

IT Security Audit Swiss Medical Clinic

Teleinformatik Services AG
Abteilung Informatik
Schulstrasse 37
CH-8050 Zürich

Vertraulich

Auftraggeber: Swiss Medical Clinic, Grütstrasse 55, CH-8802 Kilchberg ZH
Autor: David Meister, Security Engineer, Teleinformatik Services AG
Mitwirkende: Giorgio Vincenti, Head of IT, Teleinformatik Services AG
Daniel Luetold, Systems Engineer, Teleinformatik Services AG
Dr. med. Anwar Giryes, Swiss Medical Clinic
Tochter von Giryes, Swiss Medical Clinic
Datum: 11. Januar 2019

Abstract

Das Internet of Things erfreut sich immer grösserer Beliebtheit. Unternehmensanwendungen mit einer grossen Anzahl Sensoren werden in Zukunft stark zunehmen. Bereits bei herkömmlichen Computersystemen stellt das Management eine grosse Herausforderung dar. IoT Devices dürften potenziell in einer noch deutlich grösseren Anzahl verbreitet sein als herkömmliche Geräte. Unternehmen benötigen Lösungen, um eine regelrechte Flut von neuartigen Devices effizient und sicher administrieren zu können. In dieser Arbeit werden wichtige Management-Aufgaben wie beispielsweise Bootstrapping, Konfiguration, Updates und Monitoring für IoT Systeme untersucht.

Nach einer umfassenden Einarbeitung in IoT Technologien und Management-Aufgaben mussten geeignete Kommunikationsprotokolle untersucht werden. Es stellte sich heraus, dass herkömmliche Protokolle wie HTTP im IoT Umfeld schlecht geeignet sind, da sie häufig nur über eingeschränkte Ressourcen wie Bandbreite und Rechenleistung verfügen. Das von der Open Mobile Alliance (OMA) entwickelte Lightweight Machine-to-Machine (LwM2M) Protokoll verwendet unterhalb das Constrained Application Protocol (CoAP). CoAP arbeitet im Gegensatz zu HTTP über UDP und besitzt im Allgemeinen einen geringeren Protokoll Overhead, weshalb es sich im IoT Umfeld besser eignet. LwM2M wurde entwickelt, um standardisierte Zugriffe auf Device Informationen zu ermöglichen. Als nächstes wurde eine geeignete Architektur für eine Management Webapplikation evaluiert.

Aus dieser Bachelorarbeit ist die Management Webapplikation "Smartmanager" entstanden. Das Back-End wurde mit dem Java Spring Framework erstellt. Der LwM2M Server wurde mit der Eclipse Leshan Library implementiert, als Datenbank wurde MongoDB verwendet. Mit dem "Smartmanager" können LwM2M Clients administriert werden und die integrierte Gruppenverwaltung ermöglicht Devices in selbst angelegte Hierarchien zu strukturieren. Ausserdem können Konfigurationen bestehend aus unterschiedlichen Schreiboperationen erstellt -, welche wiederum auf einzelne Geräte oder ganze Gruppen geschrieben werden können. Dank der Verwendung des LwM2M Protokolls können eine grosse Anzahl an Management-Aufgaben ausgeführt werden. Es bleibt zu hoffen, dass sich LwM2M zukünftig als Standard für das IoT Device Management durchsetzen wird um die einheitliche Verwaltung verschiedenster Geräte zu ermöglichen.

Management Summary

Ausgangslage

Internet of Things wird in Unternehmen immer wichtiger. Mit Hilfe von Sensoren werden bei IoT-Anwendungen reale Objekte miteinander verbunden. Bewährte Konzepte der Informatik werden angewandt, um intelligente Systeme von vernetzten Objekten zu erschaffen. In der Industrie werden bereits heute eine grosse Anzahl unterschiedlicher Sensoren eingesetzt. Vieles scheint bei IoT auf den ersten Blick bekannt, es gibt jedoch einige technische und organisatorische Unterschiede zu beachten. IoT-Devices sind verglichen mit herkömmlichen PCs und Notebooks in vielerlei Hinsicht eingeschränkt. Sie verfügen beispielsweise über weniger Ressourcen wie Rechenleistung, Speicher und Netzwerkbandbreite. Des Weiteren werden Sensoren in einer sehr viel grösseren Anzahl als herkömmliche Geräte vorhanden sein. Zukünftig werden ganze Geschäftsprozesse von einer funktionierenden IoT-Infrastruktur abhängig sein, weshalb Sicherheit immer wichtiger wird.

In einem Firmenumfeld sollten Sensoren und IoT-Devices effizient und einheitlich ausgerollt werden. Weitere Verwaltungsaufgaben wie Patching, Monitoring oder Troubleshooting müssen auch im IoT-Umfeld berücksichtigt werden. Ziel dieses Projekts ist eine umfassende Analyse über wichtige Aspekte im Bezug auf Device Management im IoT-Umfeld. Bestehende Ansätze und Lösungen sollen analysiert und erarbeitet werden, um einen geeigneten Prototypen erstellen zu können.

Vorgehen/Technologien

Zu Beginn hat sich das Projektteam ein Basisverständnis im IoT-Umfeld verschafft. Wichtige Aspekte des Device Managements wurden erarbeitet und aufgezeigt. Damit viele Geräte effizient verwaltet werden können, muss eine IoT-Management Applikation Funktionen für automatisches und sicheres Provisioning, Konfigurationsmanagement, Updating und Monitoring bieten.

Kommunikation über Netzwerke ist, wie der Name verrät, bei Internet of Things ein zentrales Thema. In dieser Arbeit wird dem Leser eine Übersicht über die IoT-Architektur, sowie über Kommunikationsmodelle vermittelt. Aufgrund der geänderten Anforderungen sind neue Protokolle entstanden. Es hat sich gezeigt, dass für Geräte mit eingeschränkten Ressourcen Alternativen gewählt werden sollten. Die Open Mobile Alliance (OMA) hat mit dem "Lightweight Machine-to-MachineProtokoll (LwM2M) ein massgeschneidertes Protokoll für Device Management im IoT-Umfeld entwickelt. Bereits existierende Libraries für Implementierungen des LwM2M Stacks konnten für dieses Projekt verwendet werden.

Eine breite Security-Analyse hat die Notwendigkeit von Sicherheitsüberlegungen gezeigt. Gegenüber bestehenden Infrastrukturen könnten zukünftige IoT-Systeme weitaus anfälliger werden. Ausfälle und Kompromittierungen solcher Systeme könnten drastischere Auswirkungen haben als bei heutigen Systemen.

Aufgrund der Erkenntnisse hat das Projektteam Anforderungen an die Management Applikation definiert. Eine Architektur wurde erarbeitet und geeignete Technologien ausgewählt. Der Prototyp hat gezeigt, dass die gewählte Lösung mit einem Java-basierten LwM2M Server funktioniert. Durch das Web-Front-End können Benutzer über einen gängigen Webbrowser auf die Applikation zugreifen.

Ergebnisse

Mit dem "Smartmanager" ist ein funktionsfähiger Prototyp einer IoT-Management-Applikation entstanden. Durch den Einsatz des LwM2M Protokolls können theoretisch viele unterschiedliche IoT-Devices verwaltet werden. Der Benutzer hat die Möglichkeit, Konfigurationen für Devices anzulegen und diese zu verteilen. Die implementierte Gruppenverwaltung für Devices ermöglicht den Aufbau einer gewünschten Struktur. Die durch LwM2M unterstützten Operationen "Read", "Write" und "Execute" können sowohl einzeln auf Devices ausgeführt-, als auch auf Gruppenebene ausgeführt werden. Neue Devices können über die Discovery-Funktion dem System hinzugefügt werden. Dabei können initiale Konfigurationen und Gruppenzugehörigkeit bestimmt werden.

Ausblick

IoT-Device-Management wird für viele Unternehmen ein Thema werden. Es ist erfreulich, dass die OMA mit LwM2M ein Protokoll entwickelt hat, um eine einheitliche Managementlösung für IoT-Devices zu ermöglichen. Es bleibt zu hoffen, dass viele Hersteller im IoT-Umfeld dieses Protokoll implementieren werden.

Wie zu erwarten war, konnten in diesem Projekt zeitlich nicht alle wichtigen Aspekte berücksichtigt werden. Für die produktive Nutzung des Tools könnten noch wichtige Sicherheitsfeatures und automatisiertes Bootstrapping implementiert werden.

Inhaltsverzeichnis

Teil I

Analyse

1. Einführung in Internet of Things

1.1 Übersicht

Das Internet der Dinge unterscheidet sich in einigen Aspekten vom klassischen Internet. End-Benutzer haben über sogenannte Terminals wie Laptops oder Smartphones über die globale Internet Infrastruktur kommuniziert [?]. Diese Terminals wurden meist von Benutzern eingeschaltet, benutzt und wieder ausgeschaltet. Damit Geräte mit dem Internet auf sinnvolle Art und Weise kommunizieren konnten, war eine manuelle Tätigkeit von Benutzern notwendig [?]. Beispiele dafür sind das Abrufen von E-Mails, Surfen im Web, Streaming von Videos oder Spielen von Online Games [?].

Mit „Internet of Things“ (IoT) wird eine andere Philosophie verfolgt. Es gibt keine einheitliche Definition und Abgrenzung von IoT. Grundsätzlich versucht man Objekte und Gegenstände, welche im klassischen Sinne des Internets nicht berücksichtigt wurden, ans Netz anzuschliessen. Mit minimalen menschlichen Eingriffen sollen diese Geräte Daten sammeln, austauschen und aufgrund von Software und Algorithmen Entscheidungen treffen [?]. Man spricht im Zusammenhang von „Things“ auch von „Smart Devices“ oder „Smart Objects“.

1.1.1 Smart Objects

Smart Objects oder auch „Things“ ergänzen das herkömmliche Internet um eine Vielzahl neuartiger Teilnehmer. Man ist versucht, die mit dem Internet erschaffene virtuelle Welt mit Objekten der tatsächlichen „echten“ Welt zu verbinden. Der Begriff „Smart“ ist seit der Erscheinung des iPhones weltweit bekannt. Er beschreibt die Fähigkeit eines Objekts mit dem Internet zu kommunizieren.

Während Smartphones oder Smart-TVs noch als herkömmliche Internet Terminals angesehen werden können, so erweitern die Smart Objects das bisherige Internet um eine neue Art von Teilnehmer. Smart Objects lassen sich wie folgt beschreiben:

- haben eine physikalische Repräsentation mit Eigenschaften wie Form und Grösse
- haben Mindestmass an Kommunikationsfunktionalitäten wie Request/Reply
- besitzen eine UID (unique identifier)
- haben mindestens einen Namen und eine Adresse
- besitzen ein Mindestmass an Rechenfähigkeiten
- besitzen Sensoren, um physikalische Erscheinungen wie Druck, Licht, Temperatur, etc. zu messen

Der letzte Punkt in der oberen Definition beschreibt den tatsächlichen Unterschied zu herkömmlichen Devices im Internet. Konzeptionell liegt bei IoT der Fokus mehr auf Daten und Informationen von physikalischen Objekten als bei Punkt-zu-Punkt Kommunikation von Terminals [?].

1.2 Einsatzgebiete

Das "Internet of Things" hat viele Einsatzmöglichkeiten und wir stehen erst am Anfang. Es werden immer neue Einsatzbereiche entdeckt und vorhandene optimiert und erweitert. Um die Einsatzmöglichkeiten aufzuzeigen, wird hier das Beispiel "Gesundheitsvorsorge" genauer gezeigt.

1.2.1 Allgemein

Die Gesundheitsvorsorge ist ein riesiger Bereich, welcher alle Menschen betrifft und enorme Kosten verursacht. Heutzutage wird noch sehr viel manuell erledigt. In naher Zukunft wird sich das ändern und das Internet der Dinge wird immer wichtiger, um die bestmögliche Behandlung jedes Patienten sicherzustellen.

1.2.2 Im Alltag

Selbstüberwachung In den letzten Jahren ist die Selbstüberwachung zu einem riesigen Thema geworden. Firmen wie Apple oder Fitbit haben diesen Bereich stark gefördert. So gibt es heute schon für mehrere Körperdatensensoren, die alles überwachen. Ein Beispiel dafür sind die Smartwatches. Diese können bereits heute den Puls rund um die Uhr überwachen oder sie zählen die Schritte mit. So kann jeder sein eigenes Fitnessprofil von sich erstellen. Ein weiteres Beispiel findet man bei Patienten, welche Medikamente verabreicht bekommen. Da Medikamente häufig vergessen werden, gibt es kluge Medikamentenspender, welche den Patienten benachrichtigen, wenn dieser die Medikamente nicht genommen hat.

Fernüberwachung Neben der Selbstüberwachung gibt es auch noch die Fernüberwachung. Dabei wird der Patient vom Pflegepersonal überwacht und bei kritischen Situationen kann sofort gehandelt werden. Für Rollstuhlfahrer oder Senioren gibt es zum Beispiel ein Falldetektor. Falls der Patient umfällt, reagiert der Sensor und meldet diesen Unfall direkt an das Pflegepersonal. So kann schnell reagiert und dem Patienten geholfen werden. Auch das Überwachen der Patientenwerte kann so über das Internet sichergestellt werden. Durch eine kontinuierliche Abfrage der Werte, ist die Reaktionszeit enorm geringer als beim manuellen Messen. Hierfür benötigt man Pflegepersonal vor Ort.

1.2.3 Im Spital

Patientenzimmer Durch die komplette Überwachung des Patienten in seinem Zimmer, verkürzt sich die Reaktionszeit enorm. Der Patient wird zwar schon überwacht, aber durch die Anbindung an das Internet wären weitere Verbesserungen möglich. So kann man bei gewissen Problemen nicht nur das Pflegepersonal alarmieren, sondern zum Beispiel auch direkt benötigtes Personal aus anderen Spitälern anordnen. Auch könnte man die Messdaten aus der Fernüberwachung mit der aus dem Patientenzimmer verknüpfen und so ein besseres Patientenbild erstellen.

Infrastruktur In Spitälern werden viele verschiedenen Geräte eingesetzt, welche momentan noch nicht miteinander kommunizieren. Durch die Verknüpfung der Geräte könnte man die Daten verschiedener Geräte miteinander kombinieren und vergleichen, um ein besseres Bild erstellen zu können. Oder auch die Wartung würde viel einfacher werden. Aus der Ferne werden alle Geräte inventarisiert, gewartet und überwacht. So wird bei einem Fehler sofort der Techniker gerufen, damit ein defektes Gerät sofort wieder einsatzbereit gemacht wird.

Teil II

Anhang

1. Projektplan

1.1 Projektübersicht

In diesem Projekt soll der Stand der Entwicklungen im Bereich „Internet of Things“ aufgezeigt werden. Die verschiedenen Arten der eingesetzten Sensoren sollen ermittelt- und die Einsatzgebiete untersucht werden. Heutzutage werden in der Industrie bereits verschiedenartige Sensoren eingesetzt. Die Anzahl der eingesetzten Sensoren steigt drastisch. Schon bald stellt sich die Frage, wie man mit der steigenden Anzahl Sensoren deren Management realisieren soll.

1.1.1 Zweck und Ziel

Die Bachelorarbeit soll den Nachweis der Problemlösungsfähigkeit unter Anwendung ingenieurmässiger Methoden nachweisen. Entsprechend verfügt die Arbeit über einen konzeptionellen, theoretischen und einen praktischen Anteil.

1.1.2 Projektorganisation

Vorname	Name	E-Mail
Andreas	Stalder	astalder@hsr.ch
David	Meister	dmeister@hsr.ch

Das Projekt wird von Prof. Beat Stettler und Urs Baumann betreut und benotet.

1.2 Management Abläufe

1.2.1 Zeitbudget

Der Projektstart ist am Montag, dem 20. Februar 2017.

Die Projektdauer beträgt 17 Wochen, und das Projektende ist am Freitag, dem 16. Juni 2017.

Während diesen 17 Wochen sind 360 Arbeitsstunden pro Projektmitglied eingeplant. Das entspricht pro Mitglied eine Arbeitszeit von ca. 22 Stunden pro Woche. Dies ergibt einen totalen Aufwand von ca. 720 Stunden.

Die wöchentliche Arbeitszeit von 22 Stunden kann bei Verzug oder bei unerwarteten Problemen auf maximal 30 Stunden erhöht werden.

Es sind gegenwärtig keine Absenzen während dieser Zeit geplant.

1.2.2 Projektphasen

Das Projekt wird in fünf Phasen unterteilt: Initialisierung, Analyse, Design, Realisierung und Abschluss.

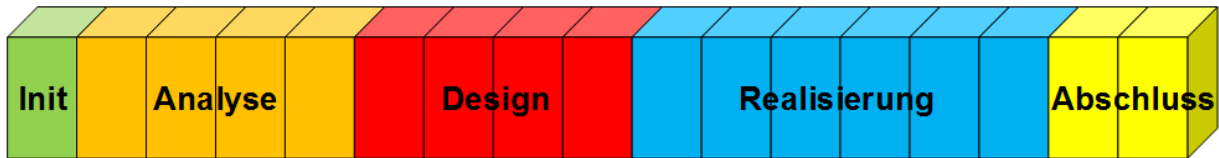


Abbildung 1.1: Projektphasen

1.2.3 Meilensteine

Das Projekt beinhaltet insgesamt vier Meilensteine.

Meilenstein	Beschreibung	Datum
MS1	Anforderungen und Scope definiert	26.03.2017
MS2	Architektur und Design beschrieben	23.04.2017
MS3	Software fertiggestellt, Codefreeze	04.06.2017
MS4	Arbeitsabgabe	16.06.2017

Tabelle 1.1: **Projekt Meilensteine**

1.2.4 Iterationen

Die Dauer eines Iterationszyklus beträgt jeweils eine Woche.

Iteration	Inhalt	Start	Ende
Initialisierung 1	Kickoff Meeting, Projektplanung, Infrastruktur	20.02.2017	26.02.2017
Analyse 1	IoT-Analyse Allgemein	27.02.2017	05.03.2017
Analyse 2	IoT-Analyse Allgemein	06.03.2017	12.03.2017
Analyse 3	Sensoren Analyse technisch & Evaluation	13.03.2017	19.03.2017
Analyse 4	Requirements definieren	20.03.2017	26.03.2017
Design 1	Architektur beschreiben, Risikoanalyse	27.03.2017	02.04.2017
Design 2	Prototyp programmieren	03.04.2017	09.04.2017
Design 3	Prototyp programmieren	10.04.2017	16.04.2017
Design 4	Requirements- und Architektur Review	17.04.2017	23.04.2017
Realisierung 1	Implementation	24.04.2017	30.04.2017
Realisierung 2	Implementation	01.05.2017	07.05.2017
Realisierung 3	Implementation	08.05.2017	14.05.2017
Realisierung 4	Implementation	15.05.2017	21.05.2017
Realisierung 5	Implementation	22.05.2017	28.05.2017
Realisierung 6	Refactoring / Bugfixing	29.05.2017	04.06.2017
Abschluss 1	Dokumentation	05.06.2017	11.06.2017
Abschluss 2	Dokumentation	12.06.2017	16.06.2017

Tabelle 1.2: **Projekt Iterationen**

1.2.5 Arbeitspakete

Name	Inhalt	Iteration	Wer	Soll	Ist
Initialisierung					
Kickoff-Meeting	Allgemeine Besprechungen zum Projektstart	Initialisierung 1	Alle	1	1
Dokumenterstellung	Erstellung L ^A T _E X Vorlagen	Initialisierung 1	Alle	4	5
Projektplan	Zeitplanung, Phasen, Meilensteine	Initialisierung 1	Alle	5	6
Einrichtung Projektmanagement Software	Installation und Einrichtung Jira	Initialisierung 1	Alle	3	5
Analyse					
Einarbeitung IoT-Allgemein	IoT-Übersicht erarbeiten und dokumentieren	Analyse 1/2	dm	25	32
Einarbeitung Sensortypen	Sensortypen recherchieren und dokumentieren	Analyse 1/2	as	15	12
Einarbeitung Kommunikation	IoT-relevante Kommunikationsbereiche analysieren und dokumentieren	Analyse 1/2	Alle	40	44
Einarbeitung Management	Device Management in IoT	Analyse 3	Alle	40	52
Requirements	Funktionale Anforderungen (Use Cases) und Nicht-funktionale Anforderungen	Analyse 4	Alle	40	29
Design					
Architekturübersicht	Übersichtsverschaffung, Grobkonzept	Design 1	dm	8	6
Domain Model	Erstellung des Domänenmodells	Design 1	Alle	8	8
Schichtenarchitektur	Erstellung der Schichtenarchitektur, Packages schnüren	Design 1	as	8	6
Auswahl Frameworks	Frameworks für Front- & Backend sowie Datenbank analysieren und auswählen	Design 1-4	Alle	60	76
Prototyp	Erstellung eines Prototypen über alle Software-schichten	Design 2-4	Alle	90	128
Realisierung					

Name	Inhalt	Iteration	Wer	Soll	Ist
Anpassungen Architektur	Anpassungen Architektur gem. Reviews	Realisierung 1	Alle	30	22
Erstellung Views	Seitenaufbau mit verschiedenen Views aufgebaut	Realisierung 1	Alle	10	14
Gruppenverwaltung implementieren	Navigationsbaum, Gruppen CRUD etc.	Realisierung 2-4	Alle	53	58
Deviceverwaltung implementieren	Anzeige, Kommunikation	Realisierung 2-4	Alle	47	37
Discovery implementieren	Implementieren Device Discovery	Realisierung 3, 5	Alle	14	16
Konfigurationsverwaltung implementieren	Implementieren Konfigurationsverwaltung	Realisierung 4-5	Alle	27	26
Login + User Management implementieren	Login + User Management implementieren implementieren	Realisierung 5	Alle	10	13
Dashboard implementieren	Login + User Management implementieren implementieren	Realisierung 5-6	Alle	18	19
Devicemap implementieren	Implementierung Google Map	Realisierung 5	Alle	4	6
Testing	Integrationstests	Realisierung 6, Abschluss 1	Alle	16	15
Refactoring	Struktur- und Codeanpassungen	Realisierung 6, Abschluss 1	Alle	32	36
Bugfixing	Beheben von Softwarefehlern	Realisierung 6	Alle	8	18
Testclient aufsetzen	LwM2M Testclient aufbauen	Realisierung 6	Alle	4	5
Abschluss					
Schlussbericht erstellen	Schlussbericht erstellen und abgeben	Abschluss 1-2	Alle	100	138
Anleitungen erstellen	Benutzer- und Installationsanleitung erstellen	Abschluss 2	Alle	12	9

Tabelle 1.3: **Arbeitspakete**

1.2.6 Teammeetings

Besprechungen finden dreimal wöchentlich jeweils an den vorgesehenen Arbeitstagen statt. Besprechungen dauern in der Regel 10-15 Minuten. Es wird das weitere Vorgehen, sowie durchgeführte Arbeiten, fällige Arbeiten und auftretende Probleme besprochen. Weiter werden Arbeitspakete verteilt, damit beide Projektmitglieder wissen was zu tun ist.

1.2.7 Meeting mit Betreuern

Die Meetings mit den Betreuern finden jeden Freitag um 14:00 Uhr statt. Die Meetings werden mit den Betreuern Prof. Beat Stettler und Urs Baumann in ihrem Büro durchgeführt. Die Meetings dauern normalerweise zwischen 30-60 Minuten.

1.3 Qualitätsmassnahmen

1.3.1 Versionierung

Wie die Dokumentation wird auch der Sourcecode mit git versioniert und auf GitHub abgelegt. Es wird darauf geachtet, möglichst häufig auf den Stamm zu commiten.

1.3.2 Reviews

Regelmässige Reviews sind in einem iterativen Vorgehen unerlässlich. Die getätigte Arbeit muss ständig abgeglichen und in Frage gestellt werden. Aus Kosten-Nutzen Sicht sind Reviews das effektivste Mittel um die geforderte Qualität zu erreichen.

In diesem Projekt werden drei verschiedene Arten von Reviews durchgeführt. Zum einen sind dies regelmässige Code Reviews, zum anderen sind dies Requirement- und Architekturreviews.

Bei den regelmässigen Code Reviews wird besonders auf die gewählten Namen (Packages, Klassen, Methoden, Variablen), die Verständlichkeit vom Code und Code Smells geachtet.

Bei den Requirement Reviews wird sichergestellt, dass man den Wünschen des Auftraggebers entsprechend entwickelt. Die Frage nach dem „Was“ wird erneut gestellt und somit sichergestellt, dass man beim Projektende nicht ein qualitativ hochwertiges Produkt entwickelt hat, welches aber nicht die Wünsche des Auftraggebers abdeckt.

Beim Architektur Review wird besonders auf die nicht-funktionalen Anforderungen (NFA) geachtet. Diese wirken sich in den meisten Fällen auf die gewählte Architektur aus. Die gewählte Architektur muss mit den NFAs verträglich sein. Wenn zu spät im Projekt bemerkt wird, dass die Architektur geändert werden muss, kann dies sehr aufwändig sein.

1.3.3 Code Metriken

Code Metriken zeigen mögliche Fehler oder Schwachstellen im entwickelten Code auf. Es wird grundsätzlich zwischen statischen- und dynamischen Metrik Tools unterschieden. Bei den dynamischen Metrik Tools wird der Code ausgeführt. Beispiele wären Unit Tests und die dazugehörige Coverage. Bei den statischen Analysetools wird der Code nicht ausgeführt. Ein Beispiel wäre Checkstyle. Dieses Tool überprüft vor allem die Einhaltung von Style Richtlinien.

1.4 Risikomanagement

1.4.1 Risiken

Nr	Titel	Beschreibung	max.	Eintrittswahrscheinlichkeit	Gewichteter	Vorbeugung	Verhalten bei Eintitt
			Schaden [h]		Schaden		
R1	Netzwerkstabilität	Netzwerkqualität reicht nicht aus um benötigte Daten zu übertragen	11	10%	1	keine Vorbeugung möglich	Einschränkungen für gewisse Funktionen definieren
R2	Performance	Applikation läuft langsam	22	10%	2	in Architektur berücksichtigen	Fehlersuche, Architektur Review
R3	Backup	Sichern von IoT Devices an Remote Destination funktioniert nicht	22	20%	4	optionales Feature, keine Vorbeugung vorgesehen	Einschränkungen für bestimmte Devicetypen definieren
R4	Implementation von Kommunikationsstandards	Kommunikation zu IoT Devices ist nicht möglich (Protokolle, Standards, etc.)	44	25%	11	Prototyp bis 16.04. klärt	Einschränkungen für Standards definieren
R5	Device Authentifizierung	Authentifizierung am Device (Password, Zertifikat) funktioniert nicht	44	25%	11	Prototyp bis 16.04. klärt	Zertifikatauthentifizierung optional
R6	Device Discovery	Automatisches Discovery von IoT Devices funktioniert nicht	44	40%	18	Prototyp bis 16.04. klärt	nur manuelles Hinzufügen von Devices hinzufügen
R7	Provisioning	Probleme bei der Verteilung von Software oder Konfigurationen an Devices	88	40%	35	Prototyp bis 16.04. klärt	Kompensation durch Überzeit, soweit als möglich implementieren
R8	Einarbeitung in neue Frameworks	Die ausgesuchten Frameworks benötigen mehr Zeit in der Einarbeitung als geplant	88	50%	44	Prototyp bis 16.04. klärt	Kompensation durch Überzeit
Summe			363		126		

Abbildung 1.2: Risikoanalyse

1.4.2 Umgang mit Risiken

Die im Dokument aufgeführten Risiken sind in der Zeitplanung nicht speziell vorgesehen. Falls beim Eintreten eines geplanten Risikos ein erhöhter Zeitbedarf entsteht, so muss dies mit hoher Wahrscheinlichkeit mit Mehrarbeit der Teammitglieder kompensiert werden. Falls die nötige Mehrarbeit ausserhalb der Möglichkeiten liegt, so muss in Absprache aller Teammitglieder mit dem Betreuer nach einer anderen Lösung (z.B. Einschränkung von Programmfeatures, etc.) gesucht werden.

2. Abbildungsverzeichnis