University of Applied Sciences
Ravensburg Weingarten

Department of Electrical Engineering and
Computer Science

Systemadministration Project

Patient Monitoring System

Author:
Niklas Kleiser
Mat.No. 35579
Emircan Tutar
Mat.No. 35606
David Metzler
Mat.No. 35582

Supervisor:
M.Sc. Benjamin Stähle

4. Dezember 2025

Inhaltsverzeichnis

1 Einleitung

Die fortschreitende Digitalisierung und technologische Entwicklung bieten große Chancen, die Pflegequalität in medizinischen Einrichtungen zu verbessern. Besonders in der intensiven Patientenüberwachung, wo präzise und zeitnahe Informationen über den Zustand der Patienten lebenswichtig sind, kann der Einsatz moderner Technologie einen entscheidenden Unterschied machen. Dieses Projekt zielt darauf ab, ein Überwachungssystem zu entwickeln, das speziell für die Bedürfnisse der Pflegekräfte und Patienten in Krankenhäusern konzipiert wurde.

In der Praxis wird die Pflegequalität durch die Verfügbarkeit und Genauigkeit der überwachten Daten direkt beeinflusst. Hier setzt unser System an, indem es innovative Technologien nutzt, um eine kontinuierliche, präzise Überwachung zu gewährleisten. Durch die Verwendung von Docker und Raspberry Pi wird eine sichere und isolierte Umgebung geschaffen, in der Anwendungen stabil laufen und leicht skaliert werden können. Dies ist besonders wichtig in der dynamischen Umgebung einer medizinischen Einrichtung, wo Anpassungsfähigkeit und Zuverlässigkeit entscheidend sind.

Ubuntu 22.04 LTS bietet eine solide Grundlage für unser System, da es langfristige Unterstützung und regelmäßige Updates gewährleistet. Zusammen mit der Implementierung von dynamischen DNS-Diensten und Fail2Ban schaffen wir eine sichere Infrastruktur, die den Datenschutz und die Integrität kritischer Patientendaten schützt. Die Integration von Kommunikationstechnologien wie MQTT und Matrix ermöglicht es Pflegekräften, in Echtzeit auf kritische Situationen zu reagieren und verbessert die Koordination innerhalb des Pflegeteams.

Durch den Einsatz dieser Technologien wird nicht nur die Effizienz der Pflegekräfte gesteigert, sondern auch die Sicherheit und das Wohlbefinden der Patienten signifikant verbessert. Dieses Projekt zeigt, wie durch den Einsatz moderner Technologien die Lebensqualität von Menschen verbessert und die Arbeit von medizinischem Personal unterstützt werden kann. Die dokumentierten Abschnitte bieten eine detaillierte Darstellung der technischen Anforderungen, der Architektur sowie der tatsächlichen Implementierung des Systems und motiviert alle Beteiligten, auf diesen Grundlagen weiter aufzubauen und die Technologie zum Wohl der Gesellschaft einzusetzen.

1.1 Motivation

Die Überwachung von Patienten in Krankenhäusern ist eine entscheidende Aufgabe, die hohe Anforderungen an Präzision und Zuverlässigkeit stellt. Traditionelle Überwachungsmethoden sind oft personalintensiv und können nicht immer die notwendige kontinuierliche Aufmerksamkeit gewährleisten. Die Integration moderner Technologien wie Videoüberwachung und Computer Vision bietet die Möglichkeit, diese Prozesse zu automatisieren und zu verbessern, um eine kontinuierliche und effektive Überwachung sicherzustellen. Dies ist besonders wichtig in Intensivstationen, wo unvorhersehbare und schnell eintretende Ereignisse rasch erkannt und adressiert werden müssen.

1.2 Problemstellung

Trotz des Einsatzes moderner Überwachungstechnologien in Krankenhäusern sind viele Systeme immer noch nicht in der Lage, kritische Situationen wie Stürze, ungewöhnliche Bewegungen oder medizinische Notfälle automatisch zu erkennen. Zudem führen Datenschutzbedenken und die Sorge um die Wahrung der Privatsphäre der Patienten zu zusätzlichen Herausforderungen bei der Implementierung von Überwachungslösungen, die Kameras und andere Überwachungsgeräte nutzen.

1.3 Zielsetzung

Das Ziel dieses Projekts ist die Entwicklung eines intelligenten Patientenüberwachungssystems, das mithilfe von Kameras und Computer Vision-Technologien eine kontinuierliche und automatisierte Überwachung bietet. Das System soll in der Lage sein, kritische Situationen wie Stürze oder medizinische Notfälle automatisch zu erkennen und sofort Alarm zu schlagen. Weiterhin soll das System datenschutzkonform gestaltet sein und die Privatsphäre der Patienten respektieren, indem es beispielsweise die Möglichkeit bietet, Überwachungsbereiche gezielt zu anonymisieren oder zu deaktivieren.

2 Lastenheft

2.1 Muss - Anforderungen

2.1.1 Docker

Docker ist ein Tool, das Anwendungen in kompakten, tragbaren Container organisiert und ausführt. Diese Container sind von der Umgebung isoliert, in der sie laufen, was bedeutet, dass sie konsistent funktionieren, unabhängig davon, wo sie eingesetzt werden. Für dieses Self-Hosted Projekt auf einem Raspberry Pi bietet Docker die Möglichkeit, verschiedene Dienste und Anwendungen ohne Konflikte nebeneinander auszuführen.
Durch die Nutzung von Docker kann man sicherstellen, dass jede Anwendung und jeder Service seine eigene Umgebung hat, was die Sicherheit verbessert und es einfacher macht, Updates und Änderungen durchzuführen, ohne andere Teile des Systems zu beeinträchtigen. Docker ist besonders wertvoll, wenn man mehrere unterschiedliche Anwendungen auf einem einzigen Raspberry Pi hosten möchte, da es die Verwaltung deutlich vereinfacht[10].

2.1.2 Ubuntu 22.04 (LTS)

Ubuntu 22.04 LTS, ist einer der beliebten Linux-Betriebssystemen. Sie bietet langfristige Unterstützung bis April 2027, was sie zu einer zuverlässigen Wahl für das Patient Monitoring System Projekt macht. Diese Version ist benutzerfreundlich und ideal für eine Vielzahl von Anwendungen, von Lernen und Experimentieren bis hin zu ernsthaften technischen Projekten [34].

2.1.3 Dynamisches DNS-Management (ddclient)

Der DDclient ist ein Tool, das speziell für die Aktualisierung dynamischer DNS-Einträge entwickelt wurde. Es ermöglicht es Geräten wie dem Raspberry Pi, die oft keine feste IP-Adresse haben, dennoch unter einem konstanten Domainnamen erreichbar zu sein. Durch die automatische Überwachung und Aktualisierung der sich ändernden öffentlichen IP-Adresse des Raspberry Pi sorgt DDclient dafür, dass der Pi auch nach einem IP-Wechsel durch den Internetdienstanbieter zugänglich bleibt. Diese Funktion ist besonders wertvoll für die Verwendung des Raspberry Pi als Home-Server für Anwendungen wie Webhosting, Dateifreigaben oder Überwachungsdienste. Die Installation und Konfiguration von DDclient ist unkompliziert und erfolgt über die Kommandozeile [8].

2.1.4 Fail2ban

Fail2Ban ist ein entscheidendes Werkzeug zur Absicherung, die dazu dient, einen Server vor unautorisierten Zugriffsversuchen und Brute-Force-Angriffe zu schützen. Durch die Überwachung von Logdateien erkennt Fail2Ban auffällige

Anmeldeversuche und blockiert die zugehörigen IP-Adressen automatisch für eine vordefinierte Zeitdauer. Mit Fail2Ban kann beispielsweise die IP-Adresse eines Geräts, das wiederholt versucht, ohne die korrekten Anmeldedaten eine Verbindung zum Server herzustellen, blockiert werden. Es lässt sich so konfigurieren, dass jede IP-Adresse, die mehr als dreimal pro Tag versucht, sich zu verbinden, gesperrt wird [33]. Diese Funktion ist besonders wichtig für die Minimierung von potenziellen Sicherheitsrisiken. Im Rahmen unseres Projekts auf dem Raspberry Pi wird Fail2Ban eine kritische Rolle spielen, indem es dazu beiträgt, die Zugriffskontrolle auf unsere Dienste wie SSH (Secure Shell) verstärken.

2.1.5 MQTT / MQTT-Client

Das System besteht aus mehreren verschiedenen Komponenten, die miteinander kommunizieren. Diese Kommunikation erfolgt über einen Message-Broker und das Protokoll "MQTT" [22]. Um das Protokoll nutzen zu können, benötigt es einen MQTT-Message Broker und Clients, die ihre Nachrichten an diesen schicken, damit sie an die Clients verteilt werden, die darum gebeten haben, Nachrichten eines Topics zu erhalten.

2.1.6 SMTP-Server / SMTP-Client

Zu Zwecken der Protokollierung und Dokumentation sendet das System zusätzlich zu einem Zustand Alarm eine Mail an einen eigenen Mail-Server. Dieser Mail-Server empfängt über das SMTP-Protokoll Mails. Dafür wird ein Mailcow-Server eingerichtet [17]. Hierfür müssen auch DNS-Einträge angepasst werden.

2.1.7 Raspberry und Raspberry Pi

Für das Projekt wird der Raspberry Pi gewählt, da dieser eine einfache Anbindung mit einer Kamera erlaubt [29] [30]. Der Raspberry Pi ermöglicht es, Microcontroller und Kamera in einem kompakten Gehäuse unterzubringen. Es gibt zwei Raspberry Pis, die eine Detektion durchführen (siehe Abb. 1). Ein Raspberry Pi mit Raspberry Pi Kamera übernimmt die Bett-Detection und ein weiterer Raspberry Pi mit Kamera übernimmt die Fall-Detection Ein dritter Raspberry Pi wird verwendet, um einen Zustand Alarm zu schalten.

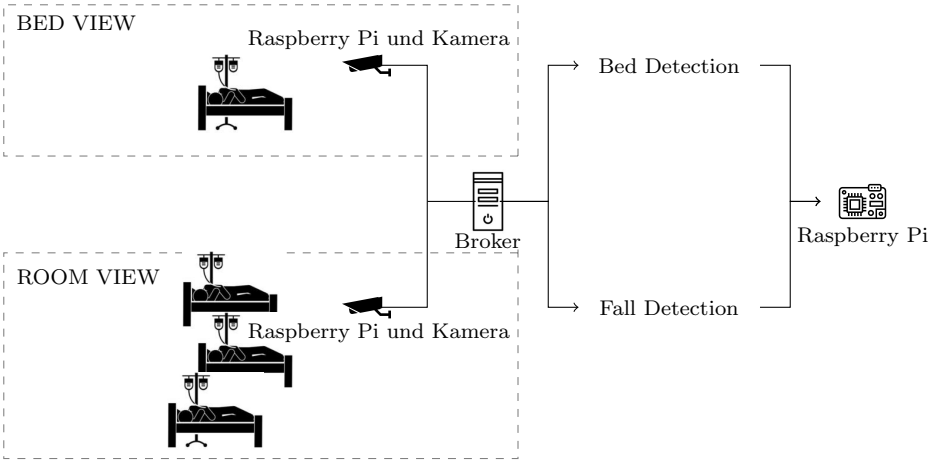

Abbildung 1: Darstellung des Systemaufbaus

2.1.8 Fall Detektion

Wie bereits in Abschnitt 2.1.7 erwähnt, führt ein Raspberry Pi mithilfe der Raspberry Pi Kamera die Fall-Detection durch. Der Raspberry Pi überprüft, ob ein Patient hingefallen ist, was mithilfe des YOLO-Frameworks realisiert wird [40]. Ziel ist es, das Modell auf dem Raspberry Pi zu implementieren, um den Bedarf an zusätzlichen Servern zu vermeiden.

2.1.9 Matrix

Ein Matrix-Server ist vorhanden [20]. Über diesen Server werden Pfleger auch auf dem Handy benachrichtigt, wenn eine Alarmsituation eintritt. Dieser Kommunikationsdienst ermöglicht eine nahtlose und sichere Echtzeitkommunikation über verschiedene Geräte. Die Verwendung des Matrix-Protokolls garantiert dabei die Ende-zu-Ende-Verschlüsselung aller übertragenen Daten, was die Vertraulichkeit und Sicherheit der sensiblen Patienteninformationen stärkt.

2.2 Soll - Anforderungen

2.2.1 Bett Detektion

Es soll eine Komponente im System geben, welche sich darum kümmert, zu erkennen, ob eine Person in einem Bett liegt. Es soll aber nicht jede Abwesenheit sofort zu einem Zustand Alarm führen, sondern nur, wenn über einen Zeitraum von 10 Minuten, das Bett leer bleibt, sollte ein Pfleger benachrichtigt werden, der dann nach dem Patienten schaut. Aufgrund der Zeitspanne von 10 Minuten, die das Bett leer bleiben muss, um einen Zustand Alarm auszulösen, ist es auch in Ordnung, wenn die Detektion etwas länger dauert.

Es soll angestrebt werden, innerhalb von 30 Sekunden nach Verlassen des Betts zu erkennen das dieses leer ist.

2.2.2 Alarm (Ton und Licht)

Eine wesentliche Anforderung des Projekts ist die Implementierung eines Alarmtons und eines Lichtsignals. Diese Funktionalität wird durch den Anschluss eines Lautsprechers und einer LED an den Raspberry Pi realisiert. Die Verbindung erfolgt über ein Steckbrett, auf dem auch die erforderlichen Widerstände und Transistoren montiert sind, um die Signale entsprechend zu steuern und zu verstärken.
Die LED dient dabei als visuelles Alarmzeichen, während der Lautsprecher akustische Warnungen ausgibt. Diese Komponenten werden über GPIO-Pins des Raspberry Pi gesteuert, was eine direkte Signalübertragung von der Steuerungssoftware zu den Warnsystemen ermöglicht.

2.2.3 Firewall

Eine Firewall ist als zusätzliche Sicherheitsebene im Projekt vorgesehen, um den Schutz des Raspberry Pi-Systems zu verstärken. Obwohl viele Heimrouter bereits mit integrierten Firewall ausgestattet sind, ist die Implementierung einer dedizierten Firewall auf dem Raspberry Pi selbst eine wichtige Soll-Anforderung. Diese Maßnahme dient dazu, die Netzwerksicherheit weiter zu erhöhen und Anwendungsspezifische Sicherheitsrichtlinien durchzusetzen.
Die Firewall auf dem Raspberry Pi wird konfiguriert, um eingehende und ausgehende Verbindungen genau zu überwachen und zu kontrollieren. Sie filtert den Datenverkehr basierend auf vordefinierten Sicherheitsregeln und hilft dabei, unerwünschten Zugriff sowie mögliche Bedrohungen abzuwehren. Diese Schutzschicht ist besonders kritisch, wenn der Raspberry Pi in einem öffentlich zugänglichen Netzwerk betrieben wird oder sensible Daten verarbeitet.

2.3 Kann - Anforderungen

2.3.1 MQTT Frontend

Es kann ein MQTT - Frontend benutzt werden, um es Entwicklern einfacher zu machen die verschiedenen Nachrichten und Kommunikationskanäle nachzuvollziehen.

2.3.2 Gesprochene Information über Patient in Hilfesituation

In den Minimalanforderungen dieses Systems weiß ein Pfleger nur das ein Patient Hilfe benötigt, nicht aber, welcher der Patienten. Um diese Proble-

matik zu lösen, kann das System anstelle des Alarmtons über eine sprechende Stimme dem Pfleger mitteilen, welcher Patient Hilfe benötigt.

2.3.3 Kameraview

Zusätzlich lässt sich mithilfe des QT-Frameworks eine Kameraview-Anwendung in Python entwickeln [26] [28]. Diese Anwendung ermöglicht es Pflegepersonal, die Patienten besser zu überwachen und auch solche Vorfälle zu erkennen, die durch die automatischen Detektoren möglicherweise nicht erfasst werden. Insbesondere kann ein Pfleger mithilfe dieser Kameraview die Extubation eines Patienten frühzeitig bemerken und entsprechend reagieren.

3 Hardware

3.1 Alarm-Hardware-System

Das Alarm-Hardware-System besteht aus einem Raspberry Pi, der als zentrale Steuereinheit fungiert. An den GPIO-Pins des Raspberry Pi sind LEDs (eine gelbe, eine grüne und eine rote), ein Piezolautsprecher (Buzzer) und ein Taster angeschlossen, um sowohl visuelle als auch akustische Alarme zu erzeugen. Dieses System wird verwendet, um das Pflegepersonal sofort auf kritische Situationen wie Stürze oder die Abwesenheit eines Patienten aus dem Bett aufmerksam zu machen. Zusätzlich sind Widerstände auf einem Steckbrett montiert, um die Signale zu steuern und zu verstärken. Der Taster dient dazu, den Buzzer stummzuschalten, wenn die rote LED leuchtet.

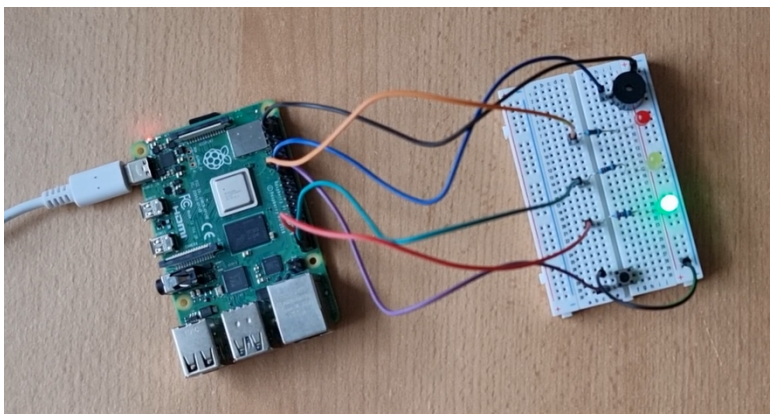

Abbildung 2: Raspberry Pi mit angeschlossenem Alarm-Hardware-System (grün)

Im normalen Betriebszustand, wenn alles in Ordnung ist, leuchtet die grüne LED (siehe Abbildung 2). Bei einer Fall-Detection, wenn ein Patient gestürzt

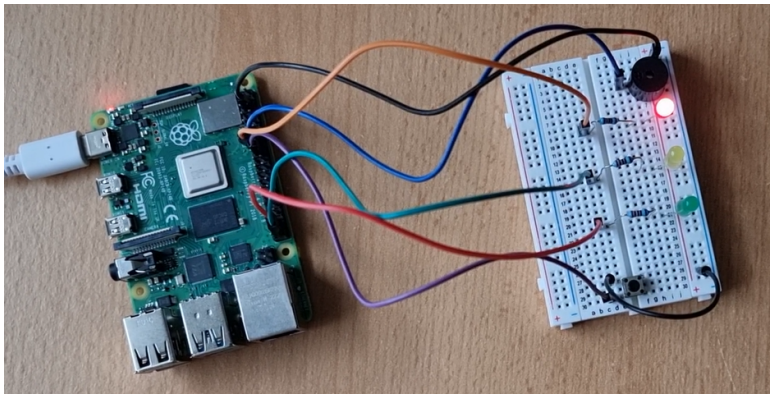

Abbildung 3: Raspberry Pi mit angeschlossenem Alarm-Hardware-System (rot)

ist, leuchtet die rote LED und der Buzzer gibt ein akustisches Signal aus. Der Taster kann verwendet werden, um den Buzzer stummzuschalten (siehe Abbildung 3). Die gelbe LED kann für weitere Signalisierungen verwendet werden.

3.2 Raspberry Pi

Der Raspberry Pi wird für mehrere Aufgaben im System verwendet. Ein Raspberry Pi ist für die Bett-Detection zuständig und ein weiterer für die Alarmausgabe. Diese Einplatinencomputer sind aufgrund ihrer Vielseitigkeit, geringen Kosten und der Möglichkeit, sie mit verschiedenen Sensoren und Kameras zu verbinden, ideal für das Projekt geeignet. Ein Raspberry Pi ist mit einer Kamera ausgestattet, um die entsprechenden Erkennungsaufgaben durchzuführen. Die Kamera wird strategisch positioniert, um eine optimale Überwachung der Patienten zu gewährleisten.

3.3 Raspberry Pi Kamera

Die Raspberry Pi Kamera ist ein wesentlicher Bestandteil des Überwachungssystems. Sie wird für die Fall-Detection eingesetzt. Die Kamera erfasst kontinuierlich Videodaten, die von den auf dem Raspberry Pi laufenden Algorithmen analysiert werden. Diese Algorithmen, basierend auf dem YOLOv5-Modell, erkennen kritische Ereignisse wie Stürze aus dem Bett und lösen entsprechende Alarme aus.

3.4 Atomic-Pi

Der Atomic-Pi hat einen x86-Prozessor. Er ist leistungsfähiger als normale Raspberry - Pis, hat aber den Nachteil, dass weniger RAM verbaut ist. Der Atomic-Pi wird unseren Mailserver, Message Broker und den Mail-Client

hosten. Ein Atomic-Pi hat wie der Raspberry Pi auch, viele GPIO Pins, die man adressieren kann. Ein Unterschied ist hingegen das der Atomic-Pi eine dedizierte GPU verbaut hat, die aber leider nicht für KI - Modell - Inferenz verwendet werden kann.

4 Kommunikation

4.1 Einführung in MQTT

MQTT (Message Queuing Telemetry Transport) ist ein leichtgewichtiges Protokoll, das speziell für die Kommunikation zwischen Geräten mit begrenzten Ressourcen entwickelt wird. Es zeichnet sich durch seine Effizienz und geringe Bandbreitennutzung aus, was es ideal für den Einsatz in IoT (Internet of Things) und anderen Umgebungen mit eingeschränkten Netzwerkressourcen macht.

4.2 MQTT - Architektur

MQTT arbeitet nach einem Publisher/Subscriber-Modell, bei dem Clients Nachrichten zu bestimmten Themen (Topics") veröffentlichen oder abonnieren. Ein zentraler Broker verwaltet diese Kommunikation, indem er die Nachrichten vom Publisher an die abonnierten Subscriber weiterleitet. Es muss gewährleistet sein, dass dieser Broker funktioniert und erreichbar ist, da ohne ihn keine Kommunikation zwischen Subscriber und Publisher möglich ist.

4.3 Sicherheitsaspekt

Authentifizierung stellt sicher, dass nur autorisierte Geräte Zugang zum Broker erhalten. Der Message-Broker erlaubt nur Kommunikation mit authentifizierten Publishern und Subscribern. Nicht authentifizierte Clients werden abgelehnt. Die Verschlüsselung der Kommunikation zwischen Clients und dem Broker erfolgt über TLS (Transport Layer Security). Dies schützt die Daten vor Abhören und Manipulation durch Dritte.

4.4 Setup des Message-Brokers

Um über das MQTT-Protokoll zu kommunizieren, ist der erste Schritt die Installation und Konfiguration eines MQTT-Brokers. Bekannte MQTT-Broker sind Mosquitto und HiveMQ. Der Eclipse-Mosquitto-Docker-Container wird verwendet. Der Message-Broker wird nun über die mosquitto.conf konfiguriert, sodass nur authentifizierte Clients mit dem Broker kommunizieren dürfen. Ein Account wird erstellt, dessen Zugangsdaten Clients zur Authentifizierung am Broker verwenden müssen.

4.5 Integration mit anderen Komponenten

Die Integration mit anderen Komponenten des Systems ist einfach, da MQTT ein weitverbreitetes Protokoll ist und in vielen Sprachen Implementationen eines MQTT-Clients existieren. Über diesen können Nachrichten an den Message-Broker und somit an andere Clients gesendet werden. Es muss

nur noch festgelegt werden, wie die Nachrichten aussehen. Da die Nachrichtenformate den speziellen Bedingungen und Anforderungen der Kommunikationspartner unterliegen, wird nicht darauf festgelegt, dass dieselben Nachrichten-Inhalte im ganzen System gleich aussehen.

5 Implementierung

5.1 Sicherheit

Die Sicherheit des Systems wird durch verschiedene Maßnahmen gewährleistet. Dazu gehören die Implementierung von Fail2Ban zur Verhinderung von Brute-Force-Angriffen, die Nutzung von DDclient zur dynamischen DNS-Aktualisierung und die Konfiguration der Uncomplicated Firewall (UFW) zur Verwaltung von Netzwerkanfragen. Diese Maßnahmen tragen dazu bei, das System vor unbefugtem Zugriff und anderen potenziellen Sicherheitsbedrohungen zu schützen.

5.1.1 Fail2Ban

Im Rahmen dieses Projekts wird Fail2Ban so konfiguriert, dass bei mehr als fünf fehlgeschlagenen SSH-Anmeldeversuchen die IP-Adresse des Geräts, von dem aus die Anmeldeversuche unternommen werden, dauerhaft gesperrt wird. Diese Sperre bleibt bestehen, bis der Administrator die IP-Adresse manuell wieder entsperrt. Diese strenge Maßnahme dient dazu, die Zugriffskontrolle auf unseren Raspberry Pi zu verstärken und potenzielle Sicherheitsrisiken zu minimieren.
Die Konfigurationsdatei für Fail2Ban (`jail.local`) wird wie folgt angepasst:

```
[sshd]
enabled  = true
port     = ssh
logpath  = /var/log/auth.log
maxretry = 5
bantime  = -1
```

Diese Einstellungen sorgen dafür, dass Fail2Ban das SSH-Protokoll überwacht und nach fünf fehlgeschlagenen Versuchen die IP-Adresse dauerhaft sperrt. Der Parameter `bantime = -1` stellt sicher, dass die Sperre unendlich lang andauert, bis der Administrator eingreift.
Durch die Implementierung von Fail2Ban wird die Sicherheit des Systems erheblich verbessert, da automatisierte Brute-Force-Angriffe effektiv abgewehrt werden können.

5.1.2 DDclient

DDclient ist ein Perl-basiertes Programm, das dynamische DNS-Dienste (DDNS) unterstützt. Es wird verwendet, um die IP-Adresse eines Systems bei einem DNS-Dienstanbieter zu aktualisieren, wenn sich die IP-Adresse des Systems ändert. Im Rahmen dieses Projekts wird DDclient auf dem Atomic Pi verwendet, der den MQTT-Dienst betreibt. Dadurch wird sichergestellt, dass das Alarmsystem immer unter einem festen Domainnamen erreichbar ist, auch wenn sich die öffentliche IP-Adresse des Netzwerks ändert.Dies ist besonders wichtig für die ständige Kommunikation und Überwachung des Alarmsystems [8].

5.1.3 UFW - Uncomplicated Firewall

Zur Erhöhung der Sicherheit auf dem Raspberry Pi wird UFW (Uncomplicated Firewall) installiert und konfiguriert. Dabei wird sichergestellt, dass notwendige Dienste erreichbar sind, während andere Verbindungen blockiert werden.

Zuerst erfolgt die Installation von UFW, um die grundlegenden Sicherheitsmaßnahmen einzurichten. Die Standardregeln werden so gesetzt, dass eingehender Netzwerkverkehr standardmäßig blockiert wird, es sei denn, es gibt explizite Regeln, die bestimmte Arten von eingehendem Verkehr erlauben. Diese Maßnahme stellt eine grundlegende Sicherheitsmaßnahme dar, um ungewollten oder schädlichen Datenverkehr abzuwehren.

Anschließend werden spezifische Verbindungen erlaubt. Zunächst wird der SSH-Zugang freigegeben, um die Fernverwaltung des Systems zu ermöglichen. Dann werden die Ports für HTTP und HTTPS geöffnet, um den Zugang zu Webdiensten zu gestatten.

Darüber hinaus werden Docker-spezifische Ports freigeschaltet, um die Kommunikation von Docker-Containern zu ermöglichen. Ebenso werden die MQTT-Ports geöffnet, die für die Kommunikation innerhalb des Systems notwendig sind. Schließlich werden die Netzwerkmanagement-Ports freigegeben, um die grundlegenden Netzwerkdienste zu gewährleisten.

Nach der Konfiguration der Regeln wird UFW aktiviert und der Status überprüft, um sicherzustellen, dass die Einstellungen korrekt angewendet wurden. Diese Konfiguration von UFW stellt sicher, dass nur autorisierte Verbindungen zu den notwendigen Diensten auf dem Raspberry Pi zugelassen werden, wodurch die Sicherheit des Systems erhöht wird.

5.2 Einsatz von Docker

Für alle Komponenten des verteilten Systems kam Docker zum Einsatz. Diese Technologie wurde gewählt, weil Docker Anwendungen in Container isoliert, die alle notwendigen Abhängigkeiten und Konfigurationen enthalten. Dadurch konnten die Komponenten auf den verschiedenen Geräten problemlos betrieben und getestet werden [7].

5.2.1 Docker Compose

Zusätzlich wurde Docker Compose verwendet, um mehrere Container als Teil eines einzigen Anwendungsstapels zu verwalten. In einer YAML-Datei (docker-compose.yml) werden alle benötigten Services, Netzwerke und Volumes definiert. Dies ermöglicht es, alle für einen Service erforderlichen Container mit einem einzigen Befehl zu starten, anstatt jeden Container einzeln konfigurieren und starten zu müssen [11].

5.2.2 Environment-Datei

Um sicherzustellen, dass keine vertraulichen Informationen fest im Code verankert werden, wird eine Environment-Datei (.env) verwendet. Diese Datei erleichtert die Verwaltung der Umgebungsvariablen, welche von Docker Compose eingelesen und in den Containern verfügbar gemacht werden [32].

5.3 YOLOv5

YOLO, ein Bilderkennungsmodell, wurde von Joseph Redmon und Ali Farhadi an der Universität von Washington entwickelt und im Jahr 2015 veröffentlicht. YOLO steht für "You Only Look Once", was darauf hinweist, dass das Modell nur einmal auf ein Bild schaut, um Objekte schnell und präzise zu erkennen. Seit seiner Veröffentlichung wurden insgesamt neun Versionen von YOLO veröffentlicht [40] .

5.3.1 Das Modell

Ein vortrainiertes Modell der fünften Version von YOLO wird verwendet, welches bereits spezifisch für die gegebenen Anforderungen trainiert ist. Obwohl es sich um eine ältere Version von YOLO handelt, ist die Effektivität des Modells nicht beeinträchtigt. YOLOv5 zeigt gerade bei begrenzten Hardware-Ressourcen wie dem Raspberry Pi 4 eine höhere Geschwindigkeit im Vergleich zu neueren Versionen wie YOLOv8. Vorteilhaft ist auch, dass eine kleinere Variante von YOLO, nämlich Yolov5s, genutzt wird. Dieses Modell wurde mit Bildern aus diversen Quellen trainiert, wobei ein benutzerdefiniertes Datenset aus 374 Trainingsbildern und 111 Validierungsbildern erstellt wurde.

5.3.2 Makesense.ai

Für das Labeln der Daten kommt Makesense.ai zum Einsatz, eine Plattform, die es Nutzern erlaubt, Bilder hochzuladen und mit Annotationen wie Bounding Boxes und Klassifizierungen zu versehen. Dies dient der Vorbereitung von Datensätzen für Anwendungen in der Computer Vision [19]. Die Daten sind dabei in drei Klassen untergliedert: 'walking', 'sitting' und 'fall detected' (siehe Abbildung 4).

Klasse: "walking"

Klasse: "sitting"

Klasse: "fall detected"

Abbildung 4: YOLOv5 Detektion Klassen

5.3.3 Anbindung

Die Anbindung an das YOLOv5-Modell erfolgt über PyTorch, ein Open-Source-Framework für maschinelles Lernen, das auf der Programmiersprache Python und der Torch-Bibliothek basiert. PyTorch wurde 2016 von einem Forscherteam für künstliche Intelligenz bei Facebook entwickelt [27].
Das Modell läuft in einem Docker-Container, der kontinuierlich über MQTT einen int-Wert veröffentlicht, um anzuzeigen, ob der Zustand Zustand Normal, Zustand Warnung oder Zustand Alarm aufgetreten ist. Zur Einbindung der Raspberry Pi Kamera verwendet der Container auch OpenCV, eine Open-Source-Bibliothek für Computer Vision [1]. Mithilfe von OpenCV wird auch eine Funktion entwickelt, die über MQTT ein Bild mit allen Detektionen versendet. Diese Bilder werden später mithilfe eines weiteren Docker Containers auf einem lokalen Rechner angezeigt.

5.4 Matrix

Matrix ist einen offener Standard für interoperable, dezentrale, Echtzeitkommunikation über das Internetprotokoll (IP). Nutzer verbinden sich mit einem Heim-Server und können Räume auf jedem Matrix-Server betreten, was die Kommunikation über Servergrenzen hinweg ermöglicht. Nachrichten werden zwischen den Servern synchronisiert, was eine unterbrechungsfreie Kommunikation sicherstellt, auch wenn ein Server offline geht. Matrix unterstützt Ende-zu-Ende-Verschlüsselung für sichere Gespräche. Zur Erprobung von Matrix lassen sich zahlreiche Matrix-Clients verwenden [14]. Ein

Docker Container wird eingesetzt, um den Matrix Home-Server zu hosten [21]. Matrix dient dazu, Pfleger mittels eines Zustand Alarm zu informieren.

5.4.1 Element

Der Matrix-Client Element wird für die Kommunikation genutzt [12]. Element, geleitet von den Entwicklern von Matrix, bietet eine benutzerfreundliche Oberfläche sowie zahlreiche Funktionen, die den spezifischen Anforderungen entsprechen.

5.4.2 Nginx reverse proxy

Ein Reverse-Proxy wird benötigt, damit der Matrix-Server auch SSL nutzen kann. Ein Reverse-Proxy agiert als Vermittler zwischen Clients, wie beispielsweise Webbrowsern, und Webservern. Anstelle einer direkten Kommunikation mit dem Ursprungsserver senden Clients ihre Anfragen an den Reverse-Proxy, der diese an den zuständigen Server weiterleitet und die Antworten an die Clients zurücksendet [38]. Nginx wird als Reverse-Proxy verwendet [23], da dieser Server umfassend dokumentiert und weit verbreitet ist. Nginx ist als Docker-Container verfügbar, was die Integration erleichtert [23].

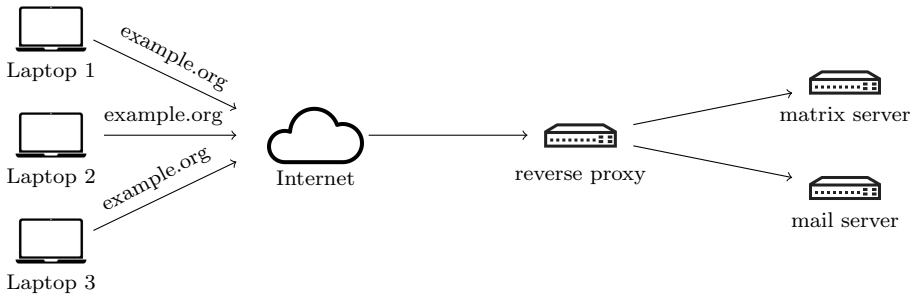

Abbildung 5: Aufbau eiens reverse proxies

5.4.3 Postgres

Die Einrichtung des Matrix Docker-Containers mit PostgreSQL wird empfohlen (siehe [13]). Standardmäßig dient SQLite als Datenbank, da es die Konfiguration vereinfacht. Für eine verbesserte Leistungsoptimierung empfiehlt es sich jedoch, PostgreSQL zu verwenden, das besser optimiert ist. PostgreSQL ist ein leistungsstarkes Datenbanksystem, das SQL verwendet und erweitert. Es bietet zahlreiche Funktionen zur sicheren Speicherung und Skalierung komplexer Datenlasten und wurde im Rahmen des POSTGRES-Projekts an der University of California in Berkeley entwickelt [25].

5.4.4 Certbot

Um die Unterstützung der SSL-Verschlüsselung auf dem Matrixserver sicherzustellen, ist ein SSL-Zertifikat notwendig. SSL (Secure Sockets Layer) sichert Internetverbindungen durch Verschlüsselung der Daten zwischen Browsern und Servern und gewährleistet die Vertraulichkeit sowie Integrität sensibler Informationen. Zudem ermöglicht es die Authentifizierung des Servers gegenüber dem Browser, um sicherzustellen, dass die Verbindung authentisch und sicher ist [39]. Certbot ist ein benutzerfreundlicher Client, der ein Zertifikat von Let's Encrypt abruft. Das SSL-Zertifikat von Let's Encrypt ist kostenlos.

5.4.5 Matrix Bridge

Eine Matrix-Bridge wird verwendet, um den E-Mail-Server mit Matrix zu verbinden [15]. Bridges in Matrix erleichtern die Verbindung zu anderen Plattformen und unterstützen die Interoperabilität, was die Vernetzung von Matrix über verschiedene digitale Plattformen hinweg ermöglicht [4]. Die Bridge gestattet es, E-Mails über Matrix zu senden und E-Mail-Inhalte in Matrix verfügbar zu machen. Alarme werden mit der Bridge, ausschließlich über E-Mail ausgelöst, die automatisch auch nach Matrix versendet wird.

5.4.6 Matrix Client

Eine eigenständige Matrix-Clientanwendung wurde als zusätzliche Sicherheitsmaßnahme entwickelt, um im Falle eines Ausfalls des E-Mail-Servers Nachrichten zu versenden. Sie ist direkt mit der Matrix-Bridge integriert und wurde unter Einsatz der Nio-Bibliothek für die Matrix-API entwickelt [24]. Diese Anwendung sendet automatisch eine Benachrichtigung, sobald über MQTT gemeldet wird, dass eine Person gestürzt ist.

5.5 Mailcow

5.5.1 Einführung in Mailcow

Mailcow ist eine Open-Source-Mailserver-Suite, die eine umfassende E-Mail-Lösung bietet. Verschiedene Dienste wie Postfix für den Mail-Transport, Dovecot für die Speicherung und den Zugriff auf E-Mails sowie SOGo als Webmail-Oberfläche werden kombiniert. Eine benutzerfreundliche Weboberfläche ermöglicht die Verwaltung von Domains, Benutzern und E-Mail-Quotas. Durch die Integration von modernen Sicherheitsmechanismen wie DMARC, DKIM und SPF wird ein hohes Maß an Sicherheit für die E-Mail-Kommunikation gewährleistet.

5.5.2 IMAP / SMTP

IMAP (Internet Message Access Protocol) und SMTP (Simple Mail Transfer Protocol) sind grundlegende Protokolle für den E-Mail-Verkehr. IMAP wird verwendet, um E-Mails vom Server zu lesen und zu verwalten, während SMTP zum Versenden von E-Mails dient. In Mailcow übernimmt Dovecot die Rolle des IMAP-Servers und Postfix die des SMTP-Servers.

5.5.3 DNS MX-Record

Der DNS MX-Record (Mail Exchanger Record) ist ein entscheidender Teil der E-Mail-Infrastruktur. Dieser Record gibt an, welche Mailserver für den Empfang von E-Mails für eine bestimmte Domain verantwortlich sind. Bei der Konfiguration von Mailcow muss der MX-Record der Domain auf den Mailcow-Server zeigen, um sicherzustellen, dass eingehende E-Mails korrekt zugestellt werden. Eine richtige Konfiguration des MX-Records ist notwendig für die Zustellung von E-Mails. Da die richtige Konfiguration des DNS notwendig ist, hat das Mailcow - Dockerized Projekt in ihrer Dokumentation Beschreibungen, wie man diesen richtig aufsetzt [18].

5.5.4 Sicherheit: DKIM

DKIM (DomainKeys Identified Mail) ist ein Authentifizierungsprotokoll, das E-Mails mit einer digitalen Signatur versieht. Diese Signatur wird von einem öffentlichen Schlüssel überprüft, der im DNS der sendenden Domain veröffentlicht ist [9]. DKIM hilft sicherzustellen, dass E-Mails nicht verändert wurden und wirklich von der angegebenen Domain stammen. In Mailcow kann ein DKIM-Schlüssel-Paar leicht über die Weboberfläche erstellt werden, um die Authentizität der ausgehenden E-Mails zu gewährleisten. Der öffentliche Schlüssel muss dann noch in einem DNS Record eingetragen werden.

5.5.5 Sicherheit: SPF

SPF (Sender Policy Framework) Mail Records sind ein Mechanismus zur Verhinderung von E-Mail-Spoofing. Sie ermöglichen es einem Domaininhaber, festzulegen, welche Mail-Server berechtigt sind, E-Mails im Namen ihrer Domain zu senden [36]. Dies wird durch die Veröffentlichung eines SPF-Eintrags im DNS der Domain erreicht, der eine Liste autorisierter IP-Adressen enthält. E-Mail-Empfangsserver überprüfen den SPF-Eintrag der absendenden Domain und verwerfen oder markieren E-Mails als Spam, wenn sie von einem nicht autorisierten Server gesendet wurden. Dadurch wird die Wahrscheinlichkeit reduziert, dass schädliche oder gefälschte E-Mails erfolgreich zugestellt werden.

5.5.6 Sicherheit: DMARC

DMARC (Domain-based Message Authentication, Reporting, and Conformance) Mail Records bauen auf den Mechanismen von SPF und DKIM auf, um eine umfassendere E-Mail-Authentifizierung zu bieten. DMARC ermöglicht es, Richtlinien zu definieren, wie E-Mail-Empfangsserver mit Nachrichten umgehen sollen, die SPF- oder DKIM-Prüfungen nicht bestehen [37]. Ein DMARC-Eintrag im DNS der Domain legt fest, ob solche E-Mails abgewiesen, in Quarantäne gestellt oder zugestellt werden sollen, und bietet zudem eine Reporting-Funktion. Dadurch können Domaininhaber Berichte über gefälschte E-Mails erhalten und ihre E-Mail-Sicherheitsrichtlinien kontinuierlich verbessern. DMARC hilft somit, die E-Mail-Sicherheit zu stärken und Phishing-Angriffe effektiver zu bekämpfen.

5.5.7 Installation und Konfiguration

Zur Konfiguration von Mailcow kann die Dokumentation von Mailcow - Dockerized genutzt werden [18]. Dieses Projekt bietet fertige Docker - Container an, die nur noch über eine Konfigurationsdatei angepasst werden müssen. Die Anleitung des Projekts sollte befolgt werden. Es ist wichtig zu wissen, dass die Einstellung des DNS-Servers nicht automatisch erfolgt, sondern manuell erledigt werden muss. Glücklicherweise hilft auch hier die Dokumentation von Mailcow - Dockerized, denn es gibt ein Kapitel, das das DNS-Setup inklusive SPF, DKIM und DMARC erklärt. Es wird auch auf Websites verwiesen, die diese Sicherheitsmechanismen testen können.

5.5.8 E-Mail-Client

Zum Versenden von Mails wird ein Mail-Client benötigt, in dem eine E-Mail geschrieben und an den Mail - Server des Accounts gesendet wird. Der Mail - Server kann die E-Mail mit seinem privaten DKIM-Schlüssel signieren und die Mail an den E-Mail-Server des Empfängers weiterleiten. Da das automatische Versenden von Mails ein häufiges Problem darstellt, gibt es hierfür fertige Lösungen. Es wird das Python-Modul "emails" verwendet.

6 Evaluation

6.1 Hardware Evaluation

Die Hardware des Patient Monitoring Systems wird hinsichtlich ihrer Benutzerfreundlichkeit, Zuverlässigkeit und Effizienz bewertet. Der Einsatz von kostengünstigen und leicht verfügbaren Komponenten wie dem Raspberry Pi gewährleistet eine einfache Installation und Wartung. Die LEDs und der Buzzer bieten klare und unmittelbare visuelle und akustische Rückmeldungen, die für eine effektive Überwachung notwendig sind. Die durchgeführten Tests bestätigen die hohe Zuverlässigkeit der Hardwarekomponenten, die unter verschiedenen Bedingungen stabil und zuverlässig funktionieren. Dadurch wird sichergestellt, dass das System in der Praxis gut einsetzbar ist und die Anforderungen der kontinuierlichen Patientenüberwachung erfüllt.

6.2 Yolov5

Das Modell zeigt laut dem Entwickler eine Genauigkeit von 85 bis 90% bei Tageslicht. Es hat jedoch Schwierigkeiten, viele Personen auf einem Bild zu erkennen, was zu falschen Erkennungen führen kann. Für diesen Anwendungsfall, d.h. die Überwachung von Patienten, ist dies jedoch kein Problem, da in den Bildern normalerweise nur einzelne Patienten vorhanden sind [16].

6.2.1 Testumgebung

Das Modell wird in verschiedenen Sturzsituationen getestet. Wie im Testvideo zu sehen ist [41], funktioniert das Modell weitgehend wie vom Autor beschrieben. Allerdings treten gelegentlich Fehlermeldungen auf, die im Video ebenfalls dokumentiert sind. Diese Fehler sind vermutlich auf die Verwendung des kleineren YOLOv5s-Modells zurückzuführen.

6.2.2 Testergebnisse

Das Modell überzeugt insgesamt mit seinen Klassifikationen. Es ermöglicht, mit 1-2 FPS auf einem Raspberry Pi 4 Inferenz durchzuführen. Dadurch werden die Kosten für einen teuren Server gespart und das Projekt kann dynamisch mit den Kameras skaliert werden.

6.3 MQTT - Evaluation

6.3.1 Effizienz und Herausforderungen

MQTT bietet eine effiziente Lösung für die Kommunikation zwischen Geräten mit begrenzten Ressourcen. Im Betrieb zeigen sich jedoch einige Herausforderungen. Insbesondere das Versenden von Bildern einer Kamera über

MQTT führt zu Verzögerungen. Diese Verzögerungen beeinträchtigen die Echtzeit-Kommunikation erheblich.

6.3.2 Zuverlässigkeitsprobleme

Zusätzlich bleibt der MQTT-Broker nicht immer zuverlässig am Laufen. Es treten gelegentlich Ausfälle auf, bei denen der Broker nicht erreichbar ist, obwohl Server, DNS und Firewall ordnungsgemäß funktionieren. Diese Unzuverlässigkeit stellt ein erhebliches Problem dar und mindert die Gesamtperformance des Systems. Die Ursachen dieser Ausfälle müssen weiter untersucht werden, um eine stabile und zuverlässige Kommunikation sicherzustellen.

6.3.3 Vorteile und Fazit

Abgesehen von den genannten Problemen funktioniert die Kommunikation über MQTT gut und ist einfach zu implementieren. Das Publisher/Subscriber-Modell ermöglicht eine flexible und effiziente Nachrichtenübermittlung. Die leichte Konfiguration und der weitverbreitete Einsatz von MQTT-Clients in verschiedenen Programmiersprachen erleichtern die Integration in bestehende Systeme erheblich. Trotz der technischen Schwierigkeiten bietet MQTT somit eine solide Grundlage für die Kommunikation in dem System.

6.4 Mailcow - Evaluation

6.4.1 Einfachheit der Einrichtung

Die Einrichtung von Mailcow gestaltet sich besonders einfach, da Mailcow-Dockerized vorgefertigte Container bereitstellt, die nur noch konfiguriert werden müssen. Das spart erheblich Zeit und reduziert mögliche Fehlerquellen, die bei einer manuellen Installation auftreten könnten.

6.4.2 Hervorragende Dokumentation

Die Dokumentation von Mailcow erweist sich als sehr gut und enthält detaillierte Anweisungen, einschließlich des DNS-Setups. Diese klaren und verständlichen Schritt-für-Schritt-Anleitungen erleichtern die Konfiguration erheblich und stellen sicher, dass alle notwendigen Schritte korrekt ausgeführt werden.

6.4.3 Vorteile der integrierten Web-Oberfläche

Ein wesentlicher Vorteil von Mailcow gegenüber einer reinen Postfix/Dovecot-Installation ist die integrierte Web-Oberfläche. Diese Web-UI erleichtert die Verwaltung und Konfiguration des E-Mail-Servers erheblich, indem sie eine benutzerfreundliche und intuitive Oberfläche bietet. Im Gegensatz zu einer

alleinigen Postfix/Dovecot-Installation, bei der viele Einstellungen manuell über Konfigurationsdateien vorgenommen werden müssen, ermöglicht Mailcow eine effiziente Verwaltung über das Web-Interface.

6.4.4 Log- und Monitoring-Tools

Zusätzlich bietet die Web-Oberfläche von Mailcow integrierte Log- und Monitoring-Tools. Diese Tools ermöglichen eine umfassende Überwachung und Analyse der Serveraktivitäten, was die Fehlersuche und -behebung vereinfacht und die allgemeine Sicherheit und Stabilität des Systems erhöht. Durch die zentrale Verwaltung und das Monitoring über die Web-UI werden administrative Aufgaben deutlich effizienter und übersichtlicher.

6.5 Benutzerfreundlichkeit

Das System zeigt eine hohe Benutzerfreundlichkeit, da es einfach zu bedienen ist und durch eine intuitive Benutzeroberfläche unterstützt wird. Die Installation und Konfiguration sind klar dokumentiert und auch für Nutzer ohne tiefgehende technische Kenntnisse leicht nachvollziehbar. Pflegekräfte können das System schnell erlernen und effizient nutzen, was die Überwachung der Patienten erheblich erleichtert. Besonders die Echtzeit-Benachrichtigungen und die visuelle Alarmierung durch LEDs und akustische Signale tragen zur schnellen Reaktion in Notfallsituationen bei.

6.6 Preis

Das Patient Monitoring System ist kosteneffizient konzipiert. Durch den Einsatz von kostengünstigen Komponenten wie dem Raspberry Pi und frei verfügbarer Open-Source-Software bleiben die Gesamtkosten überschaubar. Im Vergleich zu kommerziellen Lösungen bietet das System eine vergleichbare Funktionalität zu einem Bruchteil der Kosten. Dies macht es besonders attraktiv für kleinere medizinische Einrichtungen oder Pflegeheime mit begrenztem Budget.

6.7 Einfachheit des Systems

Die Einfachheit des Systems wird durch den modularen Aufbau und die Nutzung bewährter Technologien wie Docker, MQTT und UFW unterstrichen. Die Systemarchitektur ist so gestaltet, dass Erweiterungen und Anpassungen problemlos vorgenommen werden können. Auch die Wartung des Systems erfordert nur minimalen Aufwand, da die meisten Komponenten automatisch aktualisiert werden können. Die klare Struktur und die ausführliche Dokumentation erleichtern zudem die Fehlersuche und -behebung, was die Zuverlässigkeit und Stabilität des Systems weiter erhöht.

7 Zusammenfassung und Ausblick

Das Projekt zur Patientenüberwachung demonstriert, wie moderne Technologien die Pflegequalität in medizinischen Einrichtungen verbessern können. Durch den Einsatz von Docker, Raspberry Pi und MQTT wurde ein zuverlässiges Überwachungssystem entwickelt, das die Sicherheit und Integrität der Patientendaten gewährleistet.

Für die Zukunft gibt es mehrere Erweiterungsmöglichkeiten: Die Erkennung, ob eine Person noch im Bett liegt, könnte helfen auch stürzte auf annonymen Orten wie einer Toilette zu erkennen. Ein Identifikationssystem könnte Pflegekräfte von Patienten unterscheiden und so die Effizienz erhöhen. Zudem könnte die "Bystander Anonymization"den Datenschutz verbessern, indem unbeteiligte Personen in Videoaufnahmen unkenntlich gemacht werden. Schließlich könnte "Digital Fencing"durch virtuelle Grenzen die Bewegungsüberwachung optimieren.

Schlussendlich könnte man zudem versuchen, die Genauigkeit des Modells mithilfe einer neueren und größeren Version von YOLO, wie beispielsweise YOLOv8, zu erhöhen. Hierfür müsste man jedoch statt des Raspberry Pi einen externen Server anbinden. Diese Erweiterungen würden das System noch vielseitiger und leistungsfähiger machen und die Sicherheit und das Wohlbefinden der Patienten weiter steigern.

8 Glossaries

8.1 Docker

Docker ist eine Open-Source-Softwareplattform, die die Virtualisierung auf Betriebssystemebene nutzt, um Software in Paketen, sogenannten Containern, zu entwickeln, auszuliefern und auszuführen. Container ermöglichen es, eine Anwendung mit allen Teilen, die sie zum Ausführen benötigt, wie Bibliotheken und andere Abhängigkeiten, zu kapseln und auf jedem Linux-System auszuführen, als wären sie auf einem Computer installiert [7] . Dies gewährleistet die Konsistenz unabhängig von der Umgebung und verbessert die Sicherheit, da Anwendungen isoliert voneinander ausgeführt werden [10].

8.2 Container

Ein Container in der Informationstechnologie, ist eine leichtgewichtige, ausführbare Einheit, die Softwarepakete, ihre Abhängigkeiten und andere notwendige Komponenten enthält. Container bieten eine isolierte Umgebung für die Ausführung von Anwendungen, sodass diese konsistent und zuverlässig auf unterschiedlichen Systemen funktionieren können, ohne von anderen Anwendungen beeinflusst zu werden [3].

8.3 Mailcow

Mailcow ist eine Open-Source-Mailserver-Lösung, die auf Docker basiert und die Verwaltung von E-Mail-Diensten vereinfacht. Sie bietet ein integriertes Webinterface zur Konfiguration und Verwaltung von E-Mail-Konten, Domains, Sicherheitseinstellungen und weiteren Diensten wie Kalender und Kontakte. Mailcow integriert auch Anti-Spam- und Anti-Virus-Tools, um eine sichere und effiziente E-Mail-Kommunikation zu gewährleisten [17].

8.4 MQTT

MQTT (Message Queuing Telemetry Transport) ist ein Netzwerkprotokoll, das speziell für die Bedürfnisse von Internet-of-Things (IoT)-Anwendungen entwickelt wurde. Es ermöglicht es Geräten, Daten in einem schwach vernetzten Netzwerk effizient zu übertragen. Das Protokoll basiert auf einem Publisher/Subscriber-Modell, das eine hohe Nachrichtenübermittlungseffizienz bei gleichzeitig geringem Ressourcenbedarf bietet [22]. MQTT wird häufig in der Heimautomatisierung, in der Medizintechnik und anderen Bereichen verwendet, in denen eine zuverlässige und effiziente Kommunikation zwischen einer Vielzahl von Geräten erforderlich ist.

8.5 Raspberry Pi

Der Raspberry Pi ist ein kleiner, preiswerter Einplatinencomputer, der ursprünglich für Bildungszwecke entwickelt wurde, aber aufgrund seiner Vielseitigkeit und Leistungsfähigkeit in einer Vielzahl von industriellen und Hobby-Projekten eingesetzt wird. Der Pi kann mit verschiedenen Betriebssystemen wie Raspbian, einem Debian-basierten Linux, betrieben werden und unterstützt eine breite Palette von Anwendungen und Programmiersprachen [29].

8.6 Fail2Ban

Fail2Ban ist eine Intrusion Prevention Software, die durch Überwachen von Serverprotokollen auf Muster von Missbrauchsversuchen wie zu viele fehlgeschlagene Anmeldeversuche reagiert. Wenn ein solches Muster erkannt wird, konfiguriert Fail2Ban die Firewall des Hosts temporär so, dass sie IP-Adressen blockiert, von denen aus die verdächtigen Aktivitäten ausgehen, um so das System zu schützen [33].

8.7 Ubuntu 22.04 LTS

Ubuntu 22.04 LTS ist eine Version des Ubuntu-Betriebssystems, die auf Stabilität und langfristige Unterstützung ausgerichtet ist, was sie besonders für Unternehmensumgebungen geeignet macht. Es bietet regelmäßige Sicherheitsupdates für fünf Jahre, was eine zuverlässige Grundlage für Projekte gewährleistet, die eine langfristige Wartbarkeit erfordern [34].

8.8 Firewall

Eine Firewall ist ein Sicherheitssystem, das den eingehenden und ausgehenden Netzwerkverkehr überwacht und steuert, basierend auf vordefinierten Sicherheitsregeln. Sie dient dazu, unerwünschten oder schädlichen Datenverkehr zu blockieren und somit Netzwerke und angeschlossene Geräte vor Angriffen und externen Bedrohungen zu schützen [35].

8.9 SSH (Secure Shell)

SSH ist ein Netzwerkprotokoll, das für eine sichere Kommunikation über unsichere Netzwerke konzipiert wurde. Es ermöglicht verschlüsselten Datentransfer und Fernsteuerung von Netzwerkgeräten, was die sichere Verwaltung von Servern und anderen Netzwerkkomponenten erlaubt [2].

8.10 DDclient

DDclient ist ein Perl-Script, das zur automatischen Aktualisierung von DNS-Einträgen genutzt wird, um dynamisch zugewiesene IP-Adressen mit einem

festen Hostnamen zu verknüpfen. Dies ist besonders nützlich in Umgebungen mit dynamischer IP-Zuweisung, wie sie bei vielen Internetdienstanbietern üblich ist [8] [29].

8.11 DNS-Einträge

DNS-Einträge (Domain Name System Einträge) sind Informationen in einer DNS-Datenbank, die dazu dienen, Domainnamen in IP-Adressen umzuwandeln. Diese Einträge ermöglichen es Computern, Netzwerkressourcen zu identifizieren und zu lokalisieren, wodurch Benutzer einfacher auf Websites und andere im Internet gehostete Dienste zugreifen können.

8.12 Log-Dateien

Log-Dateien sind Dateien, in denen Ereignisse aufgezeichnet werden, die in einem Betriebssystem oder einer Softwareanwendung stattfinden. Sie sind entscheidend für das Debugging und die Überwachung von Systemen, da sie detaillierte Informationen über den Betriebsstatus und aufgetretene Fehler enthalten.

8.13 Brute-Force-Angriffe

Brute-Force-Angriffe sind eine Methode der Cyberkriminalität, bei der automatisierte Software verwendet wird, um viele Kombinationen von Benutzernamen und Passwörtern zu erraten, um unerlaubten Zugriff auf Benutzerkonten zu erhalten. Diese Angriffe können durch starke Passwörter und zusätzliche Sicherheitsmaßnahmen wie Captchas oder Zwei-Faktor-Authentifizierung erschwert werden.

8.14 SMTP

Ein Simple Mail Transfer Protocol (SMTP)-Server ist ein Netzwerkserver, der E-Mails sendet und empfängt. Er spielt eine zentrale Rolle in der elektronischen Kommunikation und wird häufig in Unternehmensumgebungen eingesetzt, um die interne und externe Kommunikation zu verwalten [31].

8.15 Matrix

Matrix ist ein offenes Protokoll für Echtzeitkommunikation, das speziell für die Anforderungen des modernen Internets entwickelt wurde. Es unterstützt verschlüsselte Chat-, VoIP- und Videokonferenzdienste und ermöglicht die Interoperabilität über verschiedene Kommunikationsplattformen hinweg [20].

8.16 YOLO

YOLO (You Only Look Once) ist ein schnelles, effizientes Objekterkennungssystem, das in einem einzigen Durchlauf durch das neuronale Netzwerk sowohl die Klassifizierung als auch die Lokalisierung von Objekten in Bildern durchführt. Dieses Modell ist dafür bekannt, dass es Echtzeitverarbeitungsgeschwindigkeiten erreicht, was es besonders nützlich für Anwendungen macht, die eine sofortige Objekterkennung erfordern, wie beispielsweise in der Videoüberwachung und im autonomen Fahren [40].

8.17 Bett-Detection

Bett-Detection bezieht sich auf die automatische Erkennung der Präsenz oder Abwesenheit einer Person im Bett mithilfe von Kameraüberwachung. Dieses System ist besonders nützlich in medizinischen Überwachungsumgebungen, um sicherzustellen, dass Patienten sicher sind und sich an den zugewiesenen Orten befinden.

8.18 Fall-Detection

Fall-Detection umfasst die Technologie zur Erkennung von Stürzen, insbesondere in Umgebungen wie Krankenhäusern oder Pflegeheimen, indem ungewöhnliche Bewegungen oder Lagen der Patienten analysiert werden. Diese Systeme setzen fortschrittliche Technik ein, um Stürze sofort zu erkennen und entsprechende Alarme auszulösen, was eine schnelle Reaktion des Pflegepersonals ermöglicht.

8.19 Zustand Normal

Der Zustand Normal im beduetet, dass an einem Raspberry Pi eine grüne LED leuchtet, die über GPIO angeschlossen ist.

8.20 Zustand Warnung

Eine Warnung im Kontext dieses Projekts bedeutet, dass an einem Raspberry Pi eine gelbe LED blinkt, die über GPIO angeschlossen ist.

8.21 Zustand Alarm

Ein Alarm im Kontext des Projekt bedeutet, dass an einem Raspberry Pi eine rote LED und einen kleinen Piepser für akustische Signale eine Meldung aus geben.

8.22 SSL

SSL, oder Secure Sockets Layer, ist ein Sicherheitsprotokoll, das dazu dient, die Datenübertragung zwischen einem Webbrowser und einem Server zu verschlüsseln. Es schützt sensible Informationen vor dem Abfangen durch unbefugte Dritte während der Übertragung im Internet.

8.23 SSL-Zertifikat

Ein SSL-Zertifikat ist eine digitale Datei, die die Identität einer Website bestätigt und eine verschlüsselte Verbindung zwischen einem Webserver und dem Browser des Benutzers ermöglicht. Es verwendet Public Key-Infrastruktur, um Daten zu verschlüsseln und zu verifizieren, dass die Kommunikation nur zwischen den vorgesehenen Empfängern stattfindet [6].

8.24 Reverse-Proxy

Ein Reverse-Proxy ist ein Server, der als Vermittler zwischen den Endbenutzern und den Diensten, auf die sie zugreifen, fungiert, um Anfragen an die entsprechenden Server weiterzuleiten und Antworten zurückzusenden. Er wird häufig eingesetzt, um die Lastverteilung zu verbessern, die Sicherheit zu erhöhen und die Leistung von Webanwendungen zu optimieren[5].

Literatur

[1] *About - OpenCV*. URL: https://opencv.org/about/ (besucht am 26. 04. 2024).

[2] Daniel J. Barrett, Richard E. Silverman und Robert G. Byrnes. *SSH, the Secure Shell: The Definitive Guide*. Second. Sebastopol, CA: O'Reilly Media, Inc., 2005. ISBN: 0-596-00895-3.

[3] David Bernstein. "Containers and Cloud: From LXC to Docker to Kubernetes". In: *IEEE Cloud Computing* 1.3 (2014). DOI: 10.1109/MCC.2014.51.

[4] *Bridges*. URL: https://matrix.org/ecosystem/bridges/ (besucht am 26. 04. 2024).

[5] Cloudflare. *Was ist ein Reverse-Proxy?* 2024. URL: %5Curl%7B%20https://www.cloudflare.com/de-de/learning/cdn/glossary/reverse-proxy/%7D (besucht am 11. 05. 2024).

[6] Cloudflare. *What is an SSL Certificate?* 2024. URL: %5Curl%7B%20https://www.cloudflare.com/de-de/learning/ssl/what-is-an-ssl-certificate/%20%7D (besucht am 11. 05. 2024).

[7] Aeris Consulting. *The Compact Container Platform*. 2023. URL: https://www.aeris-consulting.com/wp-content/uploads/2022/04/Docker.pdf (besucht am 11. 05. 2024).

[8] *ddclient - A Perl client used to update dynamic DNS entries*. 2024. URL: https://ddclient.net/ (besucht am 15. 04. 2024).

[9] *DKIM - Definition Was ist das?* URL: https://www.cleverreach.com/de-de/push-magazin/email-marketing-strategie-tipps/definitionen-e-mail-marketing/dkim-was-ist-das-eine-definition/ (besucht am 23. 06. 2024).

[10] *Docker - Offizielle Webseite*. 2024. URL: https://www.docker.com/ (besucht am 15. 04. 2024).

[11] *Docker Compose overview*. URL: https://docs.docker.com/compose/ (besucht am 22. 06. 2024).

[12] *Element — Secure collaboration and messaging*. URL: https://element.io/ (besucht am 26. 04. 2024).

[13] *Installation - Synapse*. URL: https://matrix-org.github.io/synapse/latest/setup/installation.html#using-postgresql (besucht am 26. 04. 2024).

[14] *Introduction*. URL: https://matrix.org/docs/older/introduction/ (besucht am 26. 04. 2024).

[15] Jojii. *JojiiOfficial/Matrix-EmailBridge*. original-date: 2019-08-22T19:04:09Z. 23. Apr. 2024. URL: https://github.com/JojiiOfficial/Matrix-EmailBridge (besucht am 26. 04. 2024).

[16] UTTEJ KUMAR. *uttej2001/Image-based-Human-Fall-Detection*. original-date: 2022-01-07T05:39:25Z. 22. Apr. 2024. URL: https://github.com/uttej2001/Image-based-Human-Fall-Detection (besucht am 25. 04. 2024).

[17] *Mailcow - Website*. URL: https://docs.mailcow.email/ (besucht am 13. 04. 2024).

[18] *Mailcow DNS Setup*. URL: https://docs.mailcow.email/getstarted/prerequisite-dns/ (besucht am 11. 06. 2024).

[19] *Make Sense*. URL: https://www.makesense.ai/ (besucht am 25. 04. 2024).

[20] *Matrix*. URL: https://matrix.org/ (besucht am 13. 04. 2024).

[21] *matrixdotorg/synapse - Docker Image — Docker Hub*. URL: https://hub.docker.com/r/matrixdotorg/synapse (besucht am 26. 04. 2024).

[22] *MQTT - The Standard for IoT Messaging - Website*. URL: https://mqtt.org/ (besucht am 13. 04. 2024).

[23] *NGINX Reverse Proxy — NGINX Documentation*. URL: https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/ (besucht am 26. 04. 2024).

[24] *nio — nio 0.25.0rc3 documentation*. URL: https://matrix-nio.readthedocs.io/en/latest/index.html (besucht am 11. 06. 2024).

[25] *PostgreSQL: About*. URL: https://www.postgresql.org/about/ (besucht am 26. 04. 2024).

[26] *Python*. URL: https://www.python.org/ (besucht am 13. 04. 2024).

[27] *PyTorch*. Alexander Thamm GmbH. URL: https://www.alexanderthamm.com/de/data-science-glossar/pytorch/ (besucht am 26. 04. 2024).

[28] *QT Framework*. URL: https://www.qt.io/product/framework/ (besucht am 13. 04. 2024).

[29] *Raspberry Pi*. URL: https://www.raspberrypi.com/ (besucht am 13. 04. 2024).

[30] *Raspberry Pi Documentation*. URL: https://www.raspberrypi.com/documentation/accessories/camera.html (besucht am 13. 04. 2024).

[31] Vladimir V. Riabov. *SMTP (Simple Mail Transfer Protocol)*. 2005. (Besucht am 11. 05. 2024).

[32] *Set environment variables within your container's environment*. URL: https://docs.docker.com/compose/environment-variables/set-environment-variables/ (besucht am 22. 06. 2024).

[33] C. C. Teng u. a. “Firmware over the Air for Home Cybersecurity in the Internet of Things”. In: *19th Asia-Pacific Network Operations and Management Symposium: Managing a World of Things, APNOMS 2017*. 2017, S. 123–128. DOI: 10.1109/APNOMS.2017.8094190.

[34] *Ubuntu - Offizielle Webseite*. 2024. URL: https://ubuntu.com/ (besucht am 15. 04. 2024).

[35] John Wack, Ken Cutler und Jamie Pole. *Guidelines on Firewalls and Firewall Policy*. 2002-01-01 2002.

[36] *Was bedeutet SPF?* URL: https://www.proofpoint.com/de/threat-reference/spf (besucht am 23. 06. 2024).

[37] *Was ist ein DNS DMARC - Eintrag? — Cloudflare*. URL: https://www.cloudflare.com/de-de/learning/dns/dns-records/dns-dmarc-record/ (besucht am 23. 06. 2024).

[38] *Was ist ein Reverse Proxy, was ist ein Proxy Server?* URL: https://www.cloudflare.com/de-de/learning/cdn/glossary/reverse-proxy/ (besucht am 26. 04. 2024).

[39] *Was sind SSL, TLS und HTTPS? — DigiCert*. URL: https://www.digicert.com/de/what-is-ssl-tls-and-https (besucht am 26. 04. 2024).

[40] *Yolo*. URL: https://docs.ultralytics.com/ (besucht am 13. 04. 2024).

[41] *Yolo Video*. URL: https://fbe-gitlab.hs-weingarten.de/stud-iki/prj-sysadmin/ss24_tutar_kleiser_metzler_AAL_Patientenmonitoring/-/blob/dev/source/yolov5/detection_video/video.mp4?ref_type=heads (besucht am 11. 06. 2024).