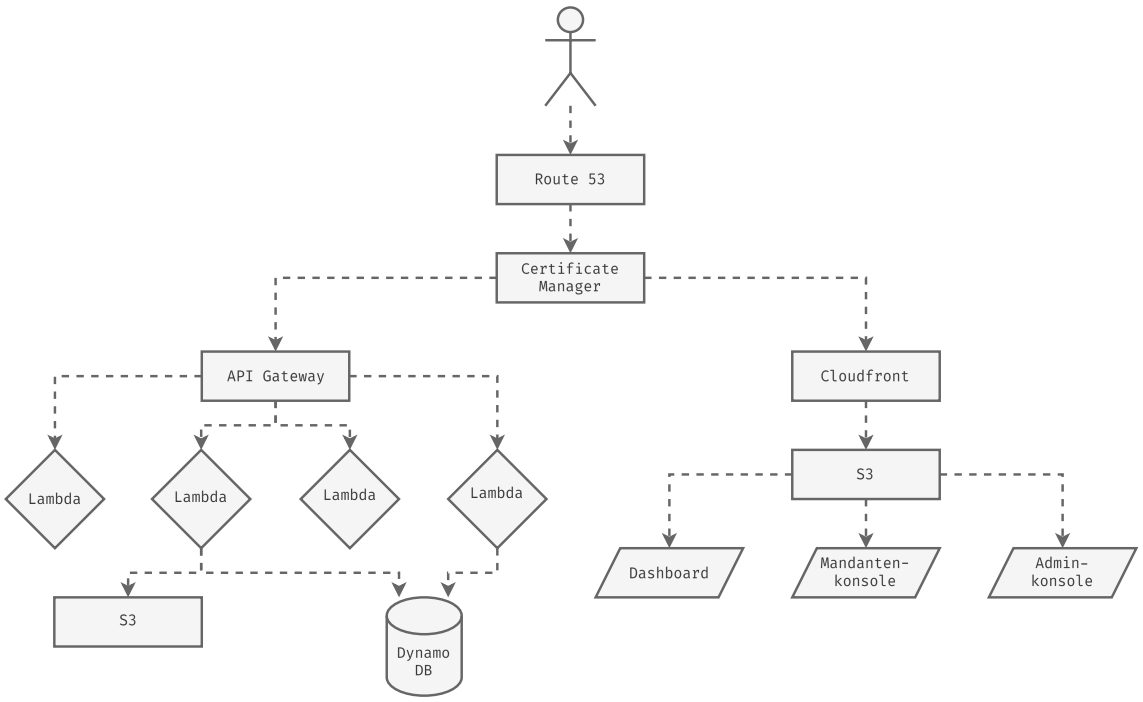

Abbildung 5.2: *KatApp* Architektur (Cloud)

Das Produktivsystem der *KatApp* ist über die Domäne *katapp.org* erreichbar, welche über Amazons Service *Route 53*[7] verwaltet wird. Analog zur Produktivumgebung werden weitere Domänen für Entwicklungs- und Testumgebungen bereitgestellt. Über den *Certificate Manager*[8] von AWS werden die Zertifikate für eine sichere Kommunikation via HTTPS bereitgestellt. Somit sind alle vier Umgebungen über einen sicheren Kommunikationskanal erreichbar.

Die Schnittstellen des *KatApp* Backends werden über Amazons *API Gateway*[9] bereitgestellt. Dieser fungiert als Reverse-Proxy und handhabt zudem eingehende *WebSocket* Verbindungen. Die eigentlichen REST Endpunkte der *KatApp* sind nicht direkt, sondern ausschließlich über den Gateway erreichbar. Der *API Gateway* übernimmt zudem weitere Aufgaben wie die Drosselung von Anfragen oder Lastenausgleich.

Mit Hilfe von AWS *Lambda* und FaaS als Entwurfsmuster lassen sich Microservice-ähnliche Architekturen entwerfen. Logikeinheiten lassen sich so vereinzelt als *Funktionen* ausführen, die nur gestartet und ausgeführt werden, wenn Anfragen eingehen. Gehen keine Anfragen ein, wechseln die Funktionen vorübergehend in den Leerlauf, bis sie komplett heruntergefahren werden. Bis zu einem Neustart (hervorgerufen durch eingehende Anfragen) verursachen die Funktionen keine Kosten. Gehen viele Anfragen ein, werden mehrere Instanzen derselben Funktion bereitgestellt, sodass Anfragen parallel bearbeitet werden können. Aufgrund der dadurch entstehenden Flexibilität können Lastspitzen effizient abgefangen werden. Jeder einzelne REST Endpunkt des *KatApp* Backends wird als individuelle *Lambda* Funktion bereitgestellt. Durch diese extrem feingranulare Aufteilung kann AWS vereinzelte *Lambda* Funktionen basierend auf ihrer individuellen Last skalieren.

Für die Haltung von strukturierten Daten greifen sämtliche *Lambda* Funktionen auf eine *DynamoDB* zu. Aufgrund der vorwiegend objektorientierten Daten, die während einer Katastrophe anfallen (Patienten, Triagen, etc.), wird eine dokumentenbasierte Datenbank gewählt. *DynamoDB* überzeugt in zwei Aspekten, sie ist eine dokumentenbasierte Key-Value Datenbank, und unterstützt serverlose Bereitstellung. Infolgedessen skaliert *DynamoDB* automatisch basierend auf den eingehenden Anfragen. Unstrukturierte bzw. binäre

[7]Route 53: `https://aws.amazon.com/de/route53/`
[8]Certificate Manager: `https://aws.amazon.com/de/certificate-manager/`
[9]API Gateway: `https://aws.amazon.com/de/api-gateway/`

Daten werden in einem *S3* Speicher hinterlegt. Dieser dient der Ablage von Bildern, die mit der mobilen App von Patienten gemacht werden können.

Dashboard, Mandanten- und Adminkonsole sind Web-Applikationen, welche ebenfalls in einem separaten *S3* Speicher abgelegt werden. Bereitgestellt werden die Web-Applikationen über AWS *Cloudfront*[10].

Durch die Flexibilität, welche mit FaaS Architekturen einhergeht, erfüllt die Architektur die Qualitätsziele für Skalierbarkeit und Kosten der *KatApp*. Mit AWS als Cloud-Anbieter gilt ebenso das Qualitätsziel Verfügbarkeit als erfüllt, da eine jeweilige Verfügbarkeit von mindestens 99.95 % für die individuellen Services garantiert wird [28] [29] [30]. Diese Verfügbarkeitsgarantie wird als ausreichend betrachtet.

5.7 Lokale Architektur

Im Gegensatz zur Bereitstellung über die Cloud, sind Qualitätsziele wie Skalierbarkeit, Kosten und Performance von nicht so hoher Bedeutung für die lokale Bereitstellung. Während das System in der Cloud mit mehreren Katastrophen parallel umgehen muss, ist dies bei einem lokalen Betrieb nicht der Fall. Wird das System in einem Einsatzfahrzeug vor Ort mit Hilfe eines lokalen Netzwerks betrieben, so muss das System mit maximal einer Katastrophe zur gleichen Zeit umgehen. Die Belastungsanforderungen sind also wesentlich geringer.

Dabei erfolgt die lokale Bereitstellung der *KatApp* über Container-Services in einer lokalen Infrastruktur, die in einem der Einsatzfahrzeuge betrieben wird. Dadurch ändert sich auch die Architektur der *KatApp*. Während jeder einzelne REST Endpunkt beim Betrieb in der Cloud aufgrund von AWS *Lambda* als separater Container ausgeführt wird, ist die *KatApp* im lokalen Betrieb eine monolithische Applikation. Diese Eigenschaft erlaubt eine einfachere Bereitstellung und verbesserte Testbarkeit für Entwickler. Zudem wäre eine Bereitstellung jedes einzelnen REST Endpunkts als Container eine unnötige Belastung der Ressourcen und bringt keine messbare Verbesserung der Performance mit sich.

Im Falle einer lokalen Bereitstellung ändern sich zudem die Abhängigkeiten der *KatApp*. *DynamoDB* und *S3* Speicher können nicht länger verwendet werden, da diese Services nicht außerhalb der AWS Cloud existieren. Anders als bei einer Bereitstellung über die Cloud, verwendet die *KatApp* in einem lokalen Szenario *MongoDB*[11] als Speicher, sowohl für strukturierte als auch für binäre Daten.

Durch die Anpassung der Abhängigkeiten muss eine Abstraktionsschicht in die *KatApp* integriert werden, welche den Betrieb mit zwei verschiedenen Datenbanken und Blockspeichern ermöglicht, je nachdem in welcher Bereitstellungsumgebung man sich befindet. Der initiale Aufwand für die Abstraktion der nötigen Kernkomponenten ist zwar hoch, die dadurch entstehenden Vorteile überwiegen diese Aufwandskosten jedoch. Cloud-Architekturen sind meist nur schwer auf lokaler Hardware auszuführen. Oftmals ist dies nur mit aufwendigen Attrappen der originalen Cloud-Services zu erreichen, welche an sich selten in vollem Funktionsumfang verfügbar sind. Und eine Eigenentwicklung dieser Attrappen wäre aufwendig. Zudem sind Fehleranalysen teuer, da ein Debugging solcher Architekturen (ohne die nötigen Attrappen, welche in sich ebenfalls fehlerfrei sein müssen) praktisch unmöglich ist. Folglich ist die Möglichkeit einer lokalen Bereitstellung nicht nur wichtig für die Portabilität des Systems, sondern macht das System zudem entwicklerfreundlich, was zeitliche und monetäre Ressourcen schont.

[10] Cloudfront: `https://aws.amazon.com/de/cloudfront/`
[11] MongoDB: `https://www.mongodb.com/`

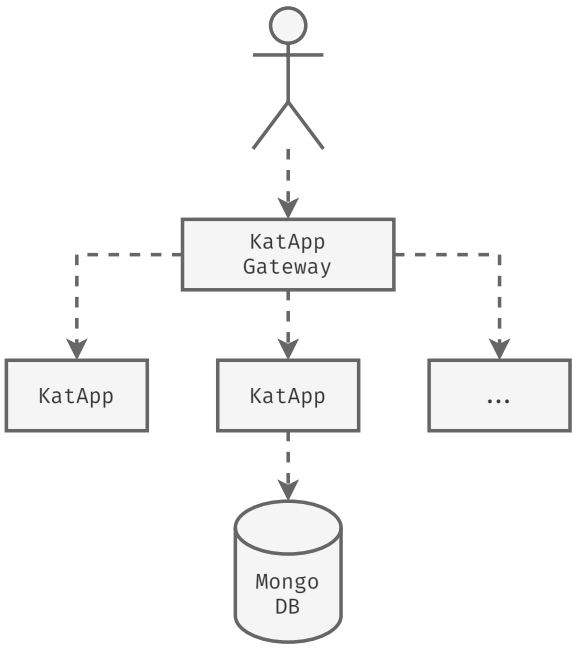

Abbildung 5.3: *KatApp* Architektur (Lokal)

Abbildung 5.3 zeigt eine lokale Bereitstellung der *KatApp* Architektur. Analog zur Cloud-Architektur wird ein Gateway vor die bereitgestellten *KatApp* Instanzen geschaltet.

Um eine Abstraktion der lokalen Dienste und der Cloud Dienste einfacher zu gestalten, wird die lokale Architektur soweit möglich an die Cloud Architektur angeglichen. Dies dient vor allem dem Umgang mit *WebSocket* Verbindungen. In der Cloud verwaltet der *API Gateway* alle eingehenden *WebSocket* Verbindungen. Der Grund dafür ist, dass *Lambda* Funktionen im Leerlauf heruntergefahren bzw. unter Last mehrfach bereitgestellt werden können. Da eine *WebSocket* Verbindung immer nur mit einer Instanz eingegangen werden kann, können diese nicht horizontal skaliert werden. Durch den Einschub des *API Gateway* jedoch, können *WebSocket* Verbindungen unabhängig vom Verhalten einzelner *Lambda* Funktionen offen gehalten werden. *Lambda* Funktionen wiederum können über den *API Gateway* mit den verbundenen Klienten kommunizieren. Dadurch werden *Lambda* Funktionen zustandslos und können horizontal skaliert werden.

Durch die Einführung eines Gateways in lokalen Umgebungen kann dieses Verhalten auch lokal repliziert werden. Auch als monolithisches System in lokalen Umgebungen ist das *KatApp* Backend aufgrund der *WebSocket* Verbindungen kein zustandsloses System. Durch die Verwaltung der *WebSocket* Verbindungen, welche immer direkt mit einer *KatApp* Instanz bestehen, wird die *KatApp* zustandsbehaftet. Analog zur Cloud Architektur wird diese Problematik mit Hilfe eines Gateways umgangen. So wird die *KatApp* auch in lokalen Szenarien horizontal skalierbar.

Neben der horizontalen Skalierbarkeit hat dieses Vorgehen auch weitere Vorteile für die Entwickler. Durch die analoge Verhaltensweise lässt sich die *WebSocket* Kommunikation auf Code-Ebene einfacher abstrahieren. über den Gateway kann jede horizontal skalierte *KatApp* Instanz Daten über eine Verbindung versenden oder empfangen. Das rührt daher, dass Gateway und *KatApp* auch im Falle von *WebSocket* Verbindungen synchron über REST interagieren. Verbindet sich der Klient zum *KatApp Gateway*, ruft der Gateway einen entsprechenden REST Endpunkt der *KatApp* auf. Selbiges Verfahren wird für das Beenden einer Verbindung verwendet. Der Gateway weist dabei jedem Klienten eine eindeutige Kennung zu, die der *KatApp* mitgeteilt wird. Über diese Kennung kann die *KatApp* den Gateway dazu anhalten, eine Nachricht über *WebSocket* an einen bestimmten Klienten zu versenden.

Aufgrund der proprietären Kommunikation zwischen Gateway und *KatApp* ist der *KatApp Gateway* eine Eigenentwicklung des *KatApp* Teams.

5.8 Technologien

Sowohl das *KatApp* Backend als auch der *KatApp Gateway* ist in der Programmiersprache *Go*[12] geschrieben. Dies hat vor allem zwei Gründe:

- *Go* ist eine Programmiersprache, die sich vor allem im Umfeld Web-Services (*Docker*[13], *Kubernetes*[14], *Traefik*[15], *Hugo*[16]) bewährt hat. Zudem wird die Sprache von AWS *Lambda* unterstützt, was ein essenzielles Kriterium für die Cloud-Architektur ist.
- *Go* ist eine kompilierte, und daher sehr performante Sprache, die aber aufgrund des eingebauten Garbage Collectors trotzdem einfach in ihrer Handhabung ist.

Im Vergleich zu anderen kompilierten Programmiersprachen punktet *Go* mit seiner umfangreichen Standardbibliothek im Bereich Webentwicklung. Dies ist einer der zentralen Gründe, warum *Go* in Projekten wie *Kubernetes* eingesetzt wird [31]. Außerdem, im Gegensatz zu anderen kompilierten Sprachen, benötigt *Go* keine separate Laufzeitumgebung, um ausgeführt zu werden. Das erleichtert eine Bereitstellung und verbessert zudem die Laufzeit-Performance.

Neben der Programmiersprache *Go* ist *Docker*[17] ein unverzichtbares Werkzeug im Umgang mit der *KatApp*. Mit Hilfe von *Docker* lässt sich die *KatApp* nicht nur leicht verteilen, sondern auch effizient bereitstellen. Neben der produktiven Bereitstellung ist *Docker* zudem essenziell für Entwickler. Dadurch können Abhängigkeiten wie Datenbanken oder Blockspeicher einfach bereitgestellt werden.

[12] Go: `https://go.dev/`
[13] Docker GitHub: `https://github.com/docker/cli`
[14] Kubernetes GitHub: `https://github.com/kubernetes/kubernetes`
[15] Traefik GitHub: `https://github.com/traefik/traefik`
[16] Hugo GitHub: `https://github.com/gohugoio/hugo`
[17] Docker: `https://www.docker.com/`

6 Implementierung

Das vorliegende Kapitel widmet sich der Implementierung zentraler Abläufe der *KatApp*. Ziel ist es, ein Verständnis der Systemabläufe zu gewinnen. Darüber hinaus werden die wichtigsten Elemente des Datenmodells dargelegt. Auf diese Weise wird ein Einblick in die Abbildung der Fachlichkeit aus der realen Welt im virtuellen System ermöglicht.

6.1 Datenmodell

Die *KatApp* arbeitet, unabhängig von ihrer Laufzeitumgebung, jeweils mit einer NoSQL Datenbank, weswegen auch das Datenmodell keinem typischen relationalen Modell entspricht. Sowohl *MongoDB* als auch *DynamoDB* sind dokumentenbasierte Datenbanken, daher ist das Datenmodell vorwiegend objektorientiert, mit relationalen Aspekten, wo sie erforderlich sind.

Betrachtet man die fachlichen Abläufe einer Katastrophe und die involvierten Entitäten, lässt sich daraus bereits auf einen Großteil des Datenmodells schließen. Im Folgenden wird auf die identifizierten Entitäten eingegangen. Des Weiteren wird dargelegt, welche Überlegungen den getroffenen Designentscheidungen zugrunde liegen.

Mandant Das Datenmodell des Mandanten bildet alle Eigenschaften einer Organisation ab, die sich für die *KatApp* registriert hat. Abbildung 6.1 beschreibt das Datenmodell eines Mandanten.

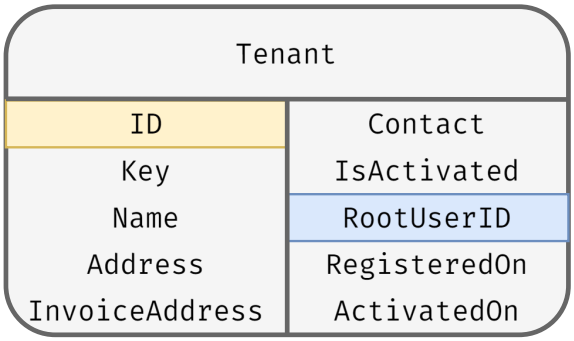

Abbildung 6.1: Datenmodell: Mandant

Wie beinahe jedes Datenobjekt der *KatApp* erhält ein Mandant eine eindeutige ID, über die dieser identifiziert werden kann. Neben der ID wird zusätzlich ein Schlüssel (*Key*) vergeben. Dieser Schlüssel kann von dem Mandanten selbst bestimmt werden. Der Schlüssel des Mandanten muss ebenfalls einzigartig sein. Er dient der Identifizierung des Mandanten für Anwender, damit diese nicht mit einer generischen ID arbeiten müssen. Neben den Adressdaten, die für die Ausstellung von Rechnungen erfasst werden, hat jeder Mandant einen Administrator, welcher hier per ID (*RootUserID*) referenziert wird. Der Administrator wird stets mit dem Mandanten selbst registriert, weswegen dieser zum Zeitpunkt der Registrierung bereits bekannt ist. Aufgrund des Authentifizierungsverfahrens der *KatApp* kann ein Mandant zwar registriert, aber noch nicht aktiviert sein. Ist ein Mandant nicht aktiviert, ist dieser handlungsunfähig. Mandanten können ausschließlich durch den Systemadministrator aktiviert werden. Abgebildet wird dieses Verhalten durch die Eigenschaft *IsActivated*.

Der Systemadministrator selbst ist wiederum ein Mandant. Im Gegensatz zur Registrierung für reguläre Mandanten muss der Systemadministrator nicht explizit aktiviert werden. Die Registrierung eines Systemadministrators ist in der *KatApp* nur einmalig möglich, und auch nur, wenn es noch keine anderen Mandanten oder Nutzer gibt. Die Registrierung eines Systemadministrators bildet somit den initialen Schritt für die Inbetriebnahme der *KatApp* ab.

Nutzer Das Datenmodell des Nutzers ist der Kern der Authentifizierung. Die meisten Endpunkte der *KatApp* sind geschützt und somit nur von authentifizierten Nutzern aufrufbar. Ein Nutzer kann hierbei zum Beispiel ein Mitglied einer Leitstelle, einer Einsatzleitung, oder auch ein Ersthelfer sein. Da Ersthelfer präparierte Geräte nutzen, die vorher durch ihre Organisation freigeschaltet werden, gibt es auch sogenannte Geräte-Nutzer oder anonyme Nutzer. Diesen anonymen Nutzern kann zwar ein Name (*FirstName* und *LastName)* zugeordnet werden, verpflichtend ist dies jedoch nicht. Gekennzeichnet werden diese Geräte-Nutzer über die Eigenschaft *IsDeviceUser*. Des Weiteren besteht für Mandanten die Möglichkeit, Informationen über ein authentifiziertes Gerät zu hinterlegen, um eine nachträgliche Identifizierung der Geräte zu erleichtern. Diese Informationen können über die Eigenschaften *DeviceImei*, *DeviceName* und *DeviceOs* hinterlegt werden.

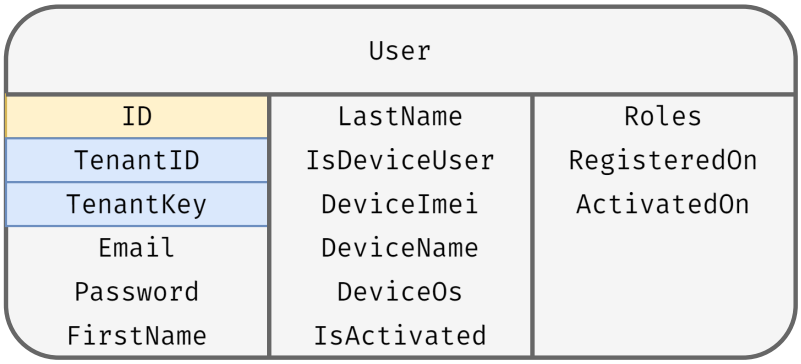

Abbildung 6.2: Datenmodell: Nutzer

Aus Abbildung 6.2 ist außerdem das Rollenmodell der Nutzer ersichtlich. Ein Nutzer kann keine, eine, oder mehrere Rollen (*Roles*) innehaben. Im Datenmodell werden diese Rollen dabei über eine ID referenziert. So können Rollen durch eine Organisation aktualisiert werden, ohne dass die Einträge der Nutzer aktualisiert werden müssen. Dies hat zudem den Vorteil, dass diese Änderungen sofort im System reflektiert werden.

Ebenso ist ersichtlich, dass der Mandant des Nutzers im Datenmodell referenziert wird (*TenantID*). Diese Vorgehensweise wird auf beinahe jedes Modell angewandt, um nachvollziehen zu können, welchem Mandanten eine Ressource angehört.

Analog zu den Mandanten muss auch ein Nutzer explizit vom Mandantenadministrator aktiviert werden, vorher ist dieser handlungsunfähig. Dieses Verhalten ist ebenso Folge der Authentifizierungshierarchie (Unterabschnitt 2.8). Abgebildet wird dieses Verhalten über die Eigenschaft *IsActivated*.

Rolle Das Datenmodell einer Rolle bildet den Kern des Autorisierungsverfahrens ab. Wie in Unterabschnitt 2.9 beschrieben, kann eine Rolle stets eine Menge von Rechten beherbergen (*Rights*). Neben dieser essenziellen Information werden einer Rolle noch weitere Metadaten wie Name (*Name*) oder Beschreibung (*Description*) hinzugefügt, um die Verwaltung von Rollen für die Mandanten übersichtlicher zu machen. Abgebildet ist das Datenmodell in Abbildung 6.3.

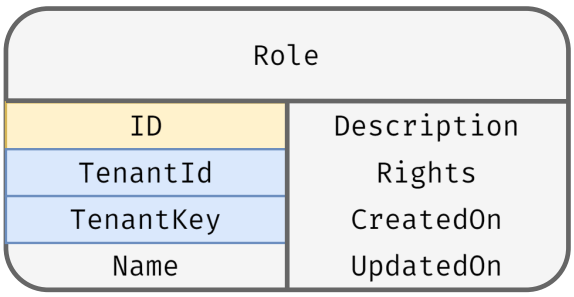

Abbildung 6.3: Datenmodell: Rolle

Katastrophe In der aktuellen Version der *KatApp* sind Ressourcen wie Katastrophen, Patienten und Triagen ebenfalls direkt an Mandanten gebunden (*TenantID*). Ein Mandant besitzt diese Ressourcen und andere Mandanten können nicht auf diese zugreifen.

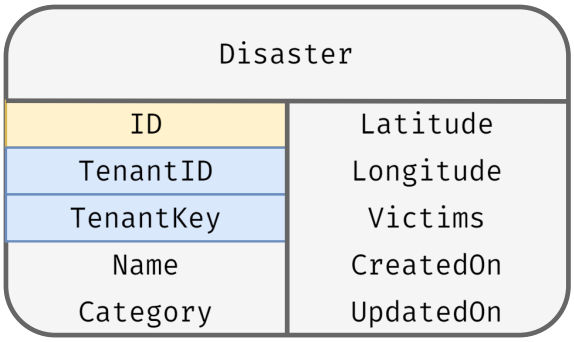

Abbildung 6.4: Datenmodell: Katastrophe

Zu den wichtigsten Eigenschaften einer Katastrophe (Abbildung 6.4) gehören die Koordinaten (*Latitude* und *Longitude*) und das letzte Aktualisierungsdatum (*UpdatedOn*). Sowohl die Koordinaten einer Katastrophe als auch ihr letztes Aktualisierungsdatum sind essenziell für die Zuordnung von Triagen zu Katastrophen. Anhand des letzten Aktualisierungsdatums wird bestimmt, ob eine Katastrophe noch aktiv oder bereits beendet ist. Eine Katastrophe gilt als beendet, sollte innerhalb von 24 Stunden keine Aktualisierung erfolgen.

Für eine bessere Identifizierbarkeit wird einer Katastrophe automatisiert ein Name (*Name*) zugewiesen. Darüber hinaus wird die aktuelle Anzahl an Patienten (*Victims*), welche in der Katastrophe triagiert wurden, hinterlegt. Diese Eigenschaft dient der Darstellung von Katastrophen im Dashboard.

Patient Das Datenmodell des Patienten (Abbildung 6.5) enthält alle nötigen Eigenschaften, die während einer Triage für einen bestimmten Patienten erfasst werden. Dazu gehören unter anderem Name (*Name*), Geschlecht (*Gender*) und Alter (*Age*) sowie evtl. vorliegende Vorerkrankungen (*PEHC*). All diese Eigenschaften sind jedoch optional, da ein Ersthelfer evtl. keine Kenntnis darüber hat oder keine falschen Annahmen treffen will.

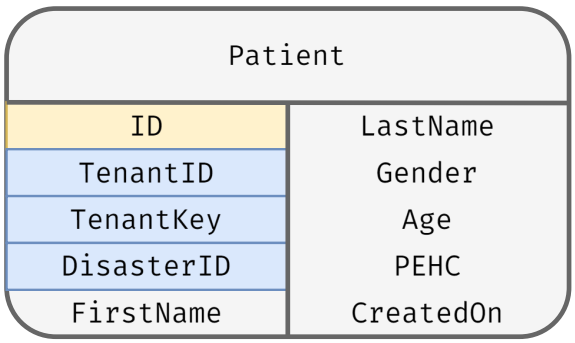

Abbildung 6.5: Datenmodell: Patient

Triage Das Datenmodell einer Triage (Abbildung 6.6) umfasst alle Daten, die während der Triage eines Patienten erfasst werden. Jede Triage wird zudem genau einem Patienten (*PatientID*) und genau einer Katastrophe (*DisasterID*) zugeordnet. Optional kann eine Triage zudem Informationen über den Nutzer bzw. Ersthelfer enthalten (*UserInfo*), falls diese Informationen beim Präparieren der Ersthelfer-Geräte angegeben werden.

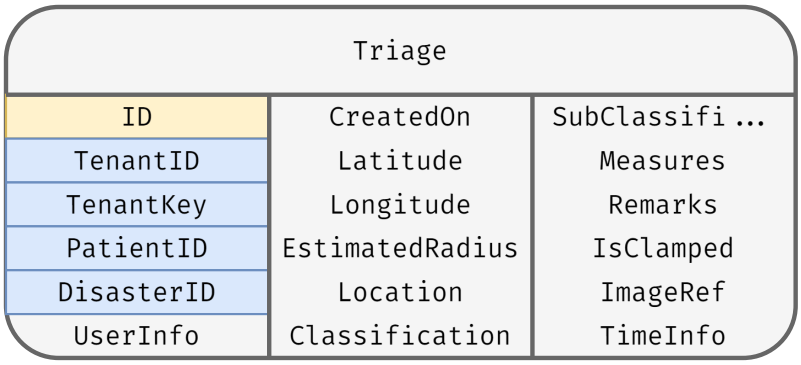

Abbildung 6.6: Datenmodell: Triage

Neben den eigentlichen Koordinaten der Triage (*Latitude* und *Longitude*), kann der geschätzte Radius (*EstimatedRadius*) angegeben werden, der die Ungenauigkeit der Standorterfassung beschreibt. Die Mehrheit der auf dem Markt erhältlichen Smartphones verfügt über eine Funktion zur Schätzung der Genauigkeit der Standorterfassung.

6.2 Authentifizierung & Autorisierung

Der Authentifizierungsmechanismus der *KatApp* basiert auf *JSON Web Tokens* (JWTs). Grundsätzlich lassen sich in JWTs eine beliebige Menge an Attributen speichern. Als Beispiel sei hier die ID des Nutzers genannt. Darüber hinaus besteht die Möglichkeit, weitere Eigenschaften wie Zugangsberechtigungen oder andere Informationen im Token zu hinterlegen. Dies kann zu einer Reduktion von Datenbankabfragen im Verlauf der Authentifizierung führen.

Die *KatApp* speichert ausschließlich die Nutzer-ID im Zugangstoken. Aufgrund dieser Entscheidung müssen bei jeder authentifizierten und autorisierten Anfrage mindestens zwei Datenbankabfragen gemacht werden. Zum einen muss der Nutzer selbst über die gegebene ID, zum anderen die Rolle des Nutzers und somit seine Berechtigungen abgefragt werden. Der hier beschriebene Ansatz weist nicht nur aus Sicht der Laufzeit eine geringere Effizienz auf, sondern führt zudem durch die zusätzlichen Datenbankabfragen zu einer Steigerung der Kosten.

Man hat sich für diesen Ansatz aus folgendem Grund entschieden: hinterlegt man alle Berechtigungen eines Nutzers in seinem Token, so hat eine Änderung der Nutzerberechtigungen im System keine Auswirkung auf diesen Nutzer, solange er weiterhin dieses

Token verwendet. Problematisch ist dieser Ansatz für die meisten Systeme nicht. Die *KatApp* hingegen macht Gebrauch von den bereits erwähnten Geräte-Tokens, welche Nutzerunabhängig in den Smartphones, welche für die Triagierung von Patienten verwendet werden, im Voraus hinterlegt werden. Um den Verwaltungsaufwand der Mandanten so gering wie möglich zu halten, ist die Gültigkeitsdauer dieser Tokens sehr hoch gewählt. Die Gültigkeit dieser Tokens kann sich von einem Jahr bis in die Unendlichkeit strecken. Durch den in der *KatApp* gewählten Ansatz kann ein Mandant nun auch nach Generierung der Tokens die Berechtigungen einzelner Geräte ändern, ohne dass er die Tokens neu generieren muss.

6.3 Erstellung einer Triage

Dieses Unterkapitel widmet sich dem Ablauf einer Triage-Erstellung. Dieser Prozess enthält diverse Schritte. Der gesamte Ablauf von der Durchführung der Triage durch den Ersthelfer bis zur Persistierung in der Datenbank wird in Abbildung 6.7 aufgezeigt.

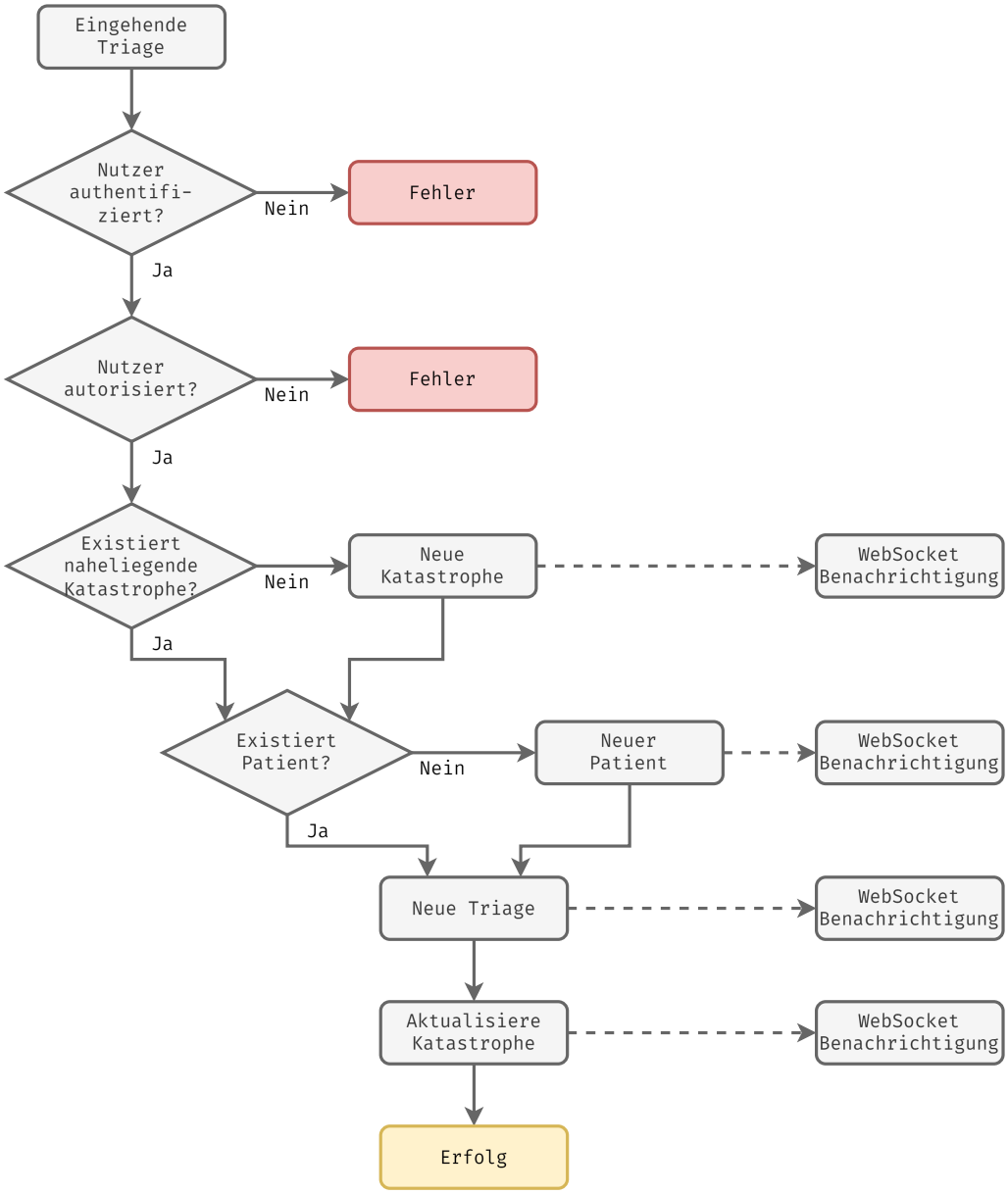

Abbildung 6.7: Ablaufdiagramm: Erstellung einer Triage

Die Prozedur beginnt mit der Durchführung einer Triage durch einen oder mehrere Ersthelfer am Schadensort der Katastrophe. Hier nutzt der Ersthelfer die App der *KatApp*, um die eigentliche Triage durchzuführen. Ist die Triage am Schadensort abgeschlossen, beginnt mit Eingang der Triage im Backend die eigentliche Prozedur der Triage-Verarbeitung.

Im ersten Schritt überprüft das Backend, ob der Nutzer/Ersthelfer, welcher die Triage durchgeführt hat, authentifiziert ist. Das System akzeptiert ausschließlich Triagen von authentifizierten Nutzern (siehe Unterabschnitt 2.8). Im Anschluss wird zusätzlich überprüft, ob die eingehende Anfrage autorisiert ist, sprich, ob entsprechende Berechtigungen zur Erstellung einer Katastrophe, eines Patienten und einer Triage vorhanden sind. Ist der Nutzer, welcher die Triage durchgeführt hat, nicht authentifiziert oder autorisiert, wird die Triage nicht vom Backend angenommen.

Ist der Nutzer überprüft, sucht das Backend im nächsten Schritt alle naheliegenden, aktiven Katastrophen in einem vorgegebenen Radius um die Triage herum, die zum Mandanten des Nutzers gehören. Werden existierende, aktive Katastrophen gefunden, so wird die naheliegendste ausgewählt und die eingehende Triage wird dieser einen Katastrophe zugeordnet. Falls keine umliegende Katastrophe gefunden wird, wird automatisch eine Neue eröffnet. Falls alle umliegenden Katastrophen inaktiv sind (Katastrophen beendet), wird ebenfalls eine Neue erstellt. Wird eine neue Katastrophe erstellt, so erhalten alle aktuell verbundenen Dashboard-Nutzer eine Benachrichtigung über die *WebSocket* API, und können somit die Katastrophe auf der im Dashboard abgebildeten Karte sehen, sofern sie demselben Mandanten angehören wie der Nutzer, welcher die Katastrophe/Triage erstellt hat.

Als nächstes überprüft das Backend, ob die eingereichte Triage zu einem bereits bestehenden Patienten gehört (Folgetriage), oder ob es sich um einen neuen Patienten (Ersttriage) handelt. Existiert der Patient bereits, wird die Triage diesem zugeordnet. Handelt es sich um einen neuen Patienten, wird dieser vom Backend erstellt. Auch hier benachrichtigt das Backend die aktiven Dashboard-Nutzer des gleichen Mandanten über die *WebSocket* API. Dadurch können entsprechende Dashboard-Nutzer die Aufnahme von Patienten in Echtzeit nachverfolgen.

Sind alle obigen Schritte erfüllt, so legt das Backend im vorletzten Schritt die eigentliche Triage an. Aufgrund des bisherigen Ablaufs kennt das Backend nun die Katastrophe und den Patienten und kann somit die Triage richtig einordnen. Die Dashboard-Nutzer erhalten auch hier die Daten der erstellten Triage in Echtzeit.

Abschließend wird die aktuelle Katastrophe aktualisiert. Hier wird zum einen der aktuelle Zeitstempel hinterlegt, welcher angibt, wann die Katastrophe zuletzt aktualisiert wurde (wichtig für die Identifizierung des Endes einer Katastrophe), zum anderen wird die Anzahl der in der Katastrophe befindliche Patienten aktualisiert (falls nötig). Dashboard-Nutzer haben somit stets einen Überblick auf die Gesamtzahl der Patienten einer Katastrophe. Ist die Aktualisierung erfolgreich, sind alle Schritte abgeschlossen und die Triage wird erfolgreich angelegt.

6.4 Verbindung via WebSocket

Der Verbindungsaufbau via *WebSocket* findet in zwei Schritten statt. Zunächst wird überprüft, ob der Nutzer, der die Verbindung aufbaut, authentifiziert ist. Ist das nicht der Fall, wird keine Verbindung zugelassen. Ist der Nutzer authentifiziert, werden seine Berechtigungen überprüft. Fehlen dem Nutzer entsprechende Rechte, wird der Verbindungsaufbau ebenfalls abgebrochen. Der Ablauf ist in Abbildung 6.8 abgebildet.

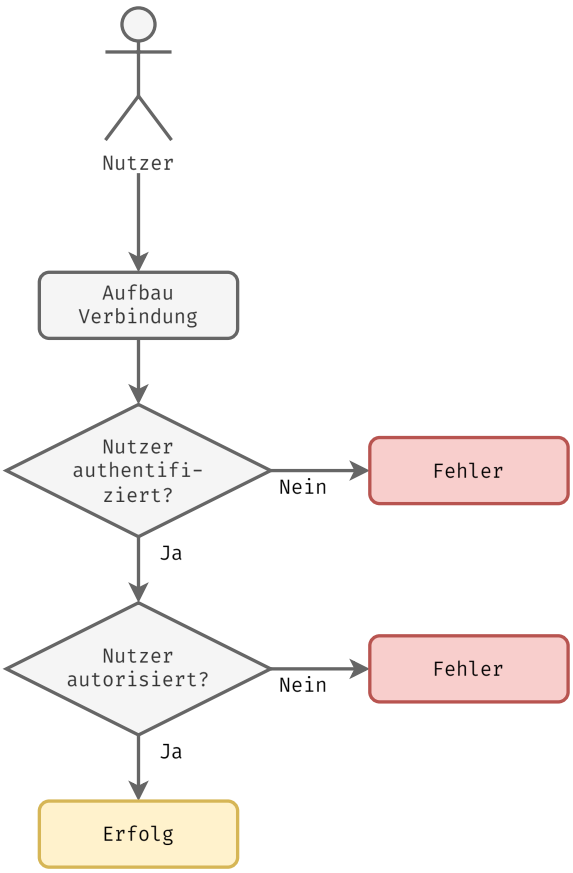

Abbildung 6.8: Ablaufdiagramm: Verbindungsaufbau via *WebSocket*

Beim Verbindungsaufbau werden stets Metadaten für jede Verbindung hinterlegt. Diese Daten enthalten sowohl den Mandanten des verbundenen Nutzers als auch dessen Berechtigungen. So kann beim Versenden von Benachrichtigungen geprüft werden, ob ein Nutzer berechtigt ist, die entsprechenden Daten zu empfangen.

7 Anforderungsanalyse

In diesem Kapitel werden die Anforderungen an die *KatApp* genauer definiert. Insbesondere wird hier auf die Anforderungen für die Umsetzung der Kooperation unter Mandanten, aber auch auf die Anforderungen, die damit einhergehen, eingegangen. Bevor die Anforderungen genauer definiert werden, werden noch einmal die technischen Herausforderungen dargelegt, die bisher in dieser Arbeit herausgearbeitet wurden.

Die Hauptproblemstellung dieser Arbeit ist die Kooperation von Mandanten während einer Katastrophe. Treffen mehrere Organisationen am Schadensort ein, muss bestimmt werden, welche Personen tatsächlich zur Katastrophenhilfe beitragen dürfen. Zudem muss sichergestellt werden, dass alle teilnehmenden Mandanten mit derselben Datenbasis arbeiten, um einander unterstützen zu können. Die aktuelle Herausforderung besteht darin, Daten mandantenübergreifend zugreifbar zu machen, ohne dass die Daten von Unberechtigten oder Dritten eingesehen werden können.

Darüber hinaus muss bestimmt werden, welcher der Mandanten die Einsatzleitung vor Ort übernimmt. Es ist aktuell unklar, wie diese Rolle besetzt wird. Muss der erste Mandant vor Ort diese Rolle übernehmen? Kann irgendeiner der Mandanten vor Ort diese Rolle übernehmen? Und falls einer der Mandanten vor Ort diese Rolle übernimmt, werden die anderen Mandanten automatisch von dieser Rolle ausgeschlossen?

Analog zur Einsatzleitung muss ebenfalls die Rolle der Leitstelle zugewiesen werden. Hier ergibt sich eine ähnliche Problematik wie bei der Einsatzleitung. Es stellt sich hier aber darüber hinaus die Frage, ob die Rolle der Leitstelle nicht von einem externen Mandanten, also einem Mandanten, der nicht bei der Katastrophe vor Ort ist, besetzt werden kann. Zudem ist offen, wie diese Rolle vom System ausgewählt wird. Erfolgt die Zuweisung automatisch durch das System, oder muss diese Rolle manuell durch einen Nutzer vergeben werden?

Neben diesen Kernproblematiken ergeben sich weitere Fragen, zum Beispiel wie eine Katastrophe beendet wird. Das Ende einer Katastrophe zu kennen ist ein wichtiger Faktor, wenn es um die Sicherheit der erfassten persönlichen sowie medizinischen Daten geht. Grundsätzlich sind die erfassten Daten von großem Wert für analytische Zwecke und sollten daher aufbewahrt werden. Sie sollten jedoch nicht universell zugänglich sein, sobald eine Katastrophe und damit die Gefahrensituation beendet ist.

Diese Problemstellungen werfen fachliche Fragen auf, aus welchen die Anforderungen an das System gewonnen werden. Zur Beantwortung dieser Fragen werden Experten aus dem medizinischen sowie medizintechnischen Sektor herangezogen. Im Dialog werden dann die Anforderungen an die *KatApp* erhoben. Als Experten werden Dr. med. Robert Wunderlich[18] und M.Sc. Eric Stricker[19] herangezogen, welche zugleich die Kunden und Auftraggeber der *KatApp* sind.

Die Experten sind mit den fachlichen Abläufen und den technischen Hintergründen durchaus vertraut, dennoch müssen technische Spezifika und Herausforderungen klar kommuniziert werden, um Kompromisse, falls diese nötig sind, für die Implementierung zu finden. Um die Fragen zu beantworten, werden diese in Kontext gebracht und diskutiert. Dadurch soll zunächst die ideale, fachlich korrekte Lösung herausgearbeitet und im Anschluss mit den technischen Mitteln abgeglichen werden. Das heißt, ist die ideale Lösung in der *KatApp* umsetzbar, wird dieses Ziel angestrebt, ist das nicht der Fall, soll eine alternative Lösung erarbeitet werden, welche die fachlichen Anforderungen zur Zufriedenheit der Experten erfüllt.

Um den Befragungsprozess so transparent wie möglich zu gestalten, werden die gestellten Fragen sowie deren Kontext im Folgenden dargestellt. Zudem wird eine kurze Zusammenfassung der jeweiligen Lösung bereitgestellt.

[18] `https://www.medizin.uni-tuebingen.de/de/das-klinikum/mitarbeiter/profil/346`
[19] `https://www.medizin.uni-tuebingen.de/de/das-klinikum/mitarbeiter/profil/257`

7.1 Kooperation

In einer Katastrophe kommt es häufiger vor, dass mehrere Organisationen/Mandanten (Kliniken, DRK, THW, etc.) vor Ort zusammenarbeiten. Wie wird die Teilnahme der individuellen Organisationen reguliert?

Kontext und technischer Hintergrund:

Es muss bestimmt werden, ob ein Teilnehmer berechtigt ist, Triagen an den Patienten in einer Katastrophe durchzuführen. Zudem muss sichergestellt werden, dass die triagierende Person mit den Daten der Ersthelfer anderer Organisationen, die an der Katastrophe beteiligt sind, arbeiten kann.

Aktueller Stand:

Aktuell können nur Triagen von Ersthelfern durchgeführt werden, welche dem Mandanten angehören, dem die Katastrophe zugeordnet ist. Die Katastrophe wird dem Mandanten zugeordnet, welcher die Organisation des Ersthelfers repräsentiert, der die erste Triage durchführt. Falls ein Ersthelfer eines anderen Mandanten eine Triage (in der örtlich gleichen Katastrophe) durchführt, wird eine neue Katastrophe erstellt, die dann dessen Mandanten zugeordnet wird. Die Katastrophe wird also zweimal geführt und keiner der Mandanten hat Zugriff auf die erfassten Patientendaten des jeweils anderen.

Mögliche Optionen:

- Jeder Ersthelfer (unabhängig seiner Mandantenzugehörigkeit) kann der Katastrophe beitreten? Folglich werden alle Mandanten als Teilnehmer geführt? (Automatisch)
- Unterstützende Mandanten müssen explizit vom leitenden Mandanten (Ersteller der Katastrophe) freigeschaltet werden? (Manuell)
- Andere

Lösung:

Aus dem Experteninterview ergibt sich, dass die Möglichkeit einer automatischen, eigenständigen Teilnahme der Mandanten klare Priorität hat. Man habe im Geschehen der Katastrophe keine Zeit, Mandanten manuell für die Teilnahme freizugeben. Die Lösung, dass Mandanten durch das Triagieren von Patienten in der Katastrophe zu Teilnehmern werden, sei eine realistische und fachlich akkurate Lösung und solle daher umgesetzt werden.

7.2 Einsatzleitung

In einer Katastrophe kommt es häufiger vor, dass mehrere Organisationen/Mandanten (Kliniken, DRK, THW, etc.) vor Ort zusammenarbeiten. Welche der teilnehmenden Organisationen übernimmt die Rolle der Einsatzleitung?

Technischer Hintergrund:

Durch die Wahl der Einsatzleitung wird der Zugriff auf die Echtzeit-Daten der Katastrophe im Dashboard reguliert. Nur Dashboard-Nutzer des/der ausgewählten Mandanten erhalten die Echtzeit-Daten (Triagen, Patienten) der Katastrophe.

Aktueller Stand:

Aktuell erhalten lediglich Nutzer des Mandanten Echtzeit-Daten, welcher die Katastrophe begonnen hat (Erstellung der ersten Triage).

Mögliche Optionen:

- Der Mandant, welcher als Erstes vor Ort ist, und mindestens eine Triage durchgeführt hat? (Automatisch)
- Einer der Mandanten, welche vor Ort sind, und mindestens eine Triage durchgeführt hat? (Automatisch)
- Alle Mandanten, welche vor Ort sind, und jeweils mindestens eine Triage durchgeführt haben? (Automatisch)
- Keine Regulierung? Zugriff auf das Dashboard und die Katastrophe für jeden authentifizierten Nutzer möglich (enthält nicht teilnehmende Mandanten)? (Automatisch)
- Ein Mandant muss explizit ausgewählt werden? Wer wählt den Mandanten aus? (Manuell)
- Andere

Folgefragen:

Wird der Mandant, welcher die Einsatzleitung stellt, automatisch ausgewählt, soll anderen Mandanten der Zugriff auf das Dashboard verweigert werden?

Lösung:

Analog zur Katastrophenteilnahme ergibt sich auch hier aus den Experteninterviews, dass eine automatische Zugriffsregelung klar zu bevorzugen ist. Es sei durchaus erlaubt, dass Rettungskräfte am Katastrophenstandort die Daten auf dem Dashboard einsehen dürfen, denn sie seien ja an der Katastrophe beteiligt. Daraus ergäbe sich eine Zugriffsberechtigung für alle an der Katastrophe beteiligten Mandanten.

Es trete jedoch auch häufiger der Fall auf, dass sich die Einsatzleitung bereits vor Ankunft am Katastrophenstandort in das Dashboard einloggen will. Zudem besteht die Möglichkeit, dass die Kollegen, welche die Einsatzleitung übernehmen, einem Mandanten zugeordnet sind, welcher kein aktiver Teilnehmer der Katastrophe ist. Aus diesen Kriterien ergäbe sich ein alternatives Zugriffskriterium, basierend auf dem Standort der Kollegen. Ein Zugriff sei daher auch für nicht teilnehmende Mandanten erlaubt, sofern diese sich in einem vorher festgelegten Radius um die Katastrophe herum befinden.

Aus der Perspektive der Experten wird die zweitgenannte Lösung präferiert, da sie alle potenziellen Zugriffsfälle der Einsatzleitung berücksichtigt. Eine Kombination beider Optionen kann als iterativer Verbesserungsprozess betrachtet werden.

7.3 Leitstelle

In einer Katastrophe kommt es häufiger vor, dass mehrere Organisationen/Mandanten (Kliniken, DRK, THW, etc.) vor Ort zusammenarbeiten. Welche Organisation übernimmt die Rolle der Leitstelle?

Technischer Hintergrund:

Durch die Wahl der Leitstelle wird der Zugriff auf die Echtzeit-Daten der Katastrophe im Dashboard reguliert. Nur Dashboard-Nutzer des/der ausgewählten Mandanten erhalten die Echtzeit-Daten (Triagen, Patienten) der Katastrophe.

Aktueller Stand:

Aktuell werden die Echtzeit-Daten nur an Nutzer des Mandanten geschickt, welcher die Katastrophe begonnen hat (Erstellung der ersten Triage).

Mögliche Optionen:

- Die Leitstelle wird von einem der teilnehmenden Mandanten übernommen?
- Die Leitstelle wird manuell gewählt und kann somit auch von einem nicht teilnehmenden Mandanten übernommen werden? (Manuell, zum Beispiel über Zuweisung durch Einsatzleitung?)
- Andere

Folgefragen:

Wird der Mandant, welcher die Leitstelle stellt, automatisch ausgewählt, soll anderen Mandanten der Zugriff auf das Dashboard verweigert werden?

Lösung:

Den Experten nach ist die Leitstelle stets die erste Organisation, welche Zugang zu den Katastrophendaten (Dashboard) bekommen sollte. Leitstellen werden deutschlandweit stets nach Landkreis vergeben. Das heißt, es ist bereits zu Beginn der Katastrophe klar, welche Leitstelle die Katastrophe übernimmt, basierend auf dem Landkreis, in dem die Katastrophe stattfindet. Auch hier möchte man auf eine automatische Vergabe der Berechtigung hinaus, da einfach keine Zeit für eine manuelle Vergabe sei. Eine Leitstelle solle also automatisch Zugriff auf alle Katastrophendaten innerhalb ihres Landkreises bekommen.

Alternativ, falls diese Lösung nicht umsetzbar sein sollte, könne für jeden Mandanten bei Registrierung bereits die verantwortliche Leitstelle angegeben werden. Dann könne man aus den teilnehmenden Mandanten automatisch auf die berechtigten Leitstellen schließen. Es wird jedoch zum Ausdruck gebracht, dass erstere Lösung klar zu bevorzugen sei.

7.4 Katastrophenende

Wie bestimmt das System, ob eine Katastrophe beendet ist?

Technischer Hintergrund:

Wird eine Katastrophe nicht für beendet erklärt und es findet eine neue Katastrophe beispielsweise ein Jahr später in der näheren Umgebung statt, wird die ein Jahr alte Katastrophe fortgesetzt, anstatt eine Neue zu starten. Außerdem muss der Datenzugriff nach Katastrophenende entsprechend reguliert werden (siehe unten).

Aktueller Stand:

Katastrophen werden für beendet erklärt, nachdem 24 Stunden lang keine neuen Daten (Triagen, Patienten) eingegangen sind.

Mögliche Optionen:

- Die Katastrophe wird für beendet erklärt, wenn in den letzten x Stunden keine neuen Daten eingegangen sind? (Automatisch, aktuelle Lösung)
- Die Katastrophe muss manuell (zum Beispiel durch die Einsatzleitung oder die Leitstelle) beendet werden? (Manuell)
- Andere

Lösung:

Typischerweise sei eine Katastrophe dann beendet, wenn der letzte Patient vom Katastrophengebiet abtransportiert/versorgt wurde. Da es aber aktuell keine Unterstützung für Rettungsmittel und Abtransporte in der *KatApp* gibt, lässt sich über den Abtransport des letzten Patienten nicht auf das Ende der Katastrophe schließen. Eine Katastrophe solle von der verantwortlichen Leitstelle beendet werden (über das Dashboard). Falls eine Katastrophe nicht von der Leitstelle beendet wird, soll die Katastrophe vom System automatisch 24 Stunden nach der letzten Triage für beendet erklärt werden.

7.5 Datenzugriff

Wer hat nach Beendigung der Katastrophe Zugriff auf die in dieser Katastrophe erfassten Daten (Triagen, Patienten)?

Technischer Hintergrund:

Nach Beendigung einer Katastrophe können Daten für analytische Zwecke exportiert werden. Da es sich um sensible Daten handelt, muss der Zugriff reguliert werden.

Aktueller Stand:

Jeder Nutzer mit entsprechenden Berechtigungen, welcher dem Mandanten angehört, der die Katastrophe behandelt hat, kann die Daten der Katastrophe im Nachhinein einsehen.

Mögliche Optionen:

- Entsprechende Nutzer aller teilnehmenden Mandanten können die Daten der Katastrophe im Nachhinein abfragen?
- Nur Nutzer des Mandanten, welcher die Einsatzleitung übernommen hat, können die Katastrophendaten abfragen?
- Nur Nutzer des Mandanten, welcher die Leitstelle gestellt hat, können die Katastrophendaten abfragen?
- Kein Mandant kann die Daten abfragen?
- Andere

Lösung:

Den Experten nach kämen zwei mögliche Lösungen für den Datenzugriff nach Beendigung einer Katastrophe infrage. Die Daten sollen entweder von der Leitstelle (über das Dashboard) oder dem globalen Systemadministrator exportiert werden können. Alternativ sollen die Daten ausschließlich vom Systemadministrator exportiert werden können. Falls individuelle Mandanten die Daten benötigten, sollen sie diese beim Administrator anfragen. Anschließend könne eine Vergabe der Daten durch den Administrator geprüft werden.

8 Konzeption

Im vorherigen Kapitel wurden mit Hilfe von Experten aus dem jeweiligen Sektor die fachlichen Anforderungen an den PoC, welcher in dieser Arbeit implementiert werden soll, ermittelt. Aus diesen fachlichen Anforderungen resultiert eine neue Version der *KatApp*. Konzeptionelle Anpassungen müssen an der Software gemacht werden, um die neuen Anforderungen abzubilden. Dazu sind größere Änderungen notwendig. Vor allem betroffen sind die Abläufe, welche in Abschnitt 6 abgebildet sind.

In den folgenden Unterkapiteln wird zunächst die Konzeption der neuen Version vorgestellt, in der unter anderem die Herausforderungen abgebildet werden, welche mit den neuen Anforderungen einhergehen. Anschließend werden die Baustellen im Code herausgearbeitet. Mit der Umsetzung der Anforderungen entsteht die *KatApp 2.0*.

8.1 Problematik

Bisher erfolgt die Authentifizierung und Autorisierung eines Nutzers über dessen Zugangstoken und seine Rolle, die mit ausgewählten Rechten einhergeht. Über das Zugangstoken kann zudem geprüft werden, welchem Mandanten der Nutzer angehört. Zur Erinnerung: bisher konnten Benutzer nur auf eine Katastrophe zugreifen, falls diese von seinem Mandanten eröffnet wurde.

Durch die Kooperation mehrerer Mandanten an einer Katastrophe ist die Mandantenzugehörigkeit kein ausreichendes Kriterium mehr. Durch die neuen Anforderungen ergibt sich, im Falle der Einsatzleitung, dass auch unbeteiligte Mandanten Zugriff auf eine Katastrophe bekommen können, sofern sich diese in einem gewissen Radius um die Katastrophe herum befinden. Zudem wird eine Leitstelle nie ein aktiv teilnehmender Mandant einer Katastrophe sein, der aber trotzdem Zugriff auf die Katastrophe erhalten muss, sofern sie im selben Landkreis stattfindet.

Die *KatApp* geht bisher zwar mit Geodaten um, jedoch nur auf Basis von Koordinaten. Das Backend erhält bei der Erfassung von Triagen stets den Standort des Ersthelfers in Form von Längen- und Breitengraden. Daraus lassen sich zwar mittels simpler Mathematik Distanzen oder Radien berechnen, um auf Nähe von Nutzern (zum Beispiel im Falle der Einsatzleitung) zu schließen, jedoch kann die *KatApp* weder Länder noch Bundesländer geschweige denn Landkreise ermitteln. Genau diese Daten sind jedoch nötig, um die Zuweisung einer Leitstelle zu einer Katastrophe automatisiert durchführen zu können.

8.2 Administrative Ebenen

Die *KatApp* wird primär für den deutschen Markt entwickelt. Daher bildet die Software auch die Abläufe des deutschen Katastrophenschutzes ab. Koordiniert wird der Katastrophenschutz stets von den Leitstellen, welche pro Landkreis vergeben werden. Landkreise sind Teil der Verwaltungsgliederung in Deutschland und bilden eine mehrerer administrativer Ebenen ab. Eine vereinfachte Darstellung der administrativen Ebenen in Deutschland ist in Abbildung 8.1 zu sehen.

Mit der Einführung von Leitstellen, welche in der vorherigen Version der *KatApp* lediglich implizit existieren, muss auch das Konzept der administrativen Ebenen eingeführt werden. Die *KatApp* benötigt also mindestens Informationen über die Landkreise in Deutschland, um Katastrophen und Leitstellen einander zuordnen zu können.

Während Landkreise bei der Registrierung von Leitstellen durchaus manuell angegeben werden können, ist die automatische Bestimmung des Landkreises einer Katastrophe alles andere als trivial. Aus den Anforderungen geht klar hervor, dass die Rettungskräfte vor Ort keine Zeit haben, den Landkreis einer Katastrophe manuell ins System einzugeben. Sollte dieser Schritt zudem vergessen werden, hätte die verantwortliche Leitstelle auch keinen Zugriff auf die Daten. Daher muss der Landkreis einer Katastrophe automatisiert bestimmt werden können.

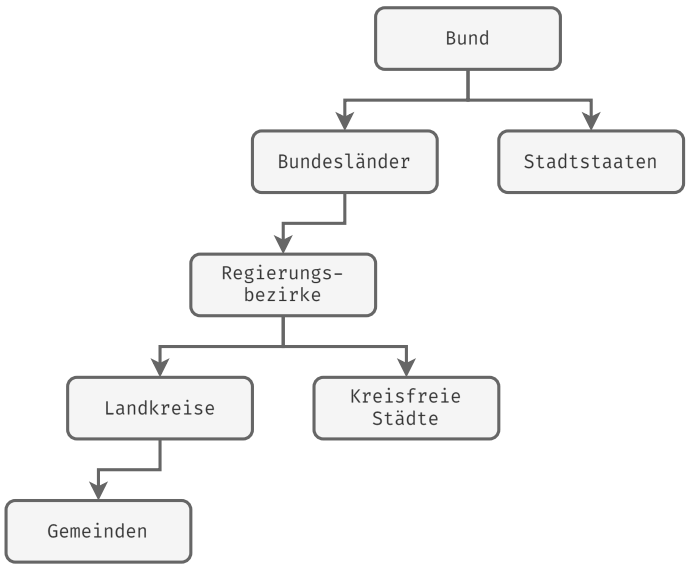

Abbildung 8.1: Vereinfachte Darstellung der Verwaltungsgliederung Deutschlands [32]

Für die Bestimmung von Landkreisen bzw. administrativen Ebenen aus Koordinaten sind erhebliche Datenmengen und komplexe Mathematik erforderlich. Den Prozess der Zuordnung von Koordinaten zu Adressdaten bzw. von Adressdaten zu Koordinaten nennt man Geokodierung. Eine solche Ermittlung erfolgt über Geokodierungsdienste, mit denen Adress- oder Regionaldaten über Koordinaten (Längen- und Breitengrade) bezogen werden können. Da Geokodierung riesige Mengen an Daten erfordert, muss diese Funktionalität über externe Dienste eingeführt werden. Bekannte Anbieter für Geokodierungsdienste sind unter anderem *Google* (via *Google Maps*) und *OpenStreetMap*.

Betrachtet man die Hierarchie der administrativen Ebenen Deutschlands (Abbildung 8.1), fällt zudem auf, dass im Falle eines Stadtstaats Ausnahmeregelungen auftreten. Die Verwaltung von Stadtstaaten bedarf keiner Landkreise. Verantwortliche Leitstellen bzw. die verantwortlichen administrativen Ebenen können sich daher in Sonderfällen verschieben. Je nach Abbildung dieser Daten im *OpenStreetMap* Datensatz, muss die *KatApp* mit diesen Sonderfällen umgehen können.

8.3 Internationaler Einsatz

Obgleich die Konzeption der *KatApp* primär auf den Einsatz in Deutschland ausgerichtet ist, erscheint eine Betrachtung ihres möglichen Einsatzes im internationalen Kontext als sinnvoll. Trotz der unterschiedlichen Abläufe des Katastrophenschutzes auf internationaler Ebene treten auch Gemeinsamkeiten auf. Eine recht häufig auftretende Gemeinsamkeit ist die Koordination des Katastrophenschutzes durch eine Organisation auf (über-) regionaler Ebene. Die daraus resultierende Problematik besteht in der Tatsache, dass die Verwaltungsgliederung in jedem Land eine andere ist. Die administrative Gliederung der Länder weist eine unterschiedliche Granularität auf, zudem sind die für die Koordination verantwortlichen Ebenen in der Regel verschieden. Die Vergabe der koordinierenden Organisation auf internationaler Ebene ist daher eine Herausforderung.

Geht man davon aus, dass stets genau eine Organisation (evtl. auch mehrere möglich) für eine bestimmte administrative Ebene verantwortlich ist, und diese administrative Ebene landesweit die gleiche Ebene ist (zum Beispiel Landkreise), lässt sich durch Angabe der verantwortlichen administrativen Ebene im Voraus (zum Beispiel durch eine Umgebungsvariable), durchaus ein internationaler Betrieb ermöglichen. Einige limitierende Faktoren dabei sind allerdings, dass zum einen, genau eine Instanz der *KatApp* mit den jeweiligen Einstellungen pro Land betrieben werden muss, und zum anderen, dass durch den Betrieb von einer *KatApp* Instanz pro Land, internationale Kooperationen, zum Beispiel an Landesgrenzen, ausgeschlossen sind.

8.4 Leitstellen und Landkreise

Für die erfolgreiche Registrierung eines Mandanten ist die Angabe der Adresse des Mandanten erforderlich. Mit Hilfe eines Dienstes für Geokodierung kann aus der für die Registrierung angegeben Adresse der deutsche Landkreis bestimmt werden, in dem der Mandant ansässig ist. Alternativ könnte man eine manuelle Eingabe des Landkreises durch den Mandanten für die Registrierung zulassen, da die Registrierung vor ihrer Freischaltung sowieso durch den Systemadministrator überprüft wird (siehe Unterabschnitt 2.8).

Eine neue Information jedoch, die für die Registrierung eines Mandanten notwendig ist, ist die Angabe, ob es sich bei diesem Mandanten um eine Leitstelle handelt oder nicht. Andernfalls kann das System nicht auf eine Leitstelle rückschließen und den entsprechenden Nutzern Zugriff gewähren. Für diese Zuordnung werden Rollen für Mandanten eingeführt (mehr dazu in Unterabschnitt 8.10).

Durch den neuen Registrierungsprozess kann somit jederzeit die Leitstelle für einen bestimmten Landkreis identifiziert werden. Damit ist die Grundlage für eine automatisierte Zuweisung von Leitstelle zu Katastrophe geschaffen.

8.5 Einsatzleitung und standortbasierter Zugriff

Während die Vergabe der Leitstelle über die Ermittlung der Landkreise automatisiert werden kann, ist der Zugriff der Einsatzleitung standortbasiert. In der vorherigen Version der *KatApp* ist die Zuordnung der Einsatzleitung klar. Nur Nutzer des Mandanten, welcher die Katastrophe eröffnet hat, können diese Rolle übernehmen. Nun, da mehrere Mandanten an einer Katastrophe teilnehmen können, ist diese Zuordnung nicht mehr eindeutig gegeben. Aus den Anforderungen ergibt sich, dass ein Mitglied der Einsatzleitung nur dann auf die Katastrophe zugreifen darf, wenn:

- der Nutzer entsprechende Berechtigungen hat und zu einem an der Katastrophe beteiligten Mandanten gehört, oder
- der Nutzer entsprechende Berechtigungen hat und sich innerhalb eines vorgegebenen Radius um den Katastrophenursprung herum befindet.

Daraus ergibt sich, dass der Standort des Zugreifenden zum Zeitpunkt des Zugriffs bekannt sein muss. Andernfalls kann dieser nicht mit dem Standort der Katastrophe abgeglichen werden. In allen Fällen muss der Nutzer authentifiziert sein.

Während die Angabe der Koordinaten einer Triage über den Körper der HTTP Anfrage eine valide Methode ist, ist diese Strategie eher problematisch für die Vergabe von Zugriffsberechtigungen. Ein großer Teil der Daten, welche die Einsatzleitung erhält, wird in Echtzeit über eine dauerhaft bestehende *WebSocket* Verbindung übertragen. Hierbei handelt es sich um sensible medizinische Daten, welche nur von autorisiertem Personal eingesehen werden dürfen. Da das Backend keine Informationen über den sich verbindenden Nutzer hat, außer dessen persönliche Informationen und Informationen über den Mandanten, kann eine Klienten-seitige Angabe von Standort-Informationen nicht trivial umgesetzt werden. Ein Klient könnte beliebige Koordinaten als seinen Standort angeben, und so Zugriff auf Katastrophen bekommen, die sich nicht in seiner Nähe befinden. Dieses Szenario ist strikt auszuschließen.

Grundsätzlich kommen zwei Wege für die Bestimmung des Klienten-Standortes infrage. Zum einen, könnte ein vorher festgelegter, geheimer Schlüssel, im Kopf der Anfrage übergeben werden. Durch diesen Schlüssel, könnte das Backend feststellen, dass die Standort-Angabe des Klienten durch eine autorisierte Anwendung ermittelt wurde, und somit den Standort als valide kennzeichnen und für die Berechnung des erlaubten Zugriffsradius verwenden. Zum anderen, könnte der Standort des Klienten über seine IP-Adresse ermittelt werden. Beide Ansätze haben ihre Problematiken.

Während die Angabe des Standorts durch den Klienten wesentlich genauer ist, lässt sich die Zugriffsbeschränkung dennoch umgehen, wenn es sich um einen besonders engagierten

Angreifer handelt. Standortermittlungen von Smartphones oder anderen mobilen Geräten, welche von der Einsatzleitung verwendet werden, können auf mehrfachem Wege umgangen werden, zum Beispiel durch die Nutzung von Entwicklerwerkzeugen, welche die manuelle Zuweisung eines Standorts für Testzwecke erlauben [33].

Die Standortermittlung über die Klienten-IP-Adresse ist eine vertrauenswürdige Methode, da das Backend keine Vertrauensbasis zum Klienten benötigt, ist jedoch sehr ungenau [34]. Dieses Problem könnte nur mittels eines großzügig gewählten Zugriffsradius mitigiert werden. Ein anderes Problem ist, dass die Ermittlung von Standorten basierend auf IP-Adressen auf großen Datenbanken beruht, die jedoch nicht für alle IP-Adressen Einträge besitzen. Es ist also durchaus möglich, dass eine Standortermittlung via IP fehlschlägt und eine Einsatzleitung keinen Zugriff auf eine Katastrophe bekommt. Des Weiteren besteht die Gefahr, dass der sich verbindende Nutzer einen VPN-Dienst verwenden könnte. Ist er zum Beispiel mit einem ausländischen Server verbunden, wird ihm evtl. zu Unrecht der Zugriff verweigert.

Je nachdem, wie engagiert ein Angreifer ist, gibt es in jedem Fall Wege, die Sicherheitsbeschränkungen des Systems zu umgehen. Für die *KatApp* gilt es daher nur, die Hürde für einen Angriff möglichst hoch zu gestalten. Die angestrebte Lösung für die Ermittlung des Standortes beim Verbindungsaufbau ist daher ein Hybrid aus obigen Methoden. Zuerst wird überprüft, ob der Klient seinen Standort selbst im Kopf der Anfrage bereitgestellt hat und ob dieser vertrauenswürdig ist. Wurde kein Standort bereitgestellt, wird versucht, den Standort mittels eines externen Dienstes über die Klienten-IP zu ermitteln. Schlägt beides fehl, so wird der Zugriff verweigert.

8.6 Problematiken eines standortbasierten Zugriffs

Die Festlegung eines zuvor definierten Radius als Zugriffsbedingung für die Einsatzleitung stellt eine anspruchsvolle Aufgabe dar. Der Radius sollte nicht zu groß sein, sodass unautorisiertes Personal sensible medizinische Daten von Patienten einsehen kann, und sollte auch nicht zu klein sein, sodass die Einsatzleitung in ihrer Arbeit behindert wird, indem sie ihre Rolle nicht rechtzeitig erfüllen kann.

Eine weitere Herausforderung ist, dass der Radius für die Zugriffserlaubnis in der Theorie ein relativer ist. Die Katastrophe selbst könnte von einem so großen Ausmaß sein, dass sich die Einsatzleitung bereits im Katastrophengebiet befindet, jedoch keinen Zugriff auf die Katastrophe bekommt, da der Katastrophenradius größer ist als der Radius, welcher die Zugriffsbedingung abbildet.

In der Praxis müsste man also beide Radien addieren. Bisher nimmt die *KatApp* für die Zuordnung von Triagen zu Katastrophen einen fixen Radius an. Was für diesen Einsatzzweck vorläufig ausreichen mag, ist jedoch problematisch für die Zugriffsberechtigung der Einsatzleitung. Für eine genaue Ermittlung des Zugriffsradius müsste also zunächst das Ausmaß der Katastrophe basierend auf den bereits durchgeführten Triagen gemessen werden, wobei die erste Triage, welche die Katastrophe eröffnet, als Zentrum der Katastrophe angesehen wird, und jede weitere Triage, basierend auf ihrer Distanz zum Zentrum, die maximale Ausdehnung angibt. Mit Hilfe dieser Strategie kann nicht nur das Ausmaß einer Katastrophe geschätzt, sondern auch ein sehr akkurater Zugriffsradius für die Einsatzleitung vergeben werden. Für die Implementierung des PoC jedoch wird vorerst ein fixer Zugriffsradius von fünf Kilometern als ausreichend erachtet.

8.7 Geokodierung und Standortermittlung

Die vorherige Version der *KatApp* nutzt keine Geokodierungsdienste. Die Zuordnung von Triagen und Patienten zu Katastrophen wird mittels Längen- und Breitengeraden durchgeführt. Durch die automatisierte Zuordnung von Leitstellen jedoch, welche über die deutschen Landkreise durchgeführt wird, werden erweiterte Geodaten benötigt. Grundsätzlich benötigt die *KatApp* drei verschiedene Zuordnungspfade zwischen Geodaten:

- eine Adresse muss einem Landkreis zugeordnet werden können,
- Koordinaten müssen einem Landkreis zugeordnet werden können, und
- IP-Adressen müssen Koordinaten zugeordnet werden können.

Die Zuordnung einer Adresse zu ihrem Landkreis dient nicht nur der Ermittlung des Landkreises bei der Registrierung eines Mandanten, sondern auch der Validierung der eingegebenen Adresse. So kann sichergestellt werden, dass eine Adresse auch tatsächlich existiert. Die Zuordnung von Koordinaten zu einem Landkreis dient der Ermittlung des Landkreises aus den Koordinaten einer Katastrophe. Der Landkreis muss hierbei ermittelt werden, um die verantwortliche Leitstelle zu bestimmen. Die Ermittlung von Koordinaten aus einer IP-Adresse wird für die Standortermittlung bei Verbindungen über *WebSocket* benötigt. Hierbei muss der Standort ermittelt werden, um zu überprüfen, ob sich die Einsatzleitung innerhalb eines bestimmten Radius um die Katastrophe herum befindet.

8.8 OpenStreetMap

OpenStreetMap[20] ist ein internationales Projekt, welches 2004 gegründet wurde. Ziel des Projekts ist es, eine freie Weltkarte zu erschaffen. Neben den Kartendaten, welche *OpenStreetMap* bereitstellt, bietet die Organisation auch die rohen Geodaten an, welche für die *KatApp* interessant sind. Der primäre Treiber für die Wahl von *OpenStreetMap* gegenüber der *Google Maps API* sind die Kosten. Die Geokodierungsdienste von *Google Maps* sind nicht kostenlos erhältlich [35].

Mit den Rohdaten von *OpenStreetMap* kann die *KatApp* wenig anfangen. Abfragen von derartigen Datenmengen sind durchaus komplex und würden einen enormen Aufwand bedeuten, falls diese eigenständig implementiert werden müssten. Zudem müsste sichergestellt werden, dass die Rohdaten stets beim Betrieb der *KatApp* zur Verfügung stehen, das heißt man müsste diese Daten auf AWS bereitstellen. Damit dieser Aufwand nicht betrieben werden muss, gibt es diverse Community-Projekte, welche auf den Daten der *OpenStreetMap* Organisation aufbauen und diese nutzbar machen. Dazu gehören unter anderem *Overpass*[21] und *Nominatim*[22].

Sowohl *Overpass* als auch *Nominatim* sind öffentlich zugängliche APIs, welche auf *OpenStreetMap* Daten beruhen. Beide Applikationen verfolgen unterschiedliche Ziele. Während sich *Nominatim* auf die Zuordnung von Koordinaten zu Adressen und Adressen zu Koordinaten beschränkt, dient *Overpass* eher einer generellen Abfrage von Geoinformationen. Solche Informationen sind zum Beispiel Metadaten ausgewählter Regionen oder Städte, sowie zum Beispiel Restaurants oder öffentliche Gebäude.

Während die API von *Nominatim* recht simpel gehalten ist, definiert *Overpass* eine proprietäre Abfragesprache, mit der *OpenStreetMap* Daten gezielt abgefragt werden können. Mit Hilfe der *Overpass* API können unter anderem die administrativen Ebenen Deutschlands, aber auch die anderer Länder, aus Koordinaten ermittelt werden.

Durch die freien und öffentlich verfügbaren APIs sind sowohl *Overpass* als auch *Nominatim* eine mehr als ausreichende Wahl für die Implementierung des PoC. Keine der beiden APIs ist bedeutend limitiert, daher stellen sie auch keine Einschränkung für die *KatApp* dar.

Im Falle einer lokalen Bereitstellung der *KatApp*, einem Szenario, bei dem keine Internetverbindung besteht und Komponenten der Software über ein lokales Netzwerk kommunizieren, können die öffentlichen APIs von *Overpass* und *Nominatim* nicht mehr verwendet werden. Sowohl *Overpass* als auch *Nominatim* sind jedoch als *Docker* Container und so auch für die Verwendung in einem lokalen Netzwerk verfügbar. Da beide APIs ihre Datenbasis von *OpenStreetMap* jeweils lokal im Container vorliegen haben, wird auch keine Internetverbindung benötigt, um die jeweiligen Daten abzufragen. Generell steht jedoch

[20] OpenStreetMap: `https://www.openstreetmap.org/`
[21] Overpass: `https://wiki.openstreetmap.org/wiki/Overpass_API`
[22] Nominatim: `https://nominatim.org/release-docs/latest/api/Overview/`

die Sinnhaftigkeit einer Abfrage von Geodaten in einem lokalen Netzwerk infrage. Genaueres dazu wird in Unterabschnitt 11.2 diskutiert.

8.9 IP-API

IP-API[23] ist eine öffentlich zugängliche API, mit welcher Koordinaten aus IP-Adressen ermittelt werden können. *IP-API* führt eine eigene Datenbank, weswegen dieser Service nicht frei für kommerzielle Zwecke erhältlich ist. Für nicht kommerzielle bzw. private oder Open-Source-Projekte hingegen, steht eine limitierte, freie API zur Verfügung. Die Limitierung umfasst eine temporäre Sperre für Anfragen, falls mehr als 100 Anfragen pro Minute von derselben IP-Adresse getätigt werden. [36]

Da es sich bei der *KatApp* zurzeit um ein nicht-kommerzielles Forschungs- und Innovationsprojekt handelt, ist eine Nutzung der API durchaus valide. Ein Hindernis stellt das Anfragenlimit der API dar. Betrachtet man jedoch erneut den Prozess zur Standortermittlung für die Einsatzleitung, stellt sich heraus, dass die Standortermittlung via IP-Adresse eine sekundäre Methode ist und daher selten verwendet wird. Für die Implementierung eines PoC ist die öffentliche Version der *IP-API* also ausreichend.

Im Gegensatz zu *Overpass* und *Nominatim* ist *IP-API* kein Open-Source Dienst, welcher in Form eines *Docker* Containers verfügbar ist. Für die Verwendung der proprietären API muss also eine Internetverbindung bestehen. In einem Szenario, in dem die *KatApp* lokal bereitgestellt wird, kann *IP-API* also nicht verwendet werden. Auch hier steht die Sinnhaftigkeit einer Standortbestimmung in einem lokalen Netzwerk infrage. Eine Diskussion dieses Themas erfolgt in Unterabschnitt 11.2.

8.10 Rollen & Mandanten

Wie in Unterabschnitt 2.9 beschrieben, verwendet die *KatApp* ein recht flexibles Rollen- und Rechte-System, womit individuelle Rechte flexibel zu Rollen gruppiert werden können. Zur Einfachheit der Benutzer erstellt die *KatApp* mit der Registrierung eines Mandanten einen Satz an Standardrollen. Diese sind ebenfalls in Unterabschnitt 2.9 beschrieben. Betrachtet man jedoch den generellen Ablauf einer Katastrophe und die damit einhergehende Fachlichkeit, sind diese Rollen nicht realitätsgetreu. Im ersten Schritt sollen die Standardrollen der *KatApp* angepasst werden, sodass diese die Fachlichkeit des Katastrophenschutzes abbilden. Dies hat zudem den Vorteil, dass die beteiligten Fachkräfte bereits mit der Terminologie vertraut sind. Die *KatApp* bildet nun folgende Rollen ab:

- Ersthelfer
- Einsatzleitung
- Leitstelle
- Mandanten-Admin
- KatApp-Admin

Um die Rolle der Leitstelle abzubilden, müssen Rollen nun auch auf Mandantenebene eingeführt werden. Durch die Zuordnung von Rollen zu Mandanten, können Leitstellen später einfacher und effizienter identifiziert werden, wenn es sich um die Zugriffsberechtigungen und Benachrichtigungen im Dashboard dreht. Die Einführung der neuen, fachlich akkuraten Nutzerrollen kann mit den bisherigen Mitteln realisiert werden. Für die Einführung von Mandantenrollen ist hingegen eine Anpassung des Datenmodells erforderlich. Dies hat direkte Auswirkungen auf die Registrierung von Mandanten. Neben den bisherigen Daten, die für die Registrierung einer Mandantschaft angegeben werden müssen, muss der Admin nun auch die Rolle seines Mandanten angeben. Das Rollen-System für Mandanten ist jedoch wesentlich simpler gestaltet. Vorläufig werden drei verschiedene Rollen für Mandanten eingeführt:

[23] IP-API: `https://ip-api.com/`

- Admin: bei diesem Mandanten handelt es sich um den Systemadministrator.
- Regulär: dieser Mandant nimmt keine besondere Rolle ein.
- Leitstelle: bei diesem Mandanten handelt es sich um eine Leitstelle.

Im Gegensatz zum Rollen- und Rechte-System für Nutzer, werden diesen Rollen keine individuellen Rechte zugeschrieben. Eine derart feingranulare Aufteilung ist derzeit nicht nötig. Eine tatsächliche Funktion hat bisher nur die Rolle *Leitstelle*. Alles andere wird über die feingranularen Nutzerrollen geregelt.

8.11 Katastrophenmanagement

Im Bereich Katastrophenmanagement muss zunächst die Grundlage für eine Zusammenarbeit mehrerer Mandanten in einer Katastrophe geschaffen werden. Dazu müssen alle beteiligten Mandanten auf dieselbe Katastrophe zugreifen können. Bisher ist die Sichtbarkeit einer Katastrophe auf den Mandanten beschränkt, welcher diese eröffnet hat. Dieses Verhalten führt zu Problemen bei der Erfassung von Triagen. Wie in Unterabschnitt 6.3 aufgezeigt, wird mit jeder durchgeführten Triage zuerst die zugehörige Katastrophe ermittelt. Dies geschieht auf Basis einiger Faktoren wie Nähe oder Katastrophenalter. Unter anderem werden hier auch Katastrophen nach Mandant gefiltert. Triagen eines Ersthelfers können also immer nur einer Katastrophe seines Mandanten zugeordnet werden. Dieses Verhalten hat ebenso zur Folge, dass, falls mehrere Mandanten Triagen in derselben Katastrophe (örtlich) durchführen, für jeden Mandanten eine separate Katastrophe mit separaten Patienten und Triagen erstellt wird.

Da die in einer Katastrophe teilnehmenden Mandanten im Voraus meist unklar sind, sollte jeder Mandant vor Ort in der Lage sein, an einer Katastrophe teilzunehmen. Die einzige Bedingung für einen Mandanten an einer Katastrophe teilzunehmen, wäre somit die Lokalität, und diese ist eindeutig durch den Standort einer Triage gegeben. Die Katastrophensuche kann somit vereinfacht werden, und der Mandant selbst ist kein Suchparameter mehr. Katastrophen sind infolgedessen öffentlich für Triagen zugänglich. Das heißt, die Zugehörigkeit zu einem bestimmten Mandanten wird nicht mehr geprüft.

Katastrophen sind nun jedoch nicht weiter nur einem einzigen Mandanten zugeordnet. Triagen können nun von mehreren Mandanten eingehen. Daher muss das Datenmodell der Katastrophe angepasst werden, sodass die involvierten Mandanten stets bekannt sind. Ein Mandant gilt nun als involviert in eine Katastrophe, wenn er mindestens eine Triage in dieser Katastrophe durchgeführt hat. Das heißt aber auch, dass ein Mandant nur durch eine Triage zum Teilnehmer einer Katastrophe werden kann.

Die Beendigung einer Katastrophe findet in der aktuellen Version der *KatApp* automatisch statt. Eine Katastrophe wird für beendet erklärt, wenn über 24 Stunden ohne Aktualisierung der Katastrophe vergehen. Somit gilt die Katastrophe 24 Stunden nach der letzten Triage als beendet.

In der neuen Version der *KatApp* soll dieses Kriterium aufrechterhalten werden. Darüber hinaus soll es jedoch möglich sein, die Katastrophe manuell zu beenden. Berechtigung hierfür soll ausschließlich die Leitstelle bekommen. Für die Umsetzung dieser Anforderung muss lediglich ein neuer REST Endpunkt eingeführt werden.

8.12 Datenzugriff & Export

Bisher sind sämtliche Dashboard-Nutzer der *KatApp* dazu in der Lage, die Daten einer Katastrophe nach deren Beendigung einzusehen und zu exportieren. Diese Regelung soll den Anforderungen entsprechend angepasst werden, sodass sowohl der Systemadministrator als auch die verantwortliche Leitstelle den Datenexport vornehmen kann. Einsicht über das Dashboard soll nach Beendigung einer Katastrophe keinem Nutzer zustehen.

Diese Einschränkungen können über das bisherige Rollen- und Rechtesystem sowie die neu eingeführten Mandantenrollen geregelt werden. Hierfür müssen die relevanten Endpunkte entsprechend der Anforderungen eingeschränkt werden.

9 Proof of Concept

Im vorliegenden Kapitel erfolgt eine detaillierte Betrachtung der Implementierungsaspekte des PoC. Diese bauen auf der vorangegangenen Konzeption auf. Dafür werden nötige Änderungen im Datenmodell sowie in den in der Implementierung (Abschnitt 6) vorgestellten Abläufen beleuchtet.

9.1 Datenmodell

Mit der Einführung von administrativen Ebenen und Regionen wird das bestehende Datenmodell um einen neuen Typen erweitert. Dieser bildet eine Region ab. Eine Region bezieht sich dabei immer auf die für das System relevante administrative Ebene, auf welcher die Leitstellen des Einsatzlandes verteilt werden. Die Daten einer Region werden über die in der Konzeption beschriebenen externen Dienste abgerufen. Abbildung 9.1 zeigt das Modell einer Region.

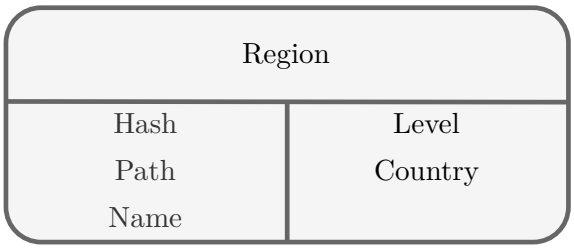

Abbildung 9.1: Datenmodell: Region

Für eine Region ist vor allem das Feld *Path* relevant. Dieses Feld bildet den Gesamtpfad von der niedrigsten bis zur relevanten administrativen Ebene in Landessprache ab. Für den Landkreis Tübingen, welcher sich auf der administrativen Ebene sechs befindet, sieht der gespeicherte Pfad zum Beispiel folgendermaßen aus: */deutschland/baden-württemberg/regierungsbezirk tübingen/landkreis tübingen*. Ein solcher Pfad enthält meist weniger Sektionen als die angegebene administrative Ebene, da diese je nach Land unterschiedlich vergeben und somit womöglich nicht alle Ebenen vergeben sind. Dieser Pfad wird zudem als Vergleichswert verwendet, um verschiedene Regionen miteinander zu vergleichen und somit zu bestimmen, ob eine Leitstelle berechtigt ist, auf Katastrophendaten zuzugreifen.

Damit diese Vergleiche effizient durchgeführt werden können, wird die relevante Region in jeder Katastrophenressource hinterlegt. Das Datenmodell wird also erweitert, indem die aktuelle Region, in der die Katastrophe stattfindet, sowohl in der Katastrophe selbst als auch in Patienten und Triagen hinterlegt wird. Dadurch werden Daten zwar redundant gespeichert, die Laufzeit hingegen wird durch Reduktion von Datenbankabfragen verkürzt.

Analog zu den Katastrophenressourcen werden auch andere Systemressourcen wie Benutzer oder Rollen angepasst. Anstatt nur einen Verweis auf den besitzenden Mandanten zu halten, werden zusätzlich Metadaten wie die Rolle des Mandanten sowie die Region, in der dieser ansässig ist, gespeichert.

9.2 Erstellung einer Triage

Die Erstellung einer Triage erhält ergänzend zum ursprünglichen Ablauf einen weiteren Schritt. In diesem Schritt wird aus den Koordinaten einer Initial-Triage (eine Triage, die eine Katastrophe eröffnet) die Region einer Katastrophe ermittelt. Abgebildet ist der veränderte Ablauf in Abbildung 9.2.

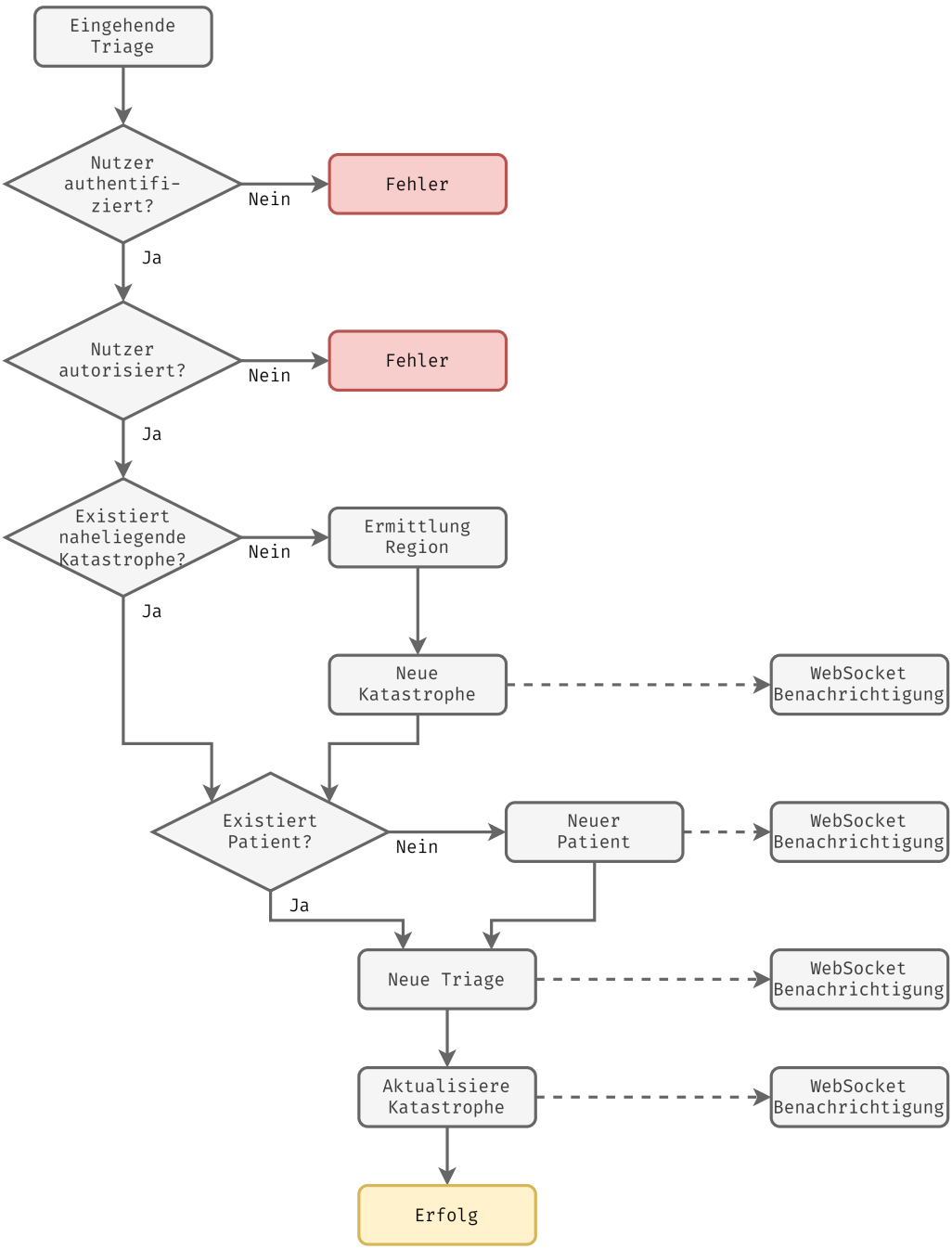

Abbildung 9.2: Ablaufdiagramm: Erstellung einer Triage (Neu)

Eine Problematik dieses Ablaufs ist durch den möglichen Misserfolg der Regionsermittlung gegeben. Um den Prozess des Katastrophenschutzes nicht zu behindern, wird durch einen potenziellen Fehler in der Regionsermittlung kein Abbruch des Ablaufs eingeleitet. Die Katastrophe, der Patient und die Triage werden weiterhin im System angelegt, allerdings mit unbekannter Region. Dieser initiale Fehler kann dazu führen, dass eine Leitstelle nicht über Katastrophen, Patienten oder Triagen benachrichtigt wird. Für eine Einsatzleitung stellt die fehlende Region kein Problem dar, da ihr Zugriff über die Koordinaten einer Katastrophe ermittelt wird.

Durch eine fehlende Region kann es zudem dazu kommen, dass Leitstellen Katastrophen nicht manuell beenden, oder Katastrophendaten nicht exportieren können. Grund für dieses Verhalten ist, dass die Region der Leitstelle aufgrund des fehlenden Eintrags der Region in der Katastrophe nicht gegeneinander abgeglichen werden können. In der Konsequenz wird einer Leitstelle der Zugang verweigert. Aktuell gibt es keinen Mechanismus im PoC, welcher eine Region bei fehlendem Eintrag nachträglich erneut ermittelt.

9.3 Standorterfassung via Trust-Key

Um einige der Problematiken von standortbasierter Autorisierung auszuschließen, wird eine Kombination der in Unterabschnitt 8.5 beschriebenen Lösungsmethoden gewählt. Die Standorterfassung via IP-Adresse des Klienten ist ungenau und erfordert Zeit, da die Erfassung über einen externen Dienst abgewickelt werden muss. Diese Methode wird daher als sekundäre Erfassungsmethode gewählt. Die primäre Standorterfassung wird durch den Klienten selbst durchgeführt. Damit das Backend dieser Standortangabe vertrauen kann, wird diese nur akzeptiert, wenn der Klient einen geheimen Schlüssel bereitstellt, welcher ausschließlich dem Backend und der Anwendung des Klienten bekannt ist. Auf diese Art und Weise wird sichergestellt, dass die Standortangabe des Klienten vertrauenswürdig ist. Durch die sichere Kommunikation via HTTPS ist zudem sichergestellt, dass der Kopf der HTTP Anfrage verschlüsselt ist. Der Trust-Key kann damit nicht auf triviale Art und Weise durch einen potenziellen Angreifer ausgelesen werden.

9.4 Verbindungsaufbau via WebSocket

Der Versand von Benachrichtigung via *WebSocket* wurde bisher durch die Zugehörigkeit eines Klienten zu seinem Mandanten eingeschränkt. Da diese Beschränkung im Rahmen der Kooperation unter Mandanten nun aufgehoben wird, muss die Einschränkung regions- bzw. standortbasiert durchgeführt werden.

Bei jeder neuen *WebSocket* Verbindung wird daher je nach Klient entweder die Region, in der der Mandant des Klienten ansässig ist (für Leitstellen), oder der aktuelle Standort des Klienten hinterlegt. Basierend auf diesen Daten kann bei der Erstellung einer neuen Triage exakt bestimmt werden, welche Klienten über Ereignisse benachrichtigt werden und welche nicht. Abgebildet ist diese Vorgehensweise in Abbildung 9.3.

Die Anpassung des Datenmodells hinsichtlich der Region eines Mandanten ist in diesem Kontext durchaus wertvoll. Handelt es sich um einen Nutzer in der Rolle der Leitstelle, so muss die Region nicht erneut über ein externes System abgefragt werden, da die Region bereits im Mandanten des Nutzers hinterlegt ist. Anders verhält sich dies, wenn es sich um ein Mitglied der Einsatzleitung handelt. Hier muss bei jedem Versuch eines Verbindungsaufbaus der Standort des Klienten bestimmt werden. Stellt der Klient seinen Standort selbst mittels Trust-Key bereit, ist kein Aufruf eines externen Dienstes erforderlich. Wird kein Standort in der Anfrage bereitgestellt, so wird versucht, den Standort über die IP-Adresse des Klienten zu ermitteln. Ist dieser Vorgang erfolgreich, wird ebenso wie in ersterem Fall der Standort des Klienten zu dessen Verbindung hinterlegt. Ist der Vorgang nicht erfolgreich, wird die Verbindung trotzdem mit Koordinaten von 0°N und 0°O angenommen. Aufgrund der Nullposition jedoch wird der Klient niemals Benachrichtigungen über eine Katastrophe erhalten (es kann mit hinreichender Sicherheit ausgeschlossen werden, dass eine Katastrophe bei Längen- und Breitengrad null stattfindet).

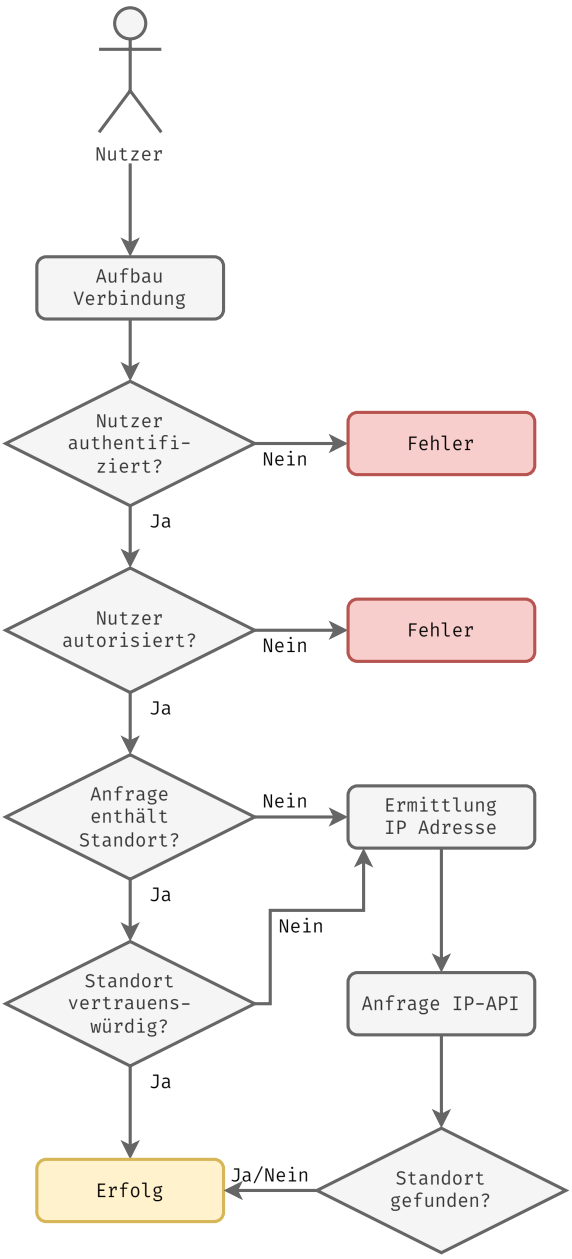

Abbildung 9.3: Ablaufdiagramm: Verbindung via *WebSocket* (Neu)

9.5 Abfrage von Katastrophendaten

Neben den asynchronen Benachrichtigungen via *WebSocket* können Katastrophendaten auch über die REST API der *KatApp* bezogen werden. Dies dient vor allem der initialen Datenerfassung durch das Dashboard. Analog zur *WebSocket* Verbindung muss daher bei jeder Abfrage die Rolle des Mandanten und dessen Region für Leitstellen bzw. der Standort des anfragenden Nutzers für Einsatzleitungen bekannt sein, damit evaluiert werden kann, ob der Zugriff berechtigt ist.

Die Vorgehensweise ist dabei analog zur *WebSocket* Verbindung (Abbildung 9.3). Im Gegensatz zur *WebSocket* Verbindung, bei der diese Prüfung einmalig durchgeführt werden muss und fortan besteht, muss die Standortabfrage bei jeder eingehenden REST Anfrage durchgeführt werden. Für einen Großteil der Endpunkte ist der Standort eines Klienten daher von enormer Bedeutung. Da das Dashboard Daten nach seiner Initialisierung überwiegend über die *WebSocket* API erhält, ist die Beeinträchtigung der Ausführungszeit für diese regelmäßige Standortprüfung der REST Endpunkte nicht weiter von Bedeutung.

9.6 Ende einer Katastrophe

Alternativ zur automatischen Beendigung einer Katastrophe nach 24 Stunden sollen Leitstellen in der Lage sein, Katastrophen in ihrem Landkreis manuell zu beenden. Um die manuelle Beendigung einer Katastrophe im Datenmodell abzubilden, wird eine neue Eigenschaft eingeführt: *IsTerminated*. Diese Eigenschaft wird bei Initial-Triagen geprüft, sodass beendete Katastrophen in der Nähe der Triage nicht mehr als laufend angesehen werden. Daher wird für die manuelle Beendigung einer Katastrophe ein neuer REST Endpunkt eingeführt, welcher diese Eigenschaft aktualisiert. Dieser Endpunkt ist nur für Leitstellen zugänglich.

Aus der Konzeption geht zudem hervor, dass der Export von Katastrophendaten eingeschränkt werden muss. Ein Export der Katastrophendaten kann nur durch Systemadministratoren der *KatApp* oder durch Leitstellen der entsprechenden Region vorgenommen werden. Im Rahmen der Implementierung des PoC werden die Endpunkte für den Export daher hinsichtlich dieser Bedingungen eingeschränkt.

10 Evaluierung

Im vorliegenden Kapitel erfolgt eine Evaluierung der *KatApp* 2.0. Dabei wird untersucht, ob der implementierte PoC die in der Anforderungsanalyse (Abschnitt 7) näher spezifizierten Anforderungen erfüllt. Darüber hinaus zielt die Evaluierung darauf ab, potenzielle Schwachstellen und Problematiken im aktuellen PoC zu identifizieren, die im Rahmen iterativer Verbesserungen adressiert werden müssen.

10.1 Kriterien & Metriken

Für eine erfolgreiche Evaluierung müssen entsprechende Kriterien bzw. Metriken etabliert werden, anhand derer das Ergebnis gemessen werden kann. Dieses Unterkapitel widmet sich den für den PoC angeführten Kriterien und Metriken sowie deren Relevanz.

Das primäre Kriterium für eine Evaluierung des PoC ist dessen fachliche Korrektheit. Diese wird anhand der Anforderungen, welche in Abschnitt 7 beschrieben werden, abgeleitet. Abgenommen werden diese Anforderungen durch die Projektleiter bzw. medizinischen Experten, welche in das Projekt involviert sind. Obwohl das System fachlich korrekt agieren kann, bedeutet dies nicht, dass es keine Mängel bzw. Herausforderungen geben kann. Aus der Analyse fachlicher Korrektheit sollen zudem potenzielle Beeinträchtigungen identifiziert und evaluiert werden. Fachliche Korrektheit des Systems hat oberste Relevanz.

Das sekundäre Kriterium für die Evaluierung des PoC ist die Zuverlässigkeit. Ein System kann die Anforderungen in ausgewählten Szenarien erfüllen, jedoch in Randfällen oder selten auftretenden Szenarien versagen. Daher ist es erforderlich, das System auch diesen Randfällen auszusetzen. Die Zuverlässigkeit des Systems spielt ebenfalls eine wesentliche Rolle.

Aus den Anforderungen des PoC geht zudem hervor, dass Hilfskräfte im Rahmen einer Katastrophe aufgrund zeitlicher Limitationen keine manuellen Einstellungen am System vornehmen können. Diese Eigenschaften weisen auf eine benutzerfreundliche Bedienweise hin, welche hiermit als tertiäres Kriterium etabliert wird.

Neben den Kriterien für fachliche Korrektheit und Benutzerfreundlichkeit sollen weithin diverse Metriken angeführt werden, welcher einer messbaren Bewertung des PoC dienen. Aufgrund der Einführung geobasierter Autorisierung kann die Ausführungszeit diverser Endpunkte beeinträchtigt werden. Der zentrale Endpunkt für die Erstellung einer neuen Triage, welcher bereits durchaus schwergewichtig ist, ist ebenfalls davon beeinflusst. Als primäre Metrik für eine objektive Bewertung wird also die Ausführungszeit herangeführt.

10.2 Allgemeines

Damit der implementierte PoC effizient getestet werden kann, müssen einige Voraussetzungen gegeben sein. Für eine Evaluierung wird unter anderem eine breite Datenbasis benötigt. Da sich die Evaluierung um die Kooperation mehrerer Mandanten dreht, müssen verschiedene Mandanten im System registriert sein. Zudem müssen Nutzer unterschiedlichster Rollen im System angelegt werden. Darüber hinaus werden diverse App-Zugänge benötigt, um die Triagen durchzuführen.

Die Evaluierung eines geobasierten Systems gestaltet sich von Grund auf schwierig, da als Grundlage für solche Tests reale Geodaten verwendet werden müssen. Für eine Evaluierung des PoC müssen daher Testszenarien entwickelt werden. Ein Testszenario orientiert sich dabei an einem möglichst realen Szenario, in welcher die *KatApp* zum Einsatz kommen könnte.

10.3 Testszenarien

Um ein besseres Verständnis der Testverfahren zu schaffen, werden die Bestandteile eines Testszenarios genauer dargelegt. Ein Testszenario setzt sich dabei immer aus einem bei-

spielhaften Katastrophenszenario sowie mehrerer Leitstellen, Einsatzleitungen und Ersthelfer zusammen. Ein Testszenario muss dabei immer bestimmte Voraussetzungen abdecken. Damit ein Testszenario als vollständig abdeckend angesehen werden kann, muss es mindestens folgende Bedingungen erfüllen:

- es muss eine Katastrophe geben,
- es muss mehrere Patienten geben,
- es müssen mehrere Ersthelfer vor Ort sein, welche aus mindestens zwei verschiedenen Organisationen stammen,
- es muss mindestens zwei Einsatzleitungen geben, wobei sich eine innerhalb des validen Katastrophenradius und eine außerhalb davon befindet,
- und es muss mindestens zwei Leitstellen geben, wobei sich eine innerhalb des Landkreises der Katastrophe und eine außerhalb davon befindet.

Durch diese Bedingungen wird sichergestellt, dass jedes Verhalten der *KatApp* in einer potenziellen Katastrophe abgedeckt wird. In einem realen Szenario werden keine zwei Einsatzleitungen vor Ort sein, ebenso werden keine zwei Leitstellen verantwortlich sein. Diese Bedingungen dienen als Verifikation gegenüber den verantwortlichen Instanzen. Eine Leitstelle außerhalb des Landkreises oder eine Einsatzleitung außerhalb des Katastrophenradius sollte keine Echtzeit-Daten der Katastrophe erhalten, die Instanzen innerhalb durchaus.

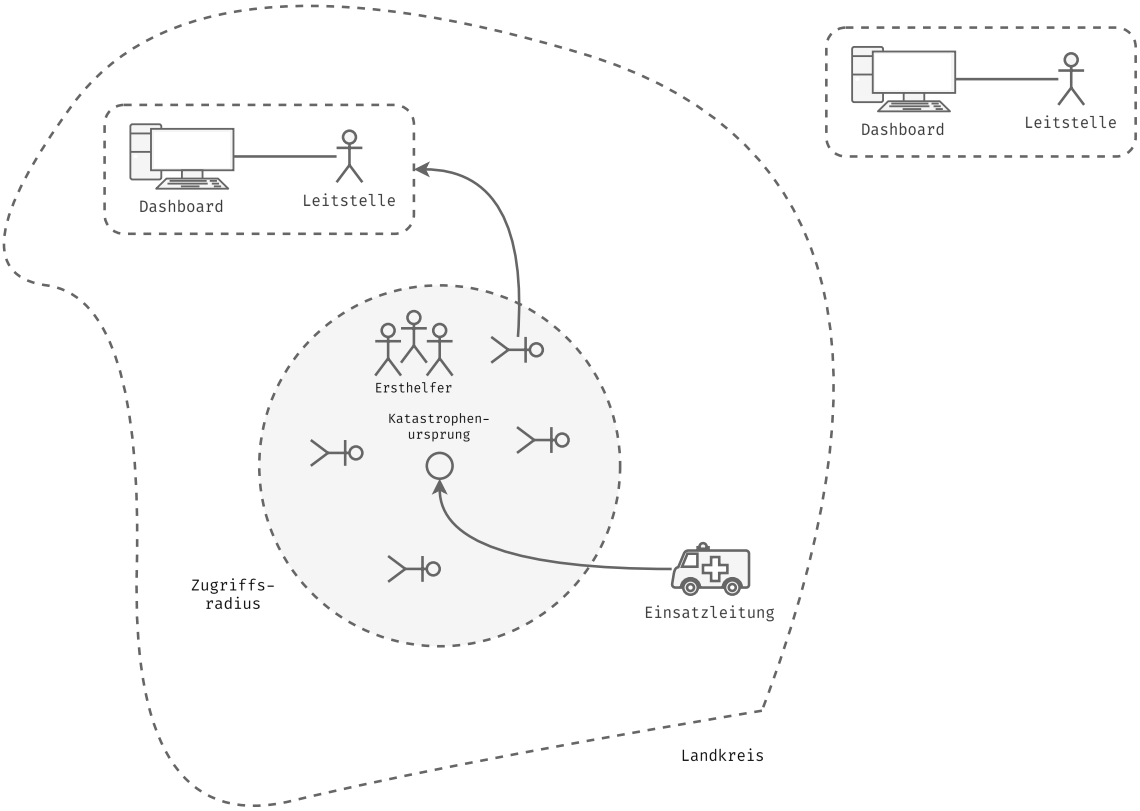

Abbildung 10.1: Geobasierte Autorisierung

Diese Testverfahren können mit Hilfe eines strukturierten Datenformats wie *JavaScript Object Notation* (JSON) beschrieben werden. Da aktuell keine Möglichkeit besteht, diese Testverfahren zu automatisieren, werden diese in einem *GeoJSON*[24] validen Format abgelegt, um das Szenario visuell auf einer Karte darstellen zu können. Dadurch wird das Testverfahren wesentlich anschaulicher. Abbildung 10.1 zeigt den Aufbau eines möglichen Testszenarios.

[24] GeoJSON: `https://geojson.io/`

Da *GeoJSON* das Einbetten beliebiger Metadaten erlaubt, wäre eine Automatisierung solcher Szenarien in Zukunft durchaus denkbar. Herausfordernd wird das Verfahren allerdings vor allem durch die notwendige Datenbasis, welche initial bestehen muss, damit ein Testszenario durchgeführt werden kann. Es muss registrierte Organisationen für Leitstellen, Einsatzleitungen und Ersthelfer geben, um die fachliche Korrektheit in einem Testszenario prüfen zu können.

10.4 Testmittel & Ablauf

Da aktuell keine Möglichkeit besteht, die etablierten Szenarios automatisiert zu testen, werden alle in der Evaluierung durchgeführten Tests manuell durchgeführt. Für die Durchführung dieser Tests wird stets eine lokal bereitgestellte *KatApp* Instanz verwendet, deren Abhängigkeiten via *Docker* ebenfalls lokal bereitgestellt werden. In dieser *KatApp* Instanz werden die benötigten Mandanten, Benutzer und Geräte bereits hinterlegt, sodass mit einer präparierten Datenbasis gearbeitet werden kann.

Für die Ausführung der manuellen Tests wird *Postman*[25] verwendet. Mit Hilfe von *Postman* können REST APIs effizient getestet werden. Zudem kann eine Verbindung zur *WebSocket* API der *KatApp* über *Postman* aufgebaut werden. So kann verifiziert werden, dass die entsprechenden Klienten eines Szenarios korrekt mit relevanten Echtzeit-Daten versorgt werden.

Der Ablauf eines Testszenarios sieht dabei folgendermaßen aus: zunächst wird sichergestellt, dass die in dem Szenario beteiligten Leitstellen mit der *KatApp* verbunden sind (über die *WebSocket* API). Analog werden die beteiligten Einsatzleitungen mit Angabe ihres jeweiligen Standortes verbunden. Ist das der Fall, wird eine Initial-Triage durch eine beliebige Organisation innerhalb des relevanten Landkreises durchgeführt. Die Leitstelle des Landkreises sowie die Einsatzleitung, welche sich innerhalb des Autorisierungsradius der Katastrophe befindet, sollten nun bereits über die neue Katastrophe, den neuen Patienten sowie die neue Triage benachrichtigt werden. Die Leitstelle außerhalb des Landkreises sowie die Einsatzleitung außerhalb des Autorisierungsradius sollten diese Informationen nicht erhalten. Ist das der Fall, spiegelt dies korrektes Verhalten wider, welches durch die Durchführungen weiterer Triagen verifiziert werden kann. Ist das nicht der Fall, ist dies ein Fehlverhalten der *KatApp*, welches adressiert werden muss.

10.5 Fachliche Korrektheit

Um die fachliche Korrektheit der *KatApp* zu überprüfen, wird eine Reihe von Testszenarien etabliert, welche anhand des beschriebenen Ablaufs durchlaufen werden. Die *KatApp* wird als fachlich korrekt angesehen, wenn sie sich in allen etablierten Szenarien den Erwartungen, welche aus der Anforderungsanalyse hervorgehen (Abschnitt 7), entsprechend verhält.

Des Weiteren soll mindestens ein Testszenario etabliert werden, in welchem zwei Katastrophen, welche sich nahe beieinander befinden, parallel stattfinden. Eine Einsatzleitung sollte, falls sie sich in einem Gebiet befindet, wo sich beide Autorisierungsradien überschneiden, Echtzeit-Daten für beide Katastrophen erhalten.

Die Durchführung diverser Testszenarien in unterschiedlichen geografischen Regionen zeigt, dass kein Fehlverhalten der *KatApp* besteht. Der PoC wird somit als fachlich korrekt angesehen.

10.6 Zuverlässigkeit

Bei Betrachtung der administrativen Ebenen Deutschlands (Abbildung 8.1) wird ersichtlich, dass im Falle eines Stadtstaates eine andere Verwaltungsgliederung vorliegt als es bei

[25] Postman: `https://www.postman.com/`

regulären Bundesländern der Fall ist. Die für den Katastrophenschutz zuständigen Leitstellen sind administrativ einer anderen Ebene zuzuordnen als die Leitstellen der regulären Bundesländer. Die *KatApp* muss mit diesen Sonderfällen umgehen können.

Um die Zuverlässigkeit der *KatApp* in solchen Szenarien zu testen, werden manuelle Testszenarien entwickelt, die auf einer lokal bereitgestellten *KatApp* getestet werden. Das System wird als zuverlässig erachtet, sollte es in allen entwickelten Szenarien fachlich korrekt agieren.

In Deutschland existieren drei Stadtstaaten, für welche Testszenarien entwickelt werden können. Diese sind Berlin, Hamburg und Bremen. Dabei stellt Bremen ein Sonderfall unter den Sonderfällen dar, da das Bundesland Bremen zwei Städte umfasst. Die *KatApp* gilt als zuverlässig, wenn die Leitstellen dieser drei Stadtstaaten korrekt (entsprechend der Anforderungen) über das Katastrophengeschehen in ihrem jeweiligen Bundesland benachrichtigt werden.

Sowohl in Hamburg als auch Berlin wurden Testszenarien durchgeführt. In allen Fällen agiert der PoC der *KatApp* wie erwartet. Aufgrund der Verwaltungsgliederung in den Stadtstaaten und den damit einhergehenden administrativen Ebenen jedoch kann nur eine Leitstelle je Stadtstaat vergeben werden. In Ausnahmefällen wie dem Bundesland Bremen, in welchem sich die Stadt Bremen und die Stadt Bremerhaven befinden, kann aufgrund einer anderen Verwaltungsgliederung eine Leitstelle je Stadt, also eine Leitstelle für Bremerhaven und eine Leitstelle für Bremen vergeben werden. Dieses Verhalten wird aus den Testszenarien in Bremen und Bremerhaven ersichtlich. Ob dieses Verhalten den fachlichen Abläufen der Stadtstaaten entspricht, bleibt mit den Experten zu klären.

10.7 Benutzerfreundlichkeit

Aus der Anforderungsanalyse (Abschnitt 7) ergibt sich, dass das System so benutzerfreundlich wie möglich sein soll. Benutzerfreundlichkeit bedeutet in diesem Fall minimale Interaktion, welche über den Normalgebrauch des Systems herausgeht. Ob als Einsatzleitung am Einsatzort angekommen oder als Mitarbeiter der Leitstelle in der Zentrale, die *KatApp* soll einfach funktionieren. Das heißt, Leitstellen werden automatisch basierend auf dem Standort einer Katastrophe autorisiert und benachrichtigt und Einsatzleitungen können Katastrophen einsehen, sobald sie sich in einem festgelegten Radius befinden. Zusammengefasst sollen also keine manuellen Eingriffe erforderlich sein.

Aus den Testszenarien, welche für fachliche Korrektheit und Zuverlässigkeit durchgeführt werden, geht hervor, dass sowohl die Autorisierung der Leitstellen, als auch die der Einsatzleitungen erwartungstreu vollautomatisch abläuft. Als potenzieller Schwachpunkt jedoch kann das Autorisierungsverfahren der Einsatzleitungen identifiziert werden. Sollte sich eine Einsatzleitung auf dem Weg zur Katastrophe außerhalb des Autorisierungsradius befinden und über das Dashboard einloggen, wird die Einsatzleitung ohne manuelles Eingreifen keine Katastrophendaten empfangen. Der Standort der Einsatzleitung wird nur einmalig bei Verbindungsaufbau erfasst und fortan nicht aktualisiert. Dadurch kann die Einsatzleitung zwar in den Autorisierungsradius eintreten, aufgrund des letzten bekannten Standortes jedoch, welcher außerhalb des Autorisierungsradius liegt, ist die Einsatzleitung nicht für den Empfang von Daten berechtigt. Um dieses Verhalten zu umgehen, muss die Einsatzleitung die Seite des Dashboards manuell aktualisieren. Dadurch wird die Verbindung zur *KatApp* neu aufgebaut, und der aktuelle Standort wird für die Autorisierung verwendet.

Mit Ausnahme dieses Verhaltens erfordert das System kein zusätzliches Eingreifen durch einen Benutzer. Das Verhalten des PoC wird als ausreichend angesehen und somit besteht der PoC auch in der Kategorie Benutzerfreundlichkeit.

10.8 Ausführungszeit

Die *KatApp* ist ein Echtzeitsystem und sollte sich auch entsprechend verhalten. Daher ist die Ausführungszeit individueller Endpunkte von großer Wichtigkeit und bildet nicht umsonst eines der Qualitätsziele der Backendarchitektur ab. Gerade durch die Einführung externer Dienste für die Abfrage von Geodaten kann die Ausführungszeit diverser Endpunkte signifikant beeinflusst werden. Vor allem relevant für diese Prüfung ist der Endpunkt für die Erstellung einer Triage. Neben der reinen Triage-Erstellung übernimmt dieser Endpunkt noch eine ganze Menge weiterer Verantwortlichkeiten (Unterabschnitt 6.3). Unter anderem muss der Landkreis bzw. die Region einer Katastrophe abgefragt werden, wenn eine Initial-Triage (eine Triage, welche eine Katastrophe eröffnet) im System eingeht.

Gerade bei der Abfrage von Geodaten können Ausführungszeiten, basierend auf der Komplexität der abgefragten Region, stark variieren. Die Ausführungszeiten für eine Initial-Triage sollten also gemessen werden. Dafür wird eine lokale *KatApp* Instanz herangezogen, welche über ein Skript, welches viele Initial-Triagen in Folge durchführt, belastet wird. Dieses Skript misst zudem die Differenz zwischen Anfrage- und Antwortzeit und mittelt diesen Wert anschließend, um ein Bild davon zu bekommen, wie lange eine Initial-Triage dauert.

Zudem sollte die Ausführungszeit von Triagen gemessen werden, welche in einer bereits existierenden Katastrophe stattfinden (keine Initial-Triagen). Durch die geobasierte Autorisierung ist die Logik in diesem Endpunkt komplexer geworden. Die Ausführungszeit des Endpunkts kann bedingt durch die Anzahl verbundener Dashboards beeinflusst werden. Auch hier soll die durchschnittliche Ausführungszeit ermittelt werden.

Die effiziente Ausführung dieses Endpunkts ist insofern von Bedeutung, dass die Ersthelfer nicht in ihrer Arbeit behindert werden. Wenn ein Ersthelfer mehrere Patienten in knapper Abfolge triagiert, sollten keine Wartezeiten zwischen den Patienten entstehen.

Die Testmessungen des Endpunkts werden auf aufseiten des Klienten durchgeführt, sodass der gesamte Rundweg einer Anfrage erfasst wird. Dabei umfasst der Rundweg zudem den lokalen *KatApp Gateway*, welcher als Reverse-Proxy fungiert.

Für die Messung der Initial-Triagen werden 1000 Triagen durchgeführt, welche überall auf der Welt verteilt stattfinden. Im Durchschnitt liegt die Ausführungszeit eines Rundwegs zwischen 200 ms und 400 ms. Über die Gesamtzahl der Triagen hinweg liegt der erfasste Durchschnitt bei 326 ms. Den Qualitätszielen (Abschnitt 5) zufolge liegt die Ausführungszeit also im unteren Drittel des validen Bereiches.

Für die Messung von Triagen innerhalb einer Katastrophe wird dasselbe Messungsverfahren verwendet wie für die Initial-Triagen. Die Messungen werden dabei mit vier verbundenen Klienten durchgeführt. Im Rahmen der Messung werden 1000 Triagen mit 1000 unterschiedlichen Patienten angelegt. Die Ausführungszeit eines Rundwegs liegt dabei zwischen 30 ms und 40 ms. Über alle 1000 Triagen hinweg wird ein Durchschnitt von 36 ms gemessen.

10.9 Ergebnisanalyse

Aus den vorliegenden Testverfahren wird die Erkenntnis gewonnen, dass der entwickelte PoC den Anforderungen entsprechend als funktional einzustufen ist, jedoch auch Optimierungspotential aufweist.

Der PoC erfüllt die in Abschnitt 7 spezifizierten Anforderungen. Eine Kooperation unter Mandanten ist nun ohne zusätzlichen Aufwand durch Einsatzpersonal möglich. Dieses Verhalten ist ein großer Gewinn für die Einsatzkräfte, da sich der Ablauf des Katastrophenschutzes mit der *KatApp* nicht geändert hat. Zudem können Administratoren weiterhin ihre Organisationen nach individuellen Bedürfnissen verwalten, ohne dass dieser Mechanismus den fachlichen Ablauf des Katastrophenschutzes beeinflusst. Des Weiteren ist der Betreiber der *KatApp* in der Lage, die Kosten für jeden einzelnen Mandanten zu kalkulieren, trotz der bestehenden Kooperation.

Eine Befürchtung bei geobasierter Autorisierung ist die Zuverlässigkeit der externen Dienste, Resultate für entsprechende Geoanfragen zu liefern. Aus den durchgeführten Testszenarien jedoch lässt sich schließen, dass die geobasierte Autorisierung zuverlässig funktioniert. In den Testszenarien ist nicht ein Fall aufgetreten, bei welchem die Autorisierung aufgrund einer nicht funktionierenden Geoanfrage scheiterte.

Trotz der erheblichen Anpassungen der *KatApp*, die im Rahmen dieser Arbeit durchgeführt wurden, gelten die Qualitätsziele der Architektur weiterhin als erfüllt.

Optimierungspotenzial besteht vor allem im Bereich der geobasierten Autorisierung der Einsatzleitung. Da die Einsatzleitung im Gegensatz zur Leitstelle womöglich in Bewegung ist (während der Anfahrt zur Katastrophe), und die geobasierte Autorisierung nur einmalig bei der Anmeldung im Dashboard durchgeführt wird, ist die Einsatzleitung in einem solchen Fall dazu gezwungen, dass Dashboard manuell zu aktualisieren, bis sie sich innerhalb des Autorisierungsradius einer Katastrophe befindet. Dieser manuell Eingriff ist zwar nicht von erheblicher Beeinträchtigung, jedoch kann dieses Verhalten aus Sicht der Nutzer das Vertrauen in das System beeinträchtigen.

Das beobachtete Verhalten ist durch architekturbedingte Limitationen zu erklären, die in der *KatApp* aktuell bestehen. Die *WebSocket* Verbindung, welche zwischen Dashboard und Backend besteht, ist derzeit unidirektional. Das bedeutet, dass das Backend zwar Nachrichten an das Dashboard versenden kann, Nachrichten vom Dashboard jedoch vom Backend ignoriert werden. Aus diesem Grund ist keine Aktualisierung des Standorts der Einsatzleitung in Echtzeit möglich. Mithilfe einer bidirektionalen Verbindung könnte dieses Problem mitigiert werden.

11 Schluss

Das vorliegende Kapitel widmet sich abschließenden Überlegungen zu dieser Arbeit. Zunächst erfolgt eine Zusammenfassung der Arbeit. Im Anschluss werden die in der Einführung dargelegten Forschungsfragen in einem Fazit reflektiert. Nach einer kritischen Reflexion werden mögliche Erweiterungen der *KatApp* erörtert. Abschließend erfolgt ein Schlusswort.

11.1 Zusammenfassung

Zu Beginn dieser Arbeit wird die Fachlichkeit des Katastrophenschutzes dargelegt. Mit Einführung der *KatApp* werden sowohl Informationsflüsse als auch Problematiken des analogen Prozesses analysiert. Es wird herausgearbeitet, inwiefern die *KatApp* die vorliegenden Prozesse optimiert und welche weiteren Chancen sich durch die Digitalisierung des Katastrophenschutzes ergeben. Aus der fachlichen Analyse der *KatApp* ergibt sich, dass Informationswege aufgrund der Digitalisierung des Prozesses durchaus effizienter gestaltet werden können. Eine Automatisierung der Informationsverteilung fördert eine schnelle und organisierte Zusammenarbeit der Hilfskräfte.

Anhand der aktuellen Implementierung der *KatApp* werden die Problematiken herausgearbeitet, welche durch die Umsetzung als mandantenfähiges System bestehen. Die logische Trennung der Daten nach Mandant verhindert dabei eine Zusammenarbeit mehrerer Organisationen.

Im Rahmen einer Analyse wissenschaftlicher Arbeiten werden typische mandantenfähige Architekturen identifiziert. Es wird eruiert, welche dieser Architekturen für die *KatApp* geeignet ist und diskutiert, warum die Architektur der *KatApp* entsprechend gewählt wurde. Da die *KatApp* primär für die Cloud entwickelt wird, werden weitere Arbeiten hinsichtlich medizinischer Anwendungen in der Cloud behandelt. Aufgrund der nötigen Kooperationsfähigkeit wird für die *KatApp* ein geteiltes Modell gewählt, welches durch logische Abschirmung eine Verletzung von Datenschutzrichtlinien verhindert. Zudem macht sich die *KatApp* Eigenschaften der Cloud wie hohe Skalierbarkeit und Datensicherheit zunutze.

Aus einem Interview mit Experten aus dem medizinischen sowie medizintechnischen Sektor werden die Anforderungen an die *KatApp* hinsichtlich einer Kooperation unter mehreren Mandanten abgeleitet. Diese Anforderungen legen die Grundlage für die anschließende Konzeption, in welcher eine Umsetzung und die damit einhergehenden Herausforderungen der Anforderungen diskutiert wird. Aus den Anforderungen ergibt sich die Einführung einer geobasierten Autorisierung, mit welcher eine sichere Kooperation unter Mandanten für die Dauer einer Katastrophe gewährleistet werden kann. Eine Einführung externer Dienste für Geokodierung ist unumgänglich.

Die Umsetzung der Konzeption wird in Form eines PoC durchgeführt, in welchem die zentralen Anpassungen, die an der *KatApp* durchgeführt werden müssen, dargestellt werden. Es werden vor allem die Unterschiede zur ursprünglichen Implementierung herausgearbeitet. Abschließend wird der entwickelte PoC einer Evaluierung unterzogen, in der die Funktionalität der Implementierung durch mehrere Testszenarien abgedeckt und geprüft wird. Der entwickelte PoC erweist sich als funktional, unter Berücksichtigung eines gesunden Optimierungspotenzials hinsichtlich der geobasierten Autorisierung für Einsatzleitungen.

11.2 Kritische Reflexion

Die Einführung geobasierter Autorisierung führt zu einer zunehmenden Divergenz zwischen Cloud-Bereitstellung und lokaler Bereitstellung der *KatApp*. Die Mandantenfähigkeit der *KatApp* erschwert die Verwendung der Software in lokalen Szenarien. Eine Nutzung ist lediglich unter der Voraussetzung präparierter Daten möglich. Unter Berücksichtigung der

Tatsache, dass ein mandantenfähiges System für den lokalen Einsatzzweck bereits als nicht zielführend zu betrachten ist, weist die Einführung geobasierter Autorisierung weiter in diese Richtung. In einem lokalen Szenario, in dem die *KatApp* in einem lokalen Netzwerk betrieben wird, erweist sich eine geobasierte Autorisierung zudem als wenig zweckdienlich. Die primäre Ausrichtung auf die Cloud macht die *KatApp* in einem lokalen Szenario kaum noch nutzbar. Und mit der Einführung weiterer Funktionen laufen der lokale und der Betrieb in der Cloud immer weiter auseinander.

Dies hat nicht nur Konsequenzen für die Nutzbarkeit der *KatApp* in lokalen Szenarien, sondern auch für die Wartbarkeit der Software. Mit jeder Einführung neuer Funktionen muss sorgfältig evaluiert werden, inwiefern sich diese auf eine lokale Bereitstellung auswirken. Im Laufe der Zeit wird eine Entscheidung getroffen werden müssen, ob die *KatApp* für den Cloud- oder den lokalen Betrieb weiterentwickelt werden soll, wobei eine simultane Entwicklung in beiden Bereichen zunehmend umständlicher bis hin zu unmöglich wird.

11.3 Fazit

Die Öffnung von Katastrophen für alle Mandanten stellt die Basis für eine Kooperation. Der Standort eines Ersthelfers ist dabei der ausschlaggebende Faktor, um an einer Katastrophe teilnehmen zu können. Durch die von einer Katastrophe gegebene Lokalität können Ersthelfer verschiedener Organisationen problemlos zusammenarbeiten. Ein Datenzugriff durch Einsatzleitungen oder Leitstellen ist ebenfalls durch Örtlichkeit oder Regionalität gegeben. Trotz der dadurch bereitgestellten Kooperationsfähigkeit bleiben Zugriffe auf die *KatApp* individuellen Mandanten zugeordnet. Dadurch kann auch weiterhin sichergestellt werden, dass Kosten, die durch die Nutzung des Systems entstehen, Organisationen präzise zugeordnet werden können.

Fachliche Konzepte und Fragstellungen werden durch die im Rahmen dieser Arbeit erhobenen Anforderungen geklärt. So zum Beispiel die Zuordnung von Verantwortlichkeiten, das Ende einer Katastrophe sowie der Datenzugriff nach Beendigung einer Katastrophe. Technische Herausforderungen wie die fachliche oder physische Trennung von Daten werden in einer technischen Diskussion evaluiert. Aufgrund der Katastrophendaten, welche in Echtzeit zwischen Mandanten geteilt werden müssen, erscheint eine physische Trennung der Daten der einzelnen Mandanten nicht zielführend.

Im Verlaufe der Arbeit hat sich zudem herausgestellt, dass der durch die *KatApp* gegebene Fall äußerst spezifisch ist. Trotz einer umfassenden Recherche konnten keine Arbeiten identifiziert werden, welche sich mit geobasierter Kooperation unter Mandanten befassen. Die für die *KatApp* angewandte Lösung scheint somit einzigartig, vor allem im Bereich der Medizin und des Katastrophenschutzes.

11.4 Ausblick

Auf Basis des PoC kann die *KatApp* weiterentwickelt werden. Dafür bietet es sich zunächst an, die bestehenden Probleme des PoC zu lösen. Dazu zählt speziell die Problematik mit der Autorisierung der Einsatzleitung, welche durch die Einführung einer bidirektionalen Kommunikation gelöst werden könnte.

Darüber hinaus könnte die *KatApp* hinsichtlich einer internationalen Nutzung untersucht werden. Dafür müsste evaluiert werden, inwiefern sich die Organisation des Katastrophenschutzes auf internationaler Ebene unterscheidet. Für eine Prüfung könnten Testszenarien in unterschiedlichen Ländern entworfen werden.

Des Weiteren ist eine Entscheidung zu treffen, ob die lokale Bereitstellung oder die Bereitstellung in der Cloud bevorzugt wird. Eine zentrale Entscheidung, welche die zukünftige Entwicklung der *KatApp* stark beeinflussen wird. Aufgrund der primären Ausrichtung des Systems auf die Cloud-Nutzung und seiner mandantenfähigen Struktur erscheint die Entscheidung für eine lokale Bereitstellung jedoch wenig ratsam, da dies womöglich zu einer Neuentwicklung des Systems führen würde.

11.5 Schlusswort

Die Aufgabenstellung dieser Arbeit befasst sich mit dem Entwurf eines Konzepts für die temporäre Kooperation unter Mandanten sowie der Umsetzung dieses Konzepts in einem PoC. Das zu erreichende Optimalziel ist die Implementierung eines PoC, welcher die fachlichen Abläufe des Katastrophenschutzes akkurat widerspiegelt. Das Resultat sollte die Einsatzkräfte in ihrer Arbeit unterstützen, nicht jedoch zu einer Beeinträchtigung ihrer Tätigkeit führen. Dies wird durch Automatisierung in zentralen Bereichen erreicht.

Für die Entwicklung des PoC werden die fachlichen Aspekte für eine Kooperation unter Mandanten identifiziert und in Anforderungen abgebildet, welche die Grundlage für die Konzeption und den implementierten PoC darstellen. Eine potenzielle Verbesserung des PoC besteht in seiner Verfeinerung sowie in der Festlegung von klaren Zielen für die Ausführungsumgebung der *KatApp*.

Temporäre Kooperation wird durch eine geobasierte Autorisierung erreicht, welche für die Dauer einer Katastrophe gültig ist. Der in dieser Arbeit entwickelte PoC ist funktional und bereit für den Einsatz.

Abbildungsverzeichnis

2.1 Ablauf einer Katastrophe [5] 6
2.2 Vereinfachte Darstellung des mSTaRT Triage-Algorithmus [4] 10
2.3 Funktionsweise der *KatApp* 12
2.4 *KatApp* Authentifizierungs-Hierarchie 15
2.5 *KatApp* Rollen & Rechte 16
3.1 Mandantenfähigkeit: Dediziertes Modell [9] 20
3.2 Mandantenfähigkeit: Isoliertes Modell 21
3.3 Mandantenfähigkeit: Geteiltes Modell 22
3.4 Mandantenfähigkeit: Hybrides Modell 22
5.1 *KatApp* Systementwurf 32
5.2 *KatApp* Architektur (Cloud) 35
5.3 *KatApp* Architektur (Lokal) 37
6.1 Datenmodell: Mandant 39
6.2 Datenmodell: Nutzer 40
6.3 Datenmodell: Rolle 41
6.4 Datenmodell: Katastrophe 41
6.5 Datenmodell: Patient 42
6.6 Datenmodell: Triage 42
6.7 Ablaufdiagramm: Erstellung einer Triage 43
6.8 Ablaufdiagramm: Verbindungsaufbau via *WebSocket* 45
8.1 Vereinfachte Darstellung der Verwaltungsgliederung Deutschlands [32] . . . 54
9.1 Datenmodell: Region 61
9.2 Ablaufdiagramm: Erstellung einer Triage (Neu) 62
9.3 Ablaufdiagramm: Verbindung via *WebSocket* (Neu) 64
10.1 Geobasierte Autorisierung 68

Glossar

API Application Programming Interface. 32, 44, 57, 58, 64, 69

ASB Arbeiter-Samariter-Bund. 2

AWS Amazon Web Services. 19, 27, 28, 30, 34–36, 38, 57

DRK Deutsches Rotes Kreuz. 2, 6, 50

DSGVO Datenschutz-Grundverordnung. 27

FaaS Function as a Service. 34–36

JSON JavaScript Object Notation. 68

JUH Johanniter-Unfall-Hilfe. 2

JWT JSON Web Token. 42

MHD Malteser Hilfsdienst. 2

PoC Proof of Concept. 3, 4, 53, 56–58, 61, 62, 65, 67, 69–71, 73–75

RBAC Role-Based Access Control. 16, 28

REST Representational State Transfer. 32, 35–37, 59, 64, 65, 69

THW Technisches Hilfswerk. 2, 6, 50

UI User Interface. 15, 23

Literaturverzeichnis

[1] Bundesamt für Bevölkerungsschutz und Katastrophenhilfe. *Triage-Sichtung*. 2024. URL: https://www.bbk.bund.de/DE/Themen/Gesundheitlicher-Bevoelkerungsschutz/Triage-Sichtung/triage-sichtung_node.html (besucht am 27. 08. 2024).

[2] Dr. Frank Antwerpes u. a. *Triage*. 2023. URL: https://flexikon.doccheck.com/de/Triage (besucht am 27. 08. 2024).

[3] Bundesamt für Bevölkerungsschutz und Katastrophenhilfe. *Vorsichtungsalgorithmus PRIOR*. 2024. URL: https://www.bbk.bund.de/DE/Themen/Gesundheitlicher-Bevoelkerungsschutz/Triage-Sichtung/Vorsichtungsalgorithmus-PRIOR/vorsichtungsalgorithmus-prior_node.html (besucht am 27. 08. 2024).

[4] Deutsches Rotes Kreuz. *mSTaRT Erwachsene*. 2024. URL: https://www.drk-ls.de/fileadmin/user_upload/downloads/downloadbereich/mstart_Algorithmus_Erwachsene.pdf (besucht am 02. 08. 2024).

[5] Deutsches Rotes Kreuz. *Durchführung und Auswertung von MANV-Übungen*. 2024. URL: https://www.drk.de/fileadmin/user_upload/Forschung/schriftenreihe/Band_3/Schriften_der_Forschung_3._Wissenschaftliche_Erkenntnisse_und_Best_Practices_.pdf (besucht am 27. 08. 2024).

[6] Bundesministerium des Innern und für Heimat. *Wer macht was beim Zivil- und Katastrophenschutz?* 2024. URL: https://www.bmi.bund.de/DE/themen/bevoelkerungsschutz/zivil-und-katastrophenschutz/gefahrenabwehr-und-katastrophenschutz/gefahrenabwehr-und-katastrophenschutz-node.html (besucht am 27. 08. 2024).

[7] Dr. No u. a. *MSTaRT-Schema*. 2024. URL: https://flexikon.doccheck.com/de/MSTaRT-Schema (besucht am 27. 08. 2024).

[8] Jaap Kabbedijk u. a. „Defining Multi-Tenancy: A Systematic Mapping Study on the Academic and the Industrial Perspective“. In: *Journal of Systems and Software* 100 (Okt. 2014). DOI: 10.1016/j.jss.2014.10.034.

[9] Adeniyi Abdul u. a. „Multi-tenancy Design Patterns in SaaS Applications: A Performance Evaluation Case Study“. In: *International Journal for Digital Society* 9 (März 2018), S. 1367–1375. DOI: 10.20533/ijds.2040.2570.2018.0168.

[10] Saurabh Singhal u. a. „A Disaster Management System Using Cloud Computing“. In: *2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*. 2023, S. 1–6. DOI: 10.1109/AICCSA59173.2023.10479284.

[11] Dimiter Velev und Plamena Zlateva. „Principles of Cloud Computing Application in Emergency Management“. In: Dez. 2011.

[12] Mohd Anwar und Ashiq Imran. „Access Control for Multi-tenancy in Cloud-Based Health Information Systems“. In: Nov. 2015. DOI: 10.1109/CSCloud.2015.95.

[13] Wayne Brown, Vince Anderson und Qing Tan. „Multitenancy - Security Risks and Countermeasures“. In: Sep. 2012, S. 7–13. ISBN: 978-1-4673-2331-4. DOI: 10.1109/NBiS.2012.142.

[14] Amazon Web Services Inc. *Datenschutz in Deutschland*. 2024. URL: https://aws.amazon.com/de/compliance/germany-data-protection/ (besucht am 09. 05. 2024).

[15] Amazon Web Services Inc. *DynamoDB encryption at rest*. 2024. URL: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/EncryptionAtRest.html (besucht am 09. 05. 2024).

[16] Marta Taggart, Bradley Roach und Patrick Woods. „Amazon Web Services: Risk and Compliance“. In: (Okt. 2021). URL: https://docs.aws.amazon.com/pdfs/whitepapers/latest/aws-risk-and-compliance/aws-risk-and-compliance.pdf.

[17] Qi Zhang, Lu Cheng und R. Boutaba. „Cloud Computing: State-of-the-art and Research Challenges“. In: *Journal of Internet Services and Applications* 1 (Mai 2010), S. 7–18. DOI: 10.1007/s13174-010-0007-6.

[18] Britta Rybicki. *Diese IT ist die Ursache für die digitale Katastrophe im deutschen Gesundheitssystem*. 2024. URL: https://www.handelsblatt.com/technik/it-internet/insight-innovation-diese-it-ist-die-ursache-fuer-die-digitale-katastrophe-im-deutschen-gesundheitssystem/100029539.html (besucht am 27. 08. 2024).

[19] Benedikt Ernst. *Das Gesundheitswesen muss seine IT modernisieren*. 2023. URL: https://www.healthcare-digital.de/das-gesundheitswesen-muss-seine-it-modernisieren-a-84b64786f63becd461eeaaa6d76374c4/ (besucht am 27. 08. 2024).

[20] Heike E. *Universitätsklinika: Die IT-Infrastruktur veraltet*. 2023. URL: https://www.aerzteblatt.de/archiv/160986/Universitaetsklinika-Die-IT-Infrastruktur-veraltet (besucht am 27. 08. 2024).

[21] Inc. Amazon Web Services. *DSGVO-Compliance bei der Nutzung von AWS-Services*. 2024. URL: https://aws.amazon.com/de/compliance/gdpr-center/ (besucht am 27. 08. 2024).

[22] Microsoft Corporation. *Datenschutz in Azure*. 2024. URL: https://azure.microsoft.com/de-de/explore/trusted-cloud/privacy (besucht am 27. 08. 2024).

[23] intersoft consulting. *Datenschutz-Grundverordnung*. 2024. URL: https://dsgvo-gesetz.de/ (besucht am 27. 08. 2024).

[24] T4 Labs Inc. *Cloud Computing Market Share*. 2021. URL: https://www.t4.ai/industry/cloud-computing-market-share (besucht am 27. 08. 2024).

[25] Statista GmbH. *Umsatz mit Cloud Computing weltweit von 2010 bis 2023 und Prognose bis 2025*. 2024. URL: https://de.statista.com/statistik/daten/studie/195760/umfrage/umsatz-mit-cloud-computing-weltweit/ (besucht am 17. 06. 2024).

[26] Precedence Research. *Serverless Architecture Market Size, Share, and Trends 2024 to 2033*. 2024. URL: https://www.precedenceresearch.com/serverless-architecture-market (besucht am 27. 08. 2024).

[27] Johannes Manner. „A Structured Literature Review Approach to Define Serverless Computing and Function as a Service“. In: Juli 2023. DOI: 10.1109/CLOUD60044.2023.00068.

[28] Amazon Web Services Inc. *Amazon API Gateway – Service-Level-Vereinbarung*. 2024. URL: https://aws.amazon.com/de/api-gateway/sla/ (besucht am 01. 04. 2024).

[29] Amazon Web Services Inc. *AWS Lambda Service Level Agreement*. 2024. URL: https://aws.amazon.com/de/lambda/sla/ (besucht am 01. 04. 2024).

[30] Amazon Web Services Inc. *Amazon DynamoDB Service Level Agreement*. 2024. URL: https://aws.amazon.com/de/dynamodb/sla/ (besucht am 01. 04. 2024).

[31] The Kubernetes Authors. *kubernetes/kubernetes*. 2024. URL: https://github.com/kubernetes/kubernetes (besucht am 25. 08. 2024).

[32] Bundesagentur für Arbeit. *Politisch-administrative Gebietsstruktur*. 2024. URL: https://statistik.arbeitsagentur.de/DE/Statischer-Content/Grundlagen/Klassifikationen/Regionale-Gliederungen/Politische-Gebietsstruktur/Politische-Gebietsstruktur-XLS.xlsx;jsessionid=FC72287CC2E1A9B215D6CAE7EE690B7B?__blob=publicationFile&v=40 (besucht am 27. 08. 2024).

[33] Sofia Emelianova. *Standorte | Chrome DevTools*. 2023. URL: https://developer.chrome.com/docs/devtools/settings/locations?hl=de (besucht am 26. 08. 2024).

[34] ip-api. *Frequently asked questions*. 2024. URL: https://members.ip-api.com/faq (besucht am 27. 08. 2024).

[35] Google LLC. *Nutzungsbasierte Preisgestaltung*. 2024. URL: https://mapsplatform.google.com/pricing/ (besucht am 26. 08. 2024).

[36] ip-api. *Frequently Asked Questions*. 2024. URL: https://ipapi.com/faq (besucht am 27. 08. 2024).

Eidesstattliche Erklärung

Ich versichere, dass ich diese Thesis/Arbeit ohne fremde Hilfe selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlichen oder sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet habe. Die Arbeit wurde noch keiner Kommission zur Prüfung vorgelegt und verletzt in keiner Weise Rechte Dritter.

Esslingen, den 31. August 2024: _____

Pascal Nerger
IT-Designers GmbH, Esslingen am Neckar