Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Simon Kallweit

Eine Big Data-Anwendung zum Identifizieren von Aktivitätsmustern und Sicherheitsproblemen

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Simon Kallweit

Eine Big Data-Anwendung zum Identifizieren von Aktivitätsmustern und Sicherheitsproblemen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Wirtschaftsinformatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Olaf Zukunft
Zweitgutachter: Prof. Dr. Martin Hübner

Eingereicht am: 29. April 2016

Simon Kallweit

Thema der Arbeit
Eine Big Data-Anwendung zum Identifizieren von Aktivitätsmustern und Sicherheitsproblemen

Stichworte
Big Data, Anwendung, Aktivitätsmuster, Sicherheit, Architektur, Infrastruktur, ARP

Kurzzusammenfassung
Diese Bachelorarbeit beschäftigt sich im Big Data Kontext mit der Analyse, Realisierung und Bereitstellung einer Infrastruktur. Auf die Infrastruktur setzen Anwendungen auf, die unterschiedliche Arten der Benutzung aufzeigen. Für die Datenakquisition werden ARP-Pakete in einem Netzwerk aufgezeichnet, die auf Grundlage verschiedener Szenarien analysiert und in einem Dashboard ausgewertet werden. Anhand einer Fallstudie wird die realisierte Architektur getestet und Ergebnisse erläutert.
Simon Kallweit

Title of the paper
A Big Data-Application to identify activity pattern and security issues

Keywords
Big Data, application, activity pattern, security issues, architecture, infrastructure, address resolution protocol

Abstract
In context of big data this Bachelor-Thesis describes the analysis, realization and deployment of an infrastructure. A variety of applications are attached to the infrastructure to demonstrate different usage scenarios. For data collection purposes ARP-packets are recorded in a network, analysed and illustrated in a dashboard. The realized architecture is tested bases on a case study and the results are evaluated.

Inhaltsverzeichnis

1 Einleitung **1**
- 1.1 Datenschutz . . . 2
- 1.2 Aufbau der Bachelorarbeit . . . 2
- 1.3 Zielsetzung . . . 2

2 Grundlagen **4**
- 2.1 Big Data Life Cycle . . . 4
- 2.2 Datenakquise . . . 6
  - 2.2.1 MAC-Adresse . . . 6
  - 2.2.2 ARP-Protokoll . . . 6
  - 2.2.3 Pcap4J (libpcap) . . . 8
  - 2.2.4 Raspberry Pi . . . 8

3 Anforderungen **10**
- 3.1 Szenarien . . . 10
- 3.2 Datenakquise . . . 11
  - 3.2.1 Voraussetzungen . . . 12
  - 3.2.2 Datenbank . . . 12
  - 3.2.3 Erfassungssoftware . . . 13
  - 3.2.4 Erfassungssystem . . . 14
- 3.3 Säuberung . . . 15
- 3.4 Analyse, Auswertung und Bericht . . . 16
  - 3.4.1 Analyse und Auswertung . . . 16
  - 3.4.2 Bericht . . . 18
    - 3.4.2.1 Pentaho . . . 19
    - 3.4.2.2 Jaspersoft . . . 19
    - 3.4.2.3 Evaluation . . . 20
- 3.5 Anwendungsfalldiagramm . . . 21

4 Konzept und Realisierung **23**
- 4.1 Konzept . . . 23
  - 4.1.1 Architektur . . . 23
  - 4.1.2 Erhebung der Daten . . . 23
  - 4.1.3 Säuberung . . . 27
  - 4.1.4 Analyse und Auswertung . . . 27
    - 4.1.4.1 Anwendungen . . . 27

4.1.4.2 Infrastruktur 29
4.2 Realisierung und Test 30
4.2.1 Erhebung der Daten 30
4.2.2 Säuberung 35
4.2.3 Analyse und Auswertung 35
4.2.3.1 Analyse 35
4.2.3.2 Anwendungen 40

5 Fallstudie HAW.1X **45**
5.1 Erfassung 45
5.1.1 Standort der Erfassung 45
5.1.2 IP-Adressbereich 46
5.1.3 Datenakquise 46
5.2 Säuberung 47
5.3 Auswertung 47
5.3.1 Szenario KMU 49
5.3.2 Szenario IT-Dienstleister 51
5.3.3 Szenario Franchise-Unternehmer 53
5.4 Schlussbetrachtung 55
5.4.1 Sicherheit 55
5.4.2 Kapazität 56
5.4.3 Weitere Auswertungen 56

6 Abschluss **57**
6.1 Zusammenfassung 57
6.2 Ergebnisse 58
6.3 Ausblick 59

Inhalt der CD-ROM **61**

Literaturverzeichnis **62**

Tabellenverzeichnis

3.1 Vergleich BI-Suites 20

4.1 Kennzahlen für die Szenarien 37
4.2 Erläuterung einzelner Analysen 38

5.1 Zeitraum der Erfassung 47
5.2 Eckdaten der Erfassung 48
5.3 Aufenthaltsdauer 54
5.4 Ergebnis - Vergleich Apple-Geräte mit Geräten anderer Hersteller - 28.10.2016 55

Abbildungsverzeichnis

2.1 Big Data Life Cycle anhand Jagadish u. a. (2014) 4
2.2 Abbildung anhand Plate (2015) 7

3.1 Anwendungsfalldiagramm 22

4.1 Architektur 24
4.2 UML 2.0 Aktivitätsdiagramm 26
4.3 MongoDB-Collection logs 31
4.4 MongoDB-Collection requests 31
4.5 MongoDB-Collection responses 31
4.6 CompDiscovery Klassendiagramm 34
4.7 Dashboard - Component Panel 42

5.1 Dashboard - Szenario KMU 49
5.2 Dashboard - Szenario IT-Dienstleister 51
5.3 Dashboard - Szenario Franchise-Unternehmer 53

1 Einleitung

Für WLAN-Betreiber, die ihren Kunden einen kostenlosen WLAN-Zugang zur Verfügung stellen gibt es derzeit immer noch Einschränkungen. Nichtsdestotrotz werde es in der Zukunft immer mehr Unternehmer sein, die diesen Service bereitstellen möchten. Mit der möglichen Verabschiedung der Reform des Telemediengesetzes, (vgl. Bundesregierung (2015)) vom 18.11.2015 (Änderung des §8 - WLAN-Störerhaftung und §10 - Hostproviderhaftung) würde ein weiterer kleiner Schritt in die Richtung getätigt, dass ein WLAN-Dienste-Anbieter nur teilweise und unter bestimmten Bedingungen von der Haftung ausgeschlossen werden kann.

Es gibt eine große Nachfrage nach den eben genannten öffentlich-verfügbaren-WLANs, aufgrund der in Deutschland noch nicht komplett flächendeckenden LTE-Verfügbarkeit und der Mangel an genügenden HSDPA und LTE Volumen der Mobilfunk-Verträge oder Prepaid-Karten. Eine größere Franchise-Kette im Gastronomiebereich hat diese Nachfrage bereits vor Jahren erkannt und versucht diesen Bedarf zu decken. Hieraus entstehen Vorteile für die Kunden (kostenloses WLAN) all auch das Unternehmen erhofft sich weitere positive Effekte wie z.B. Umsatzsteigerungen. Da ein kostenfreies WLAN vermutlich auch zusätzliche Kunden anziehen würde, die gerne ein Tablet oder einen Laptop nutzen ohne eingebaute UMTS-Verbindung.

Was wäre jetzt, wenn ein Unternehmer ein freies WLAN zur Verfügung stellt und damit nicht nur potenzielle Kunden ins Unternehmen lockt, sondern noch weitere positive Information daraus extrahieren kann, wie die Anzahl der Nutzer, die Länge des Aufenthaltes oder welcher Gerätehersteller bei den Kunden beliebt ist. Weiterhin könnten diese Daten mit Bondaten oder Verkaufsstatistiken erweitert werden. Dadurch ergeben sich wiederum Rückschlüsse auf eine Verbindung zwischen der Anzahl der Benutzern des WLAN und der eigentlichen Kunden.

Jedes netzwerkfähige Gerät verfügt über seine gesamte Lebensdauer über einen eindeutigen Identifier, dieser wird als MAC-Adresse (Media-Access-Control-Adresse) bezeichnet und wird meistens als eine 24-Bit-Hex-Zahlenfolge angegeben. Diese international eindeutige Zeichenfolge ist für die Adressierung eines Ethernet-basierten Netzwerkpaket von Nöten. Mit dieser Information kann somit u.U. auf eine bestimmte Person geschlossen werden,z.B. wenn es ein Mobilfunk-Gerät betrifft. Allerdings kann die Person nicht eindeutig identifiziert werden, da

die Adresse keine Rückschlüsse auf die Merkmale, wie z.B. den Namen des Inhabers, zulässt.

1.1 Datenschutz

Aufgrund der Aufzeichnung von Netzwerkpaketen spielt auch der Datenschutz in dieser Arbeit eine Rolle. Die MAC-Adressen (siehe Abschnitt 2.2.1) die in jedem Ethernet-basiertem Netzwerkpaket enthalten sind stellen laut Bundesdatenschutzgesetz (BDSG) §3 personenbezogene Daten dar. Begründet wird es damit, dass mit der MAC-Adresse praktisch eine natürliche Person eindeutig identifiziert werden könnte. Somit wäre es möglich, auf Grundlage der aufgezeichneten Daten und mit einer Zuordnung der MAC-Adresse zu dem Besitzer des Gerätes, die Dauer seines Aufenthalts im Netzwerk eindeutig zu bestimmen. Dieser Missbrauch der Daten wird in der Analyse weiter ausgeführt. Es ist allerdings nötig die Komponenten eindeutig zu identifizieren damit die Auswertungen einen Sinn ergeben. Aus diesem Grund werden die MAC-Adressen nur im Arbeitsspeicher abgelegt. Abgespeichert wird die MAC-Adresse mit SHA3 gehasht in der Datenbank. Zum Zeitpunkt der Abgabe dieser Arbeit ist es nicht möglich diesen Hash zurückzurechnen um die MAC-Adresse zu erhalten und bietet deshalb größtmöglichen Schutz gegen einen Missbrauch der Daten.

1.2 Aufbau der Bachelorarbeit

In den Grundlagen werden einige Teile, die für die erfolgreiche Erstellung der Big Data Anwendung nötig sind, erläutert. Fortgefahren wird mit den Anforderungen der einzelnen Bereiche der Arbeit und dem Vergleich unterschiedlicher Produkte und Funktionen. Im Anschluss wird ein Konzept auf Grundlage der Anforderungen erstellt und in dem Kapitel Realisierung umgesetzt. Nachfolgend wird in der Fallstudie auf die Benutzung in einem Netzwerk eingegangen und erlangte Erkenntnisse erläutert. Das abschließende Kapitel Abschluss fasst alle Ergebnisse der einzelnen Kapitel der Arbeit zusammen und gibt einen Ausblick auf Möglichkeiten der Verbesserungen für die Zukunft.

1.3 Zielsetzung

Das Ziel ist es eine Big Data Anwendung zu erstellen, die für unprivilegierte Nutzer sowohl Aktivitätsmuster als auch Sicherheitsprobleme in einem potenziell fremden, WLAN-basierten Netzwerk anzeigt. Mit Unterstützung des Big Data Life Cycle (siehe Abschnitt 2.1) wird vor-

gegangen um eine strukturierte Abfolge zu gewährleisten. Diese Schritte werden in den Anforderungen einzeln erläutert, definiert und abgearbeitet. Zuerst müssen Daten akquiriert werden um später, basierend auf diesen Grunddaten, Analysen und Auswertungen ausführen zu können. Diese Daten bestehen hauptsächlich aus den ARP-Antworten und Log-Einträgen die eine Software in einem größeren Netzwerk aufzeichnet. Die Grunddaten und etwaige andere zugefügte Metadaten werden von fehlerhafte Einträgen gesäubert. Dieser Schritte stellt nur einen kleinen aber durchaus wichtigen Teil der Arbeit dar, um möglichst genau die Ergebnisse bestimmen zu können. Die gesäuberten Daten werden mit Hilfe eines Scriptes, das alle Analysen beinhaltet, analysiert. Die Ergebnisse werden in einem Web-Dashboard zusammengefasst und zum Interpretieren bereitgestellt. Für die zeitnahe Umsetzung dieses Projektes sollte frühzeitig mit der Entwicklung und der Tests der Erfassungssoftware begonnen werden um genügend Grunddaten für die Analysen zusammenzutragen. Damit den späteren Benutzern des Projektes eine praktische Anwendung mit Ergebnissen aufgezeigt werden kann, wird das Projekt an Hand einer Fallstudie eines größeren WLAN-Netzes durchgeführt und die Ergebnisse präsentiert.

2 Grundlagen

2.1 Big Data Life Cycle

Der *Big Data Life Cycle* (*deut. Lebenszyklus*) beschreibt ein Big Data Modell, das aus fünf Phasen besteht.

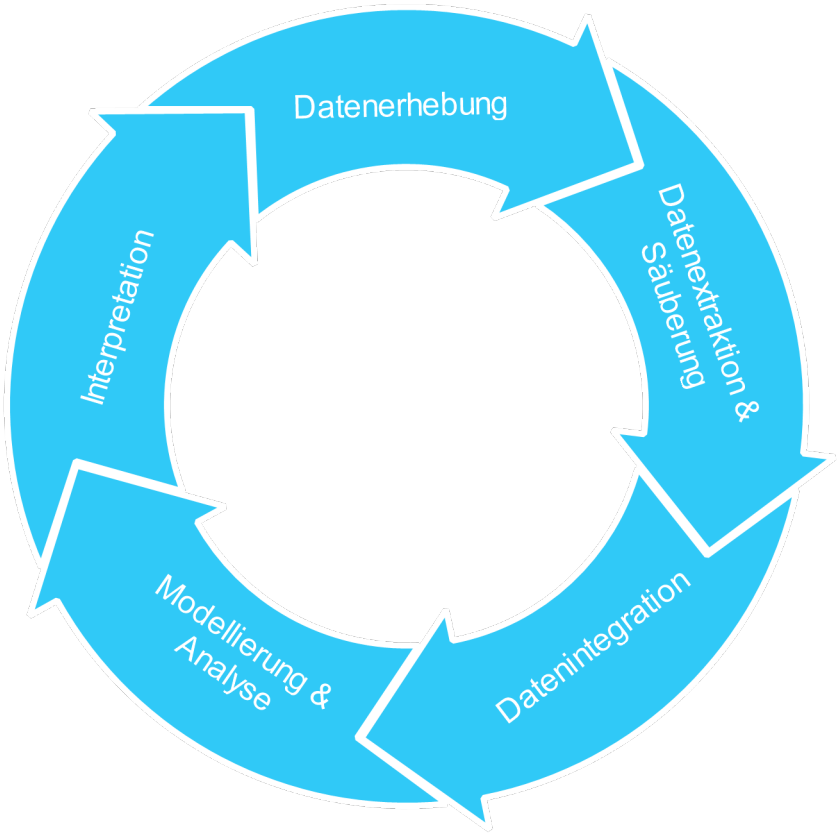


Abbildung 2.1: Big Data Life Cycle anhand Jagadish u. a. (2014)

Datenerhebung (engl. Data aquisition) In diesem Schritt geht es hauptsächlich um die Erfassung der Grunddaten, die Quelle und wie sie akquiriert wurden. Beispielsweise sind es die Ausgaben von Sensoren, Simulationen oder Teleskope, die mehrere Terrabyte an Daten pro Tag verursachen. Sie werden hier u. U. auch gefiltert und komprimiert.

Datenextraktion und Säuberung (engl. Information extraction and cleaning) Die Daten müssen in dem Schritt aus den unterschiedlichen Quellen strukturiert für die Analyse aufbereitet werden. Das beinhaltet das Auslesen der wichtigen Daten aus unstrukturierten oder heterogenen Quellen. Weiterhin werden die aufbereiteten Daten gleichzeitig gesäubert, sofern falsche Werte aus unzuverlässigen Datenquellen in die Erfassung eingeflossen sind. Dies könnte bei Sensoren der Falls sein, die kurzfristig nicht erreicht werden oder fehlerhafte Ereignisse aufzeichnen.

Datenintegration (engl. Data integration, aggregation and representation) Da die gesäuberten Daten aus unterschiedlichen Quellen kommen und somit nicht immer eindeutig zu unterscheiden sind, werden in dieser Phase weitere Metadaten hinzugezogen, um die Daten besser analysieren zu können. Beispielsweise die Standorte der Sensoren.

Modellierung und Analyse (engl. Modeling and analysis) Abfragen in Big Data unterscheiden sich grundlegend von den traditionellen statischen Analysen von kleinen Beispielen. Aufgrund dessen muss in dieser Phase mit angebrachter statistischer Sorgfalt gearbeitet werden um gute Resultate aus der immensen Fülle von Daten zu gewinnen.

Interpretation (engl. Interpretation) In der letzten Phasen des Modells wird die Analyse zum Interpretieren bereitgestellt. Hier sind gleich mehrere Dinge zu beachten, es müssen die Annahmen in den unterschiedlichen Stadien der Analyse überprüft werden; es reicht nicht einfach nur die Ergebnisse zu interpretieren, da es auch zu Fehlern bei der Aufzeichnung, Verarbeitung oder bei der Definition der Analysen gekommen sein kann.

Es wird immer mit der Datenerhebung begonnen und danach alle weiteren vier Phasen in Reihenfolge einzeln durchlaufen. Beim Interpretieren der Daten kann es zu weiteren Auffälligkeiten oder Fragen kommen. Mit den erzielten Ergebnissen wäre es möglich einen neuen Big Data Lebenszyklus zu durchlaufen, um mögliche weitere Muster aus den schon analysierten Daten zu erkennen. (vgl. Jagadish u. a. (2014))

2.2 Datenakquise

2.2.1 MAC-Adresse

Um Netzwerkgeräte eindeutig identifizieren zu können, besitzen alle Komponenten theoretisch eine weltweit eindeutige physikalische Hardwareadresse oder auch Media-Access-Control-Adresse, kurz MAC-Adresse genannt. Viele Hersteller haben unterschiedliche Bezeichnungen für eine MAC-Adresse, zum Beispiel bezeichnet *Microsoft* sie in ihren Produkten als physikalische Adresse und *Apple* als Airport-ID oder Ethernet-ID. Mit einigen Netzwerkadaptern ist es möglich per Treibereinstellungen die MAC-Adresse zu verändern, deshalb ist die MAC-Adresse nur theoretisch weltweit eindeutig. Bei dem Großteil der Komponenten ist sie allerdings fest über den gesamten Lebenszeitraum geschrieben. Die MAC-Adresse in Ethernet-Netzwerken umfasst 48 Bits, wobei die ersten 24 Bits (OUI - Organizationally Unique Identifier) für den Hersteller fest, durch die *IEEE* (vgl. IEEE (2015)) vergeben sind. Die zweiten 24 Bits werden durch den Hersteller der Komponenten festgelegt. In dem OSI-Modell wird die MAC-Adresse der Schicht 2, der Sicherungsschicht, zugeordnet. Es gibt unterschiedliche Darstellungsarten der Hardwareadresse, heutzutage wird sie größtenteils in hexadezimaler Form angegeben, wobei davon die ersten beiden der folgenden Darstellungsarten am meisten genutzt werden:

- 12:34:56:78:9A:BC
- 12-34-56-78-9A-BC
- 123456789ABC
- 1234.5678.9ABC

2.2.2 ARP-Protokoll

Das ARP-Protokoll ist ein Netzwerkprotokoll, das im Jahr 1982 (vgl. Plummer (1982)), entwickelt wurde. Im OSI-Schichtenmodell ist es in der Schicht 2, der Sicherungsschicht, angesiedelt. Die Hauptaufgabe dieses Protokolls ist es zu einer IPv4-Adresse die dazugehörige physikalische Hardwareadresse zu ermitteln, damit ein Paket zwischen zwei Netzwerkgeräten über mehrere Netzwerkweichen oder Verteiler geschickt werden kann. Dazu schickt es ein Paket an die Broadcast-Adresse. Sobald die Antwort eingetroffen ist, wird die Empfänger-MAC-Adresse in einer ARP-Tabelle, auch ARP-Cache genannt, im Betriebssystem für einen bestimmten kurzen Zeitraum (wenige Minuten) abgelegt. Dies ist von Vorteil damit nicht vor dem Versand eines Paketes jedes Mal die MAC-Adresse neu akquiriert werden muss. Die ARP-Anfrage besteht aus 60 Byte und die Antwort aus 48 Byte. Die Zusammensetzung der Pakete wird in der Abbildung

2.2 näher erläutert.

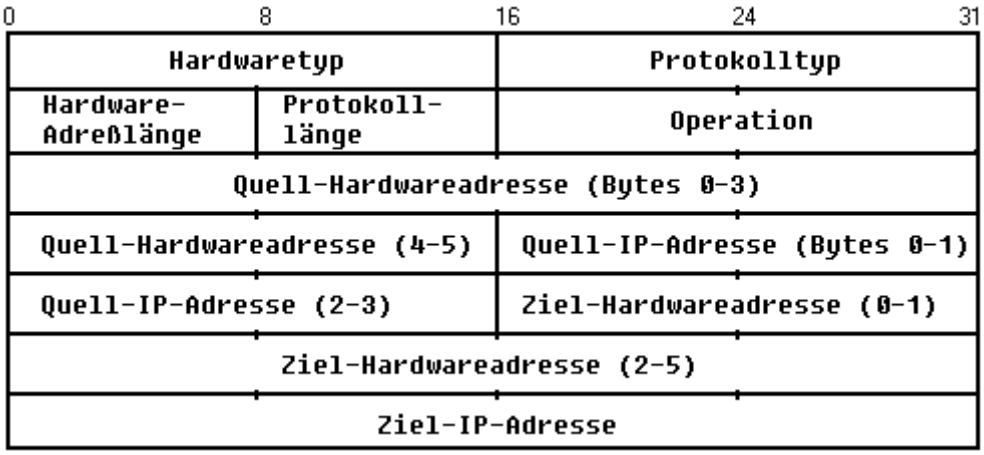


Abbildung 2.2: Abbildung anhand Plate (2015)

Auf Grund der Vorschrift (siehe Abbildung 2.2) werden die Felder für eine ARP-Anfrage beispielsweise folgendermaßen gefüllt:

Hardwaretyp	Ethernet = Wert(1)
Protokolltyp	IP = Wert(0x0800)
Hardware-Adresslänge	Wert(6)
Operation	Request = Wert(1)
Quell-Hardwareadresse	MAC-Adresse = Wert(aa:bb:cc:dd:ee:ff) // Client-MAC-Adresse
Quell-IP-Adresse	IP-Adresse = Wert(192.168.10.1)
Ziel-Hardwareadresse	MAC-Adresse = Wert(ff:ff:ff:ff:ff:ff) // Broadcast
Ziel-IP-Adresse	IP-Adresse = Wert(192.168.10.2)

Die angesprochene Netzwerkkomponente antwortet auf die ARP-Anfrage mit folgendem Paket:

Hardwaretyp	Ethernet = Wert(1)
Protokolltyp	IP = Wert(0x0800)
Hardware-Adresslänge	Wert(6)
Operation	Reply = Wert(2)
Quell-Hardwareadresse	Mac-Adresse = Wert(11:22:33:44:55:66) // Client-MAC-Adresse
Quell-IP-Adresse	IP-Adresse = Wert(192.168.10.2)
Ziel-Hardwareadresse	MAC-Adresse = Wert(aa:bb:cc:dd:ee:ff)
Ziel-IP-Adresse	IP-Adresse = Wert(192.168.10.1)

Beim Ankommen des Paketes wird die Quell-Hardwareadresse zusammen mit der IP-Adresse in den ARP-Cache aufgenommen. In der letzten Zeit kam es immer häufiger zu Sicherheitsproblemen in Verbindung mit dem ARP-Protokoll. Mit ARP-Spoofing versucht ein Angreifer einem Opfer zu einer IP-Adresse eine falsche MAC-Adresse "unterzujubeln", damit das Opfer seinen Datenverkehr an eine andere Hardwareadresse verschickt. (vgl. Plate (2015))

2.2.3 Pcap4J (libpcap)

Pcap4J ist eine Java Bibliothek und fungiert als Wrapper zwischen Java und libpcap. Lizensiert ist sie unter der *MIT License*. Libpcap ist lizensiert unter der *BSD Lizenz* und wurde für Unix-Systeme von den Entwicklern aus der *Network Research Group* des *Lawrence Berkley Laboratory* entwickelt. Die Funktion ist darauf ausgelegt Netzwerkpakete in einem Ethernet-basiertem Netzwerk aufzuzeichnen, zu erstellen und zu verschicken. Derzeit wird das Projekt von den Entwicklern von Tcpdump weiterentwickelt. Weiterhin existiert auch eine Windows-Version der libpcap und nennt sich WinPcap. Mit der Pcap4J-Library, entwickelt von *Kaito Yamada* (vgl. Yamada (2015)), kann sowohl mit libpcap als auch mit WinPcap gearbeitet werden, je nachdem auf welchem Betriebssystem die Software ausgeführt wird. (Quelle)

2.2.4 Raspberry Pi

Der *Raspberry Pi* ist ein Einplatinencomputer, von dem mehrere Versionen existieren. In dieser Arbeit wird ein *Raspberry Pi 1 Model B* (vgl. RASPBERRY PI FOUNDATION (2015a)) verwendet um die Datenakquise zu realisieren. Es ist ein Single-Core ARMv6 Prozessor mit 700 Mhz und 512 MB SDRAM verbaut. Weiterhin sind zwei USB-Ports (USB-Version 2.0) und ein 10/100-Mbit-Netzwerkinterface auf der Platine integriert. Als nicht-flüchtiger Speicher für das Betriebssystem können verschiedene Speicherkarten über ein SD-Kartenlesegerät für SDHC, SDXC, MMC-Karten mit Adapter verwendet werden. Für den Fall, dass das Betriebssystem nicht mehr reagiert und dadurch gegebenenfalls keine Anfrage von außen beantwortet werden

kann, ist ein Hardware-Watchdog integriert. (vgl. RASPBERRY PI FOUNDATION (2015b)) Für die Unterstützung von WLANs wurde von *EDiMAX* ein 300 Mbit Wireless USB Adapter (vgl. Edimax (2015)) verwendet. Als Betriebssystem können mehrere Open-Source-Betriebssysteme verwendet werden, wobei Raspbian die empfohlene Distribution des Herstellers ist und in der Arbeit benutzt wird.

3 Anforderungen

Diese Kapitel beschreibt die Erstellung von Anforderung an diese Arbeit. Um einen Überblick zu bekommen für welche Zwecke diese Arbeit eingesetzt werden könnte, wird nachfolgend mit verschiedenen Szenarien begonnen. Fortgesetzt wird nach dem *Big Data Life Cycle*, mit den Anforderungsteilen der Erfassung, der Säuberung und der Auswertung in Form eines Dashboards. Für die bessere Visualisierung wird ein Anwendungsfalldiagramm für die anfallenden Aufgaben erstellt.

3.1 Szenarien

Um ein möglichst großes Einsatzgebiet abzudecken werden einige Szenarien, die umgesetzt werden können, beschrieben. Diese Arbeit könnte u. U. auch für Missbrauch zweckentfremdet werden. Zwei, dieser möglichen Missbrauchsszenarien, werden dazu näher erläutert.

Szenario KMU: Eine kleine bis mittelständische Firma möchte ihr eigenes, allerdings nicht selbstständig administrierte, Firmennetzwerk überwachen um Rückschlüsse auf Aktivität und Leistung zu ziehen. Weiterhin wäre es sinnvoll zu prüfen, ob Geräte außerhalb der Arbeitszeit laufen, um eventuelle Stromverbrauchsquellen zu finden.

Szenario IT-Dienstleister: Ein IT-Netzwerk-Dienstleister möchte über einen längeren Zeitraum bei einem Kunden freie IP-Adressbereiche, die nicht genutzt werden, finden. Gleichzeitig sollte das Netzwerk auf etwaige Sicherheitsprobleme überprüft werden.

Szenario Franchise-Unternehmer: Ein Franchise-Unternehmer, der ein öffentliches WLAN für seine Laufkundschaft anbietet, möchte Rückschlüsse auf Geräte der Kunden ziehen, um spezifische Werbung im Unternehmen zu schalten. Weiterhin ist eine Nachfrage nach der Verbindungsdauer der Kunden, sprich den Aufenthalt im Unternehmen, dar.

Missbrauchsszenario 1: Nachfrage nach den Bewegungsmustern ohne eindeutige Bestimmung einer einzelnen Person in mehreren Standorten oder Etagen. Es existiert ein Netzwerk

mit verschiedenen DHCP-Servern pro Standort oder Etage einer Firma, die IP-Adressen aus unterschiedlichen Subnetzen verteilen, durch das Routing dennoch zusammen zu einem Netzwerk gehören. Hier wäre es möglich Bewegungsmuster aufzuzeichnen, mit der Voraussetzung, dass alle Mitarbeiter beispielsweise ihre Firmenhandys durchgängig im WLAN-Netzwerk registriert haben und diese immer bei sich führen.

Missbrauchsszenario 2: Heutzutage hat fast jeder Mensch ein Handy, viele auch schon ein Smartphone. Annahme für dieses Szenario, es wird für alle Flüchtlingsunterkünfte ein WLAN bereitgestellt, da die Asylsuchende mit ihren Verwandten in der Heimat bzw. in anderen Flüchtlingsunterkünften kommunizieren möchten, jedoch ohne einen festen Wohnsitz und einwandfreien Schufaeintrag keinen Mobilfunkvertrag bei den großen Providern abschließen können. Da es in Vergangenheit öfter vorgekommen ist, dass Flüchtlinge aus Unterkünften "verschwinden", könnten hiermit Bewegungsmuster festgestellt werden, um den Verantwortlichen zu zeigen, in welche anderen Unterkünfte betroffene Personen umgezogen sind.

Bei diesen beiden Szenarien wäre der Missbrauch noch wesentlich höher, wenn die zuständigen Administratoren, zu jeder MAC-Adresse den Namen des Besitzers wüssten. Eine mögliche Begründung wäre, um die Sicherheit des WLAN zu erhöhen wird ein MAC-Filter im WLAN eingesetzt. Hiermit könnten u.a. Zeiten im WLAN aufgezeichnet und mit den erfassten Arbeitszeiten der Mitarbeiter verglichen werden, um mögliche Falscheingaben zu erkennen. Das setzt allerdings voraus, dass die Mitarbeiter von der Erfassung nichts wüssten und immer ihre WLAN-Komponente am Körper führen.

Um in Zukunft mehrere Szenarien gleichzeitig einsetzen und möglichst anpassungsfähig auf Änderungen der Anforderungen reagieren zu können, sollte folglich eine flexible Architektur für dieses Projekt verwendet werden. Eine Trennung zwischen der Erfassung und der Analyse wäre hier angebracht.

3.2 Datenakquise

Die Datenakquise beschreibt, in welcher Form die, zu analysierende, Daten aufgezeichnet und gespeichert werden können.

3.2.1 Voraussetzungen

Wie vorab in der Zielsetzung erläutert, ist es für dieses Projekt wichtig, dass die Erhebung der Daten bereits frühzeitig durchgeführt wird, damit eine größtmögliche Anzahl von Daten erfasst und das Hauptaugenmerk auf die Analyse der Daten gelegt werden kann. Die folgenden Voraussetzungen sind wichtige Aspekte bei der Erfassung:

- Ununterbrochene Lauffähigkeit (24/7)
  - Stromversorgung
  - Netzwerkverfügbarkeit
  - Leistungsfähigkeit der Software
- Standort
- Sicherheit
- Plattformunabhängigkeit

3.2.2 Datenbank

Eine Datenbank soll alle ARP-Antworten und Logeinträge der Erfassungssoftware speichern. Wie eben beschrieben, kann davon ausgegangen werden, dass eine immense Anzahl von Daten zusammenkommt, die die Datenbank zu handhaben hat. Die Definition der fünf Datenbankmodelle nach *B.Scofield* (vgl. Scofield (2010)) kann hier für die Auswahl der richtigen Datenbank verwendet werden. Für die vorgesehen Nutzung werden diese Datenbankentypen verglichen:

- Key-Value Stores (Redis, Dynamo, ...)
- Column-Oriented Stores (HBase, BigTable, ...)
- Document-Oriented Stores (MongoDB, Riak, ...)
- Graph Databases (Neo4J, ...)
- Relational Databases (MS SQL, MySQL, FirebirdSQL, ...)

Laut Scofield (2010) haben die Key-Value Stores (deut. Key-Value-Datenbanken) eine hohe Performance, falls nach einem Schlüssel für einen Datensatz gesucht werden soll. Weiterhin bieten sie eine sehr gute Skalierbarkeit und Flexibilität, sie sind aber eher darauf ausgelegt zu einem festdefinierten Schlüssel einen Wert zu speichern. Eingesetzt werden sie beispielsweise

zum Handhaben von Session-IDs während einer Webseiten-Sitzung. Die Column-Oriented Stores (deut. spaltenorientierte Datenbank) bieten gegenüber den Key-Value Stores ähnlich gute Eigenschaften, sie sind allerdings eher auf die Verwendung von vielen Spalten anstelle von vielen Datensätzen ausgelegt. Die dokumentenorientieren Datenbanken arbeiten, wie die spaltenorientieren Datenbanken, mit halbstrukturierten Daten. Teilweise sind dokumentenorientierte Datenbanken nicht so skalierbar wie andere Datenbankmodelle, bieten allerdings eine sehr hohe Leistung und Flexibilität. Diese Datenbanken sind u.a. ausgelegt um Datensätze mit unterschiedlichen Eigenschaften in einer Tabelle/Collection zusammenzufassen, wie z.B. Log-Dateien von unterschiedlichen Anwendungen. Die vierte Art sind die Graphdatenbanken, die Knoten und ihre Beziehungen zueinander beschreiben. Die letzte und allgemein bekannteste Art sind die relationalen Datenbanken. Hier müsste sehr strukturiert und aufwändig vorgegangen werden, um die Daten in die 3. Normalform zu übernehmen und in die Datenbank zu schreiben. Beim Auslesen müssten zeitlich aufwändige Joins erstellt werden, um die Daten wieder zu verknüpfen. Weiterhin sind die spaltenorientieren und dokumentenorientieren Datenbanken schemafrei, das wäre ein weiterer Vorteil bzgl. der fortlaufenden Entwicklung der Software.

3.2.3 Erfassungssoftware

Da die ununterbrochene Lauffähigkeit eine der wichtigsten Voraussetzungen für die Datenakquise darstellt, muss selbstverständlich die Erfassungssoftware diese Leistung auch erbringen. Das bedeutet u.a., dass folgende Eigenschaften gegeben sein müssen:

- Automatischer Versand von ARP-Anfragen
- Erfassung und Speicherung von ARP-Antworten
- Performante Datenbank
- Korrektes Speicherhandling (keine Überläufe)
- Stabiles Betriebssystem

Die Frage nach Eigenfertigung oder Fremdbezug (kurz Make-or-Buy) stellt sich der Datenakquise dieser Arbeit nicht. Die erforderlichen Eigenschaften stimmen mit keinem auf dem Markt angebotenem Produkt überein und daher muss von einer Eigenentwicklung ausgegangen werden.

Aufgrund der vorher festgelegten Voraussetzungen bzgl. der Plattformunabhängigkeit sollte die Erfassungssoftware so erstellt werden, dass keine Betriebssystemfunktionen verwendet werden, die nicht in allen Betriebssystemen enthalten sind. Weiterhin müssen die Komponenten und Bibliotheken die benutzt werden sowohl für Windows als auch für Linux frei verfügbar sein.

Die Programmiersprache Java bietet nicht die nötigen Bibliotheken um in einem Netzwerk ARP-Pakete aufzuzeichnen, zu erstellen und zu verschicken. Dafür kann auf die Bibliothek *libpcap* (vgl. 2.2.3) zurückgegriffen werden. Diese bietet eine C/C++ Bibliothek an um auf einem Linux- oder auch Windows-Betriebssystem Pakete aufzuzeichnen. Um diese Bibliothek in Java zu nutzen, wird ein Wrapper wie z.B. Pcap4j oder jpcap benötigt. Beide Wrapper sind Open Source und werden weiterentwickelt, sobald eine neue Version der Bibliothek veröffentlicht wird.

Um die gesammlten MAC-Adressen einem Hersteller zuordnen zu können, sollte die OUI-Liste von der IEEE in der Erfassungssoftware verwendet werden. Diese Text-Datei beinhaltet, wie in 2.2.1 beschreiben, eine Liste mit allen Herstellern, die einen Bereich von MAC-Adressen für sich reserviert haben. Nach der ARP-Antwort von einer entsprechenden Netzwerkkomponente könnten die ersten 24 Bits in der Liste gesucht werden, um den Hersteller mit der Antwort abzuspeichern. Dies ist nötig, da die MAC-Adresse gehasht in der Tabelle gespeichert wird und im nachhinein kein Abgleich auf den Hersteller möglich ist.

3.2.4 Erfassungssystem

Für das System, auf dem die Erfassungssoftware ausgeführt wird, gibt es mehrere Lösungen. Zum einen wäre es möglich einen Server zu verwenden, mit einem Server-Betriebssystem, der per Ethernet-Kabel oder WLAN-Komponente mit einem Netzwerk verbunden ist. Das wäre eine sehr statische und nicht flexible Lösung, da bei einem WLAN der Server auch einen Standort und Platz nahe des Zugriffspunktes benötigt, um genügend Signalstärke benutzen zu können. Weiterhin ist ein richtiger Server mit mehreren Gigabyte Arbeitsspeicher und vielen Prozessoren für die Erfassung überdimensioniert. Eine andere Möglichkeit wäre einen Client-PC zu verwenden der bereits im zu durchsuchenden Netzwerk verfügbar ist und der sich in Benutzung befindet. Auf diesem würde die Erfassungssoftware und die Datenbank aufgespielt werden. Hier gibt es gleich eine Reihe von Problemen. Sobald der Sachbearbeiter aus einem bestimmten Grund das Betriebssystem neu startet ist die durchgängige Erfassung nicht mehr gegeben. Um einiges schlimmer wäre es wenn das Betriebssystem aufgrund der Erfassung viel

Systemleistung abgeben muss und dadurch sich die Wartezeit bei den Aufgaben des Mitarbeiter erhöht. Das Sicherheitsproblem ist allerdings das Größte, da der Mitarbeiter theoretisch direkt auf die Datenbankdateien zugreifen kann. Die dritte Möglichkeit wäre, ein kleines System, z.B. einen Einplatinencomputer (Raspberry Pi o.ä.). Dieses Gerät benötigt nicht viel Platz, es kann sehr schnell der Standort geändert werden und dieser würde dennoch genug Systemleistung beinhalten um die Akquise zu realisieren. Der finanzielle Aufwand ist um einiges geringer als bei einem extra für diesen Zwecke angeschaffter Server. Viele der "Single Board Computer" enthalten von Haus aus ein Ethernet-Netzwerkadapter und USB-Anschlüsse. Somit wäre es auch ohne viel Aufwand möglich, dort einen WLAN-Netzwerkadapter per USB nachzurüsten.

Als Betriebssysteme wäre für den Server- und die Clientlösung sowohl eine Microsoft-Windows-Version als auch eine Linux-Distribution möglich, bei einem Einplatinencomputer allerdings größtenteils nur Linux. Wenn auch hier die finanziellen Aufwände gegeneinander abgewogen werden und von einem noch nicht verfügbaren System ausgegangen wird, wäre die Verwendung von Linux günstiger. Für den Fall, dass die Anwendung keinen grafischen Desktop benötigt kann dieser bei Linux gelöscht werden, um weitere Systemressourcen zur Verfügung zu haben. Der Aufwand würde sich allerdings nur bei einem Single Board Computer lohnen, da diese mit viel weniger CPU-Leistung als Server arbeiten und nur eine Speicherkarte als Speichermedium für das Betriebssystem besitzen.

3.3 Säuberung

In dem *Big Data Life Cycle* ist es ein wichtiger Bereich die Datensäuberung, Punkt 2. Wie schon in den Grundlagen erläutert müssen in diesem Bereich falsche oder fehlerhafte Daten entfernt werden. Da für diese Arbeit eine Software zum Erfassen der Komponenten erstellt wird sind alle Daten, die in die Datenbank geschrieben werden, strukturell fehlerfrei. Jede Antwort wird nur geschrieben, wenn alle dazugehörigen Informationen vorhanden sind. Dennoch könnten Fehler im Ablauf verursacht worden sein oder bestimmte Informationen müssen korrigiert werden.

Die Säuberung bezieht sich in dieser Arbeit und von den erfassten Daten auf die Netzwerkkomponenten, die durchgehend im Netzwerk antworten. Das schließt die Router, Server und Arbeitsplätze, die 24 Stunden am Tag mit dem Netzwerk verbunden sind, ein. Bei einigen Szenarien ist es angebracht diese Geräte in der Analyse zu belassen, falls die Verfügbarkeit von Servern überprüft werden soll. Bei dem Großtei der Analysen ist es aber angebracht diese

Daten zu entfernen oder auszublenden. Das würde möglicherweise, je nachdem aus welchen Komponenten das Netzwerk besteht, auch bei der Analyse einen Performanceunterschied machen.

Wie unter 3.2.3 erläutert kann die IEEE-OUI-Liste für die Zuordnung der MAC-Adresse zu einem Hersteller verwendet werden. Die Säuberung bezieht sich auf die Prüfung ob die Daten in einem korrekten Format in der Text-Datei gespeichert wurden. Weiterhin wäre es nötig während des Programmstarts der Erfassungssoftware eine Prüfung auf Plausibilität durchgeführt werden.

3.4 Analyse, Auswertung und Bericht

Die Analyse und die Zusammenfassung in einem Bericht gehören zum größten Teil zusammen, da in der Berichtssoftware der Teil der Analyse durchgeführt wird. Die Berichtssoftware ist eine Business Intelligence Softwarelösung die diese Aufgabe übernehmen soll. In diesem Unterkapitel wird sowohl die eigentliche Analyse als auch der Vergleich von Business Intelligence Software Lösungen beschrieben. Um eine bessere und strukturierte Übersicht zu erhalten werden diese beiden Bereiche getrennt.

3.4.1 Analyse und Auswertung

Um auch während der Testphase die fehlerfreie Datenakquise sicherzustellen sind einfachere Analysen von Nöten. Hierzu würde es folgende Fragen geben um die Lauffähigkeit und Fehlerfreiheit zu überprüfen:

- Wie viele Antworten insgesamt gespeichert worden?
- Wie viele Suchdurchläufe gab es?
- Wie ist die eindeutige Anzahl der erfassten Komponenten?
- Wie viele und welche Hersteller produzierten die Komponenten?
- Wie hoch ist die Anzahl der Geräte pro Stunde?

Ein Teil dieser Fragen können für einzelne Szenarien verwendet werden. Für die Anwendung, bzgl. des Senarios IT-Dienstleister, wären es Fragen nach der Anzahl der erfassten Geräten oder

auch ob es Geräte gibt die sequenziell das Netzwerk durchsuchen. Weiterhin wäre zu prüfen wann bestimmte Analysen durchgeführt werden sollen. Durch die Verlegung der Analyse und Auswertungen aus der Berichtssoftware in die Infrastruktur wären einige Auswertungen um einiges performanter. Diese Frage würde sich eher im Berichtsteil stellen, wenn zwei Anwendungen verglichen werden.

Für das Szenario KMU, die kleine bis mittelständische Firma, kann die Erfassung der Daten in einem größeren Intervall durchgeführt werden, da es nicht nötig ist bestimmte kurzzeitig im Netzwerk verfügbaren Geräte zu erkennen. Dies kann sich auch ändern, falls ein WLAN für die privaten Geräte (Stichwort BYOD) der Mitarbeiter bereitgestellt wird. Allerdings sollte der Durchlauf nicht auf eine Zeitspanne eingeschränkt werden, sondern ununterbrochen ausgeführt werden. Dies hat den wesentlichen Vorteil, dass auch Geräte die über Nacht laufen, wenn die Firmenmitarbeiter nicht im Betrieb sind, aufgezeichnet werden, um so Stromverbraucher zu ermitteln. Die Auswertung in dem Bericht sollten folgende Ergebnisse liefern:

- Anzahl der täglichen Geräte
- Anzahl der IP-Adressen, die fest vergeben sind
- Anzahl der nicht vergebenen IP-Adressen
- Geräte, die außerhalb der Arbeitszeiten antworten

Im Szenario IT-Dienstleister sollte die Datenakquise auf einen Intervall eingestellt werden, damit auch die Geräte erfasst werden, die innerhalb einer kurzen Zeitspanne angeschaltet und benutzt werden. Dafür müsste nicht außerhalb der Arbeitszeiten gesucht werden, da vorausgesetzt werden kann, dass es keine Komponenten gibt, die z.B. nur Nachts ausgeführt werden. Zusätzlich wäre es aus Sicherheitsgründen angebracht zu prüfen, ob bestimmte Geräte eine ähnliche Aufzeichnung im Sinne dieser Arbeit durchführen. Für diese Prüfung ist es nötig auch alle ARP-Anfragen aufzuzeichnen. Aus diesen akquirierten Paketen könnten weitere Fehlkonfigurationen oder Sicherheitsprobleme lokalisiert werden. Wichtige Kennzahlen wären für den IT-Dienstleister in diesem Fall:

- Anzahl der IP-Adressen, die fest vergeben sind
- Anzahl der nicht vergebenen IP-Adressen
- Geräte, die sequenziell das Netzwerk durchsuchen

- Abgleich der IP-Adressen und des MAC-Hashs

Für einen Unternehmer, der ein Ladengeschäft betreibt, ist eine wichtige Kennzahl wann, wie viele und welche Geräte sich in sein WLAN anmelden. Als Beispiel hierfür wäre ein Franchise-Unternehmer mit Öffnungszeiten von 6 bis 22 Uhr, nur in dieser Zeit ist das WLAN für die Kundschaft aktiv. Die Datenakquise der WLAN-Komponenten wird auf diesen Zeitraum begrenzt, allerdings soll in einem kürzeren Intervall gesucht werden, um die Aufenthaltsdauer so genau wie möglich bestimmen zu können. Um danach betriebswirtschaftliche Erkenntnisse zum Kaufverhalten erheben zu können, müssen die Daten mit den Käufen, in einem neuen Durchlauf des *Big Data Life Cycles*, zusammengeführt werden, darauf wird aber in dieser Arbeit nicht weiter eingegangen. Wichtige Auswertungen für den Unternehmer wären:

- pro Zeitraum (täglich und stündlich)
  - Anzahl der Geräte
  - Anzahl der Hersteller
  - Anzahl der Geräte pro Hersteller
- Aufenthaltsdauer

Einige Auswertungen sind um einiges komplexer, die sich nicht unbedingt durch normale Abfragen realisieren lassen können. Hierzu wäre zu prüfen, ob diese Art von Auswertungen ohne weitere Hilfe durch andere Software durchgeführt werden können. Beispiele wären für diese komplexeren Arten von Anfragen folgende:

- Wann sind die Ankunftszeiten, die meisten “neuen” Komponenten pro Zeitbereich?
- Gibt es Komponenten, die fast immer “zusammen erscheinen”?
- Gibt es Komponenten, die den gleichen Hersteller haben und “früher gehen”?
- Gibt es Komponenten, die sequenziell das Netzwerk durchsuchen?

3.4.2 Bericht

Der Bericht soll in Form von einem Dashboard erstelle Analysen und Auswertungen anzeigen. Wenn möglich sollte es als Webseite verfügbar sein und wie beim Armaturenbrett im Auto möglichst viele Auswertungsergebnisse auf einen Blick anzeigen. Es wird hier keine Eigenentwicklung durchgeführt, sondern im Nachfolgenden wird dazu Business Intelligence Software,

fortlaufend BI-Suites genannt, verglichen. Grundvoraussetzung ist, dass der Teil kostenlos und nach Möglichkeit auch OpenSource verfügbar ist. Weiterhin sollte die Anwendung möglichst einfach zu bedienen sein damit auch in Zukunft ein Benutzer manuelle Änderungen, neue Anwendungen oder Szenarien definieren und integrieren kann. Auf zwei Business Intelligence Suites, die viele Anwendungen für die Verwendung mit Big Data zusammenfassen, werden in diesem Kapitel genauer betrachtet, analysiert und evaluiert.

3.4.2.1 Pentaho

Die Firma *Pentaho* gehört zur *Hitachi Group* und wurde 2004 gegründet. Seit 2005 arbeitet Sie an der gleichnamigen Software für die Verwendung im Big Data Bereich. Der Auftrag von *Pentaho* ist es für den Kunden eine Möglichkeit bereit zu stellen um schnellstmöglich Kennzahlen aus den eigenen Daten, ob in relationalen, cloudbasierten oder BigData-Stores, zu erhalten. *Pentaho* biete eine Data-Analysitcs-Plattform, die sowohl Data-Integration und Data-Analytics-Tool von Reporting über Ad-hoc-Disovery bis hin zu Predictive Analytics beinhaltet. Die Basis-Version mit eingeschränkter Funktionalität ist Open-Source und ist vollständig mit Java entwickelt worden. *Pentaho Community* ist die Variante der Software, die kostenlos zur Verfügung gestellt wird. Diese hat eingeschränkte Funktionalität, da die Firma noch weitere kostenpflichtige Versionen für den Kunden anbietet. Mit der Hinsicht auf eine schnelle Installation und Konfiguration bietet *Pentaho* hier einen Download an, der sowohl für Linux als auch Windows gilt. Nachdem die Datei entpackt wurde, braucht nur eine Datei ausgeführt werden und es kann auf die Oberfläche im Browser zugegriffen werden.

3.4.2.2 Jaspersoft

Der Mitbegründer von *Jaspersoft, Teodor Danciu*, begann 2001 damit, eine Berichtsfunktion in eine Anwendung einzubetten. Da auf dem Markt keine, für seine Zwecke verfügbare oder zu teure Software existierte, startete er eine Eigenentwicklung. Das Ergebnis daraus war *JasperReports: J* für Java als Open Source. Nach einigen Weiterentwicklungen auch unter Mithilfe der Community, schlossen sich einige Entwickler zusammen und gründeten 2004 die Firma *Jaspersoft* in *San Francisco, CA*. Nach und nach kamen einige Funktionen, Server und Erweiterungen dazu, wie der JasperServer, der Server-basierte Berichtserstattungen beinhaltet und JasperAnalytics. 2006 wurden diese Teile zusammengefügt und als *Jaspersoft Business Intelligence Suite* veröffentlicht. Seit jeher wird diese Suite stetig weiterentwickelt und ist mit 14 Millionen Downloads weltweit und mehr als 14.000 Handelskunden in 100 Ländern eine globale

genutzte Business Intelligence Software. *Jaspersoft* stellt fünf Versionen zur Verfügung, die unterschiedliche funktionelle Angebote für Endanwender bieten: Community, Reporting, AWS (Cloud-basiert), Professional und Enterprise. Die funktionsärmste Version ist die Community-Edition, analog dazu ist die Enterprise-Version diejenige, die alle Möglichkeiten bietet.

3.4.2.3 Evaluation

Es werden einige Anforderungen für diese Arbeit an die BI-Suites definiert, die in der Übersicht (siehe Tabelle 3.1) zusammengefasst und näher erläutert werden. Das Dashboard ist die Grundvoraussetzung, die beide Hersteller bieten. Pentahos Community Edition beinhaltet eine unentgeltliche Benutzung für einen unbestimmten Zeitraum an, hingegen ist das Dashboard von Jaspersoft nicht in der kostenlosen Community Edition enthalten, sondern erst ab der AWS-Version (cloud-basiert) verfügbar. Die Versionen ab der Reporting-Version können uneingeschränkt für drei Tage auf einer gehosteten Umgebung von Jaspersoft oder als 30 Tage Test-Version auf dem eigenen System genutzt werden.

Tabelle 3.1: Vergleich BI-Suites

Eigenschaft	**Pentaho**	**Jaspersoft**
Dashboard	Ja (Community Version)	Ja (ab AWS-Verison)
Kostenfreie Nutzung	unendlich	30-Tage Testversion
Open-Source	teilweise	teilweise
Plattform	Windows, Linux, Mac OS	Windows, Linux, Mac OS, uvm.

Beide Hersteller bieten jeweils ein eigenes Wikipedia, umgangssprachlich Wiki, an, in denen alle Teile der einzelnen Bereiche der BI-Suites erläutert werden. Weiterhin sind auch eigene Youtube-Channels der Firmen verfügbar, jeweils mit Präsentationen über Möglichkeiten und Erläuterungen zur Verwendung von Programmteilen der Business Intelligence Software-Versionen. Somit wäre es möglich schnell und einfach mit der Konfiguration und Auswertung zu beginnen. Bei größeren Problemen oder Auswertungen kann auf einen kostenpflichtigen Support zurückgegriffen werden.

Die BI-Suites bieten plattformunabhängige Versionen an, die sowohl auf Windows, Linux als auch auf Mac OS X ausgeführt werden können. Pentaho stellt einen Download zur Verfügung, der nach dem entpacken auf allen Betriebssystemen ausgeführt werden kann. Als Webserver

für das Web-Dashboard wird hier ein Tomcat-Server in dem Downloadfile bereitgestellt. Somit muss die Communitiy Edition runtergeladen, ausgepackt und kann anschließend direkt gestartet und benutzt werden. Bei Jaspersoft hingegen gibt es mehrere Downloads zu den unterschiedlichen Anwendungen in der Community Version. Beide Anwendungen haben hingegen auch für jedes Betriebssystem einen Download, für Mac OS X eine dmg-Datei, für Linux sind es deb-Dateien und für Windows Installer- und Portable-Versionen, sowohl für x86 als auch x64 Architekturen. Um das Dashboard von Jaspersoft in dem 30-Tage Testzeitraum nutzen zu können, ist eine Registrierung von Nöten.

3.5 Anwendungsfalldiagramm

Für das bessere Verständnis wurden die anfallenden Aufgaben in einem Anwendungsfalldiagramm visualisiert. Die Abbildung 3.1 beschreibt das Komplettsystem. Der Akteur “Anwender” beschreibt den Mitarbeiter der die Erfassung startet. Der Akteur “Daemon” ist das Script, das auf der Maschine ununterbrochen ausgeführt wird.

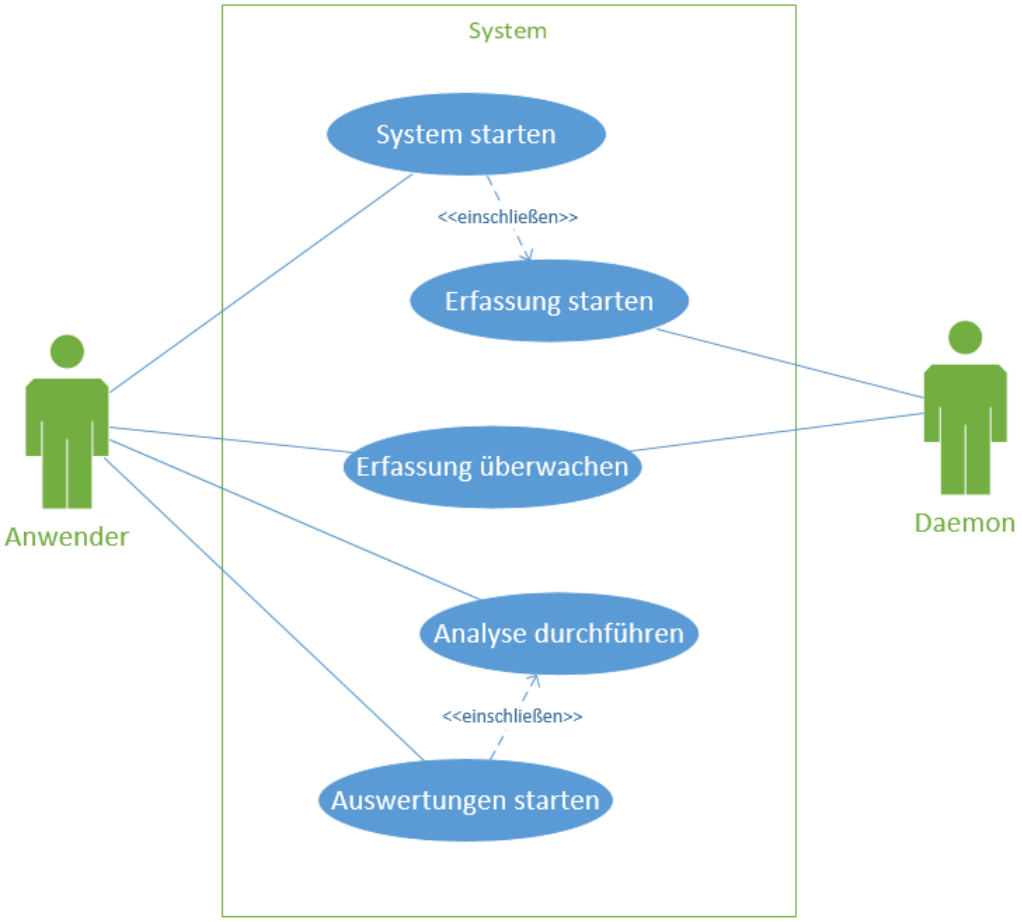


Abbildung 3.1: Anwendungsfalldiagramm

4 Konzept und Realisierung

In diesem Abschnitt wird mit Hilfe der Anforderungen ein Konzept erstellt. Es werden die Ziele für die einzelnen Teilbereiche definiert und realisiert.

4.1 Konzept

Wie in den Anforderungen kurz erläutert sollen in Zukunft möglichst schnell neue Szenarien in der bestehenden Infrastruktur umgesetzt werden können. Hierfür ist eine flexible Struktur unumgänglich. Folglich wird ein Architekturkonzept definiert und realisiert.

4.1.1 Architektur

Das Projekt wird in zwei Schichten aufgeteilt: die Infrastruktur- und die Anwendungsschicht, welche konzeptionell miteinander verbunden sind. Die Datenhaltung, Datenakquise, Säuberung und ein Großteil der Analyse wird in der Infrastruktur durchgeführt. Auf diese werden, wie auch in Abbildung 4.1 visualisiert, Anwendungen aufgesetzt. Die Anwendungen wären symbolisch die Szenarien, die durchgeführt und realisiert werden können. Somit ist es möglich beliebig viele Anwendungen zu erstellen, die auf die bestehende Infrastruktur zugreifen, um diese zu nutzen. Für diese Arbeit wurden nur die drei, in den Anforderungen definierten, Szenarien umgesetzt und implementiert, dennoch ist die Möglichkeit der schnellen Erweiterung gegeben.

4.1.2 Erhebung der Daten

Aufgrund der Vorteile einer dokumentenorientierten Datenbank zum Protokollieren der Antworten wurde MongoDB (vgl. MongoDB (2015)) als Datenhaltung der Erfassung ausgewählt. Die Hauptgründe für die Entscheidung sind die Plattformunabhängigkeit und die stetige Weiterentwicklung. Sie weist sehr gute Performance-Eigenschaften auf und dennoch muss kein Tabellenschema fest definiert werden (Stichpunkt Schemafreiheit). Dies ist evtl. wichtig, da während der Entwicklung und Erweiterung u. U. weitere Daten in der Datenbank erfasst werden sollen und es nicht nötig ist die Datenbankstruktur dahingehend anzupassen. Das Datenbanksystem kann auf Linux als auch auf Windows betrieben werden. Damit die Software

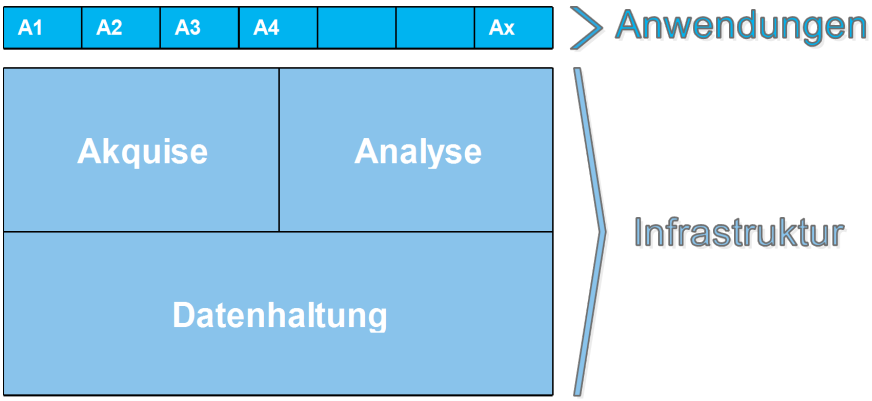


Abbildung 4.1: Architektur

auch ohne Probleme auf einer Linux ausgeführt werden kann, wird hier als Entwicklungsbetriebssystem die bekannte Linux-Distribution Ubuntu gewählt.

In der Datenbank müssen drei Datenbanktabellen, bei MongoDB werden sie Collections genannt, angelegt werden. Die erste Collection wird benötigt um den Status, die Einstellungen und den Fortschritt der Suche zu speichern. Die zweite Collection ist für die ARP-Anfragen von Nöten und die Dritte für die ARP-Antworten. Mit den ARP-Anfragen sind die Anfragen gemeint, die von anderen Netzwerkkomponenten an das Erfassungssystem gesendet werden. Mit der Speicherung dieser Daten können u.U. Sicherheitsprobleme aufgedeckt werden. Wie eben erläutert, werden in die letzte Collection alle ARP-Antworten auf die durch die Erfassungssoftware gestellten Anfragen geschrieben. Die ARP-Anfragen, die die Software generiert und versendet, werden nicht in der Datenbank gespeichert.

Die Funktionsweise der Paket-Sniffer auf dem Markt und der Fähigkeit des Versands von eigenen ARP-Anfragen, die bei diesen nicht gegeben ist, muss in diesem Fall eine Eigenentwicklung durchgeführt werden. Um die Entwicklung relativ zügig voranzutreiben wird Java 8 (JDK 8) als Programmiersprache verwendet. Aufgrund der Einfachheit der Entwicklung mit Java und der Verwendung von Eclipse wird diese Kombination gewählt. Die Hauptaufgabe in der Datenakquise ist das ständige Versenden von ARP-Anfragen, das Analysieren der Antworten in der Software und das darauffolgende Speichern in der Datenbank.

Um die Erfassungssoftware zu realisieren wurde ein Aktivitätsdiagramm (siehe Abbildung 4.2) erstellt. Nach Programmstart werden alle wichtige Komponenten geladen, Objekte aus

Klassen erstellt und Variablen initialisiert. Infolgedessen wird die Prüfung der Datenbankverbindung durchgeführt, falls der Zugriff fehl schlägt, sollte umgehend das Beenden des Programmes erfolgen. Konnte die Datenbankverbindung aufgebaut werden, wird die Netzwerkkonnektivität solange geprüft, bis sie erfolgreich zu einem Netzwerk hergestellt wurde. Im nächsten Schritt wird sowohl ein Thread zum Erfassen der ARP-Antworten erstellt, als auch einer zum Versenden der ARP-Anfragen. Die ARP-Pakete werden erstellt und versandt. In dieser Zeit läuft der Thread zur Erfassung der ARP-Antworten nebenher, um die eintreffenden Pakete zu analysieren und zu speichern. Nach einer vorher definierten Zeit wird diese Schleife wieder von vorne durchlaufen.

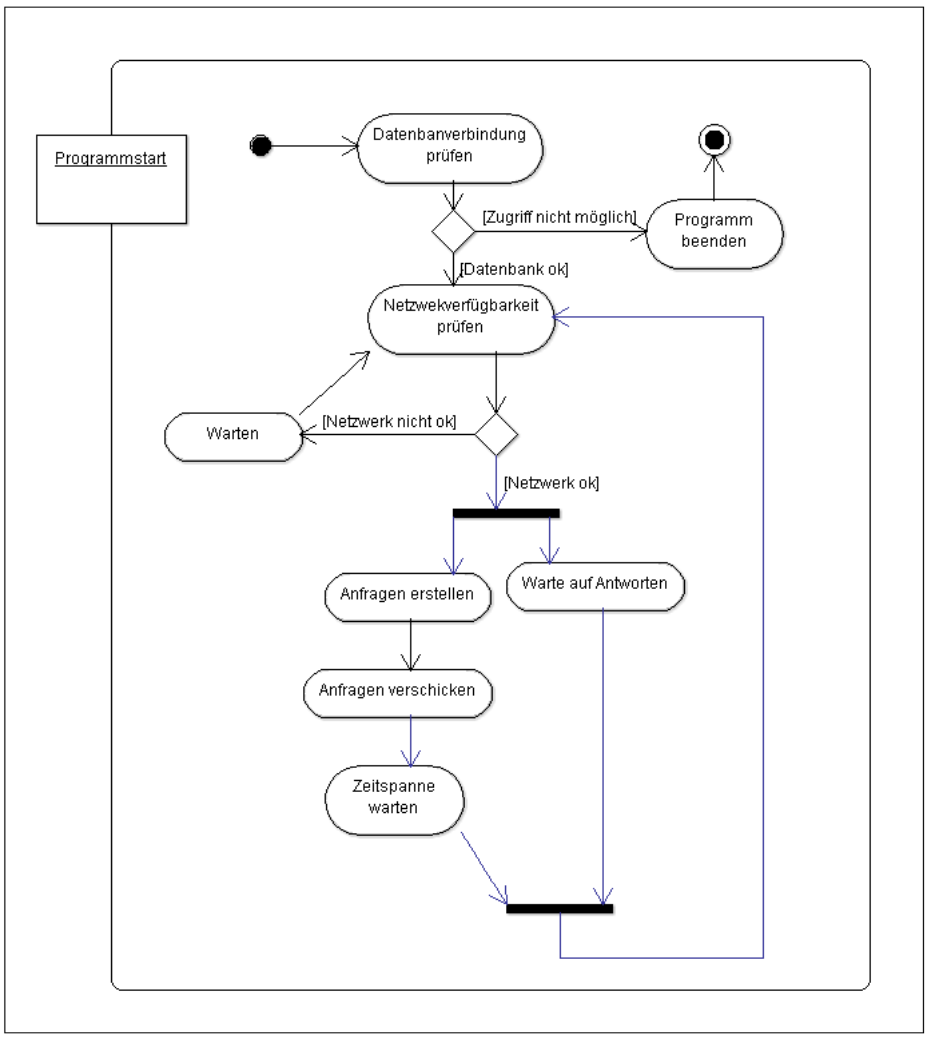


Abbildung 4.2: UML 2.0 Aktivitätsdiagramm

Als Erfassungssystem wird der Raspberry Pi verwendet, da er, wie schon vorher in der Analyse beschrieben, sehr gute Eigenschaften aufweist. Wichtige Gründe für die Entscheidung der Verwendung des Raspberry Pi ist die geringe Leistungsaufnahme gleichzeitig bietet er genügend Geschwindigkeit der Systemkomponenten, um die geforderten Aufgaben abzuarbeiten. Weiterhin kann aufgrund seiner Größe der Standort kurzfristig geändert werden. Allerdings muss bei den verwendeten Softwarekomponenten drauf geachtet werden, dass diese den ARM-

Prozessor, der im Raspberry Pi verbaut ist, unterstützen.

4.1.3 Säuberung

Um sicherzustellen, dass die erfassten ARP-Nachrichten direkt in das System fehlerfrei übernommen werden, müssen alle Arten von Zwischenspeicher abgeschaltet werden, damit bei Systemausfall keine fehlerhaften Daten in der Datenbank vorhanden sind.

Zur Säuberung für bestimmte Szenarien gehört u.a. auch, dass Geräte, die ununterbrochen ausgeführt werden eine besondere Aufmerksamkeit bekommen und auch aus bestimmten Auswertungen entfernt werden. Als Beispiel für diese Netzwerkgeräte wären hier Gateways, Access Points oder Router. Mit unter kann es auch sein, dass bestimmte Server oder Clients ununterbrochen im Netzwerk verfügbar sind, es aber nur um die Komponenten geht, die einen bestimmten Zeitraum im WLAN angemeldet sind. Das Szenario Franchise-Unternehmer, wäre ein gutes Beispiel dafür, nur WLAN-Komponenten zu registrieren, die sich nur eine kurze Zeitspanne im Unternehmen aufhalten. Für diese Art von Säuberung müssen Scripte erstellt werden, die genau dieses Szenario abdecken.

Die IEEE-OUI-MAC-Liste ist online verfügbar und kann frei heruntergeladen werden. Die Überprüfung dieser Liste ist ein Teil der Säuberung. Aufgrund der Größe der Datei muss eine Prüfung integriert werden, die die Datei auf strukturelle Richtigkeit prüft. Diese Prüfung sollte möglichst frühzeitig durchgeführt werden. Es würde sich die Prüfung auf Fehlerfreiheit in der Erfassungssoftware anbieten und sollte dort umgesetzt werden.

4.1.4 Analyse und Auswertung

Aufgrund der vorab definierten Architektur wird die Analyse und Auswertung in Form eines Berichtes in zwei Abschnitten getrennt.

4.1.4.1 Anwendungen

Als Business Intelligence Software für diese Arbeit wird hier die Community Version von Pentaho gewählt, sie bietet zwar nicht den Datenbank-Connector für MongoDB, allerdings sind die anderen Fähigkeiten genau die, die benötigt werden. Für die Community Version von Pentaho ist nur eine vorinstallierte Java-Version von Nöten und kann direkt ausgeführt werden. Somit

wird die neuste Version nur von der Webseite heruntergeladen, ausgepackt und direkt gestartet.

Das Web-Dashboard sollte, wie im Armaturenbrett im Auto, mehrere Kennzahlen auf einen Blick anzeigen. Um die unterschiedlichen Anwendungen zu realisieren sollten mehrere Dashboards erstellt werden. In den Anforderungen (siehe Abschnitt 3.4.1) wurde bereits definiert, welche Kennzahlen für die Szenarien wichtig sind. Im Nachfolgenden wird dazu definiert wie diese Daten bestmöglich angezeigt werden sollen:

Szenario KMU:

- (Tabelle) Allgemeine Kennzahlen der Datenakquise
- (Tabelle) IP-Adressen, die fest vergeben sind
- (Tabelle) IP-Adressen, die nicht vergebenen sind
- (Kreisdiagramm) Vergleich der vergebenen und nicht vergebenen IP-Adressen
- (Tabelle / Balkendiagramm) Geräte, die außerhalb der Arbeitszeit antworten

Szenario IT-Dienstleister:

- (Tabelle) Allgemeine Kennzahlen der Datenakquise
- (Tabelle / Summe) IP-Adressen, die fest vergeben sind
- (Tabelle / Summe) IP-Adressen, die nicht vergebenen sind
- (Tabelle / Summe) Geräte, die sequenziell das Netzwerk durchsuchen
- (Tabelle / Summe) Abgleich der IP-Adressen und des MAC-Hashs

Szenario Franchine-Unternehmen:

- (Tabelle) Allgemeine Kennzahlen der Datenakquise
- Abschnittszeitraum (Gesamt / pro Tag / pro Stunde)
  - (Tabelle / Summe) Anzahl der Geräte
  - (Tabelle / Summe / Torgendiagramm) Anzahl der Hersteller

  - (Tabelle / Summe / Balkendiagramm) Anzahl der Geräte pro Hersteller
- (Tabelle) Geräte, die an verschiedenen Tagen im Netzwerk angemeldet sind
- (Kreisdiagramm / Liniendiagramm) Aufenthaltsdauer der Geräte

4.1.4.2 Infrastruktur

Um die Analyse und die Auswertung zu beginnen wird eine virtuelle Maschine, nachfolgend mit VM bezeichnet, benötigt, diese muss u. U. mehrere Stunden am Tag und in der Nacht für die Auswertung lauffähig sein. Auf dieser VM sollte, wie auch auf dem Erfassungssystem, Linux verwendet werden, da Linux, wie vorab schon näher erläutert, sehr gute Eigenschaften hinsichtlich der Performance und Lauffähigkeit aufweist. Für die Entwicklung der Erfassungssoftware wurde eigens dafür eine VM erstellt, diese kann ebenfalls für die Analyse verwendet werden. Aufgrund dessen wird einiges an Zeit für die Installation und Konfiguration der Softwarekomponenten eingespart. Wie vorab definiert wird auf der VM die neuste Java- und Eclipse-Version installiert um die Erfassungssoftware zu testen und erstellen.

Damit möglichst performant mit den Grunddaten gearbeitet werden kann, sollte die Erfassungsdatenbank auch auf die VM kopiert werden. Dies hätte gleich zwei Vorteile, zum einen wäre die Performance um einiges besser, da nicht über Netzwerk auf das Erfassungssystem zugegriffen werden müsste und zum anderen würden sich die Grunddaten nicht verändern, da lokal auf der VM nur Lesezugriffe auf der Datenbank durchgeführt werden. Aus diesem Grund muss auf der VM eine MongoDB-Version installiert und konfiguriert sein, dies sollte allerdings schon in der Entwicklungsphase durchgeführt worden sein. Zum Zugriff auf die Datenbank wird die bereits vorab installierte Anwendung Robomongo verwendet.

Die Abfragen, die in den Anforderungen erläutert wurden, müssen erstellt und ausgeführt werden. Das ist wiederum die Grundlage auf dem alle weiteren Auswertungen ausgeführt werden. Zusätzlich ist es eine gute Möglichkeit um mögliche Fehler schnellstmöglich während der Entwicklungsphase zu erkennen. Bestimmte Ergebnisse können u.a. szenarienübergreifend genutzt werden. Beispiele hierfür wären die Anzahl der Geräte pro Zeiteinheit und die Anzahl der Gerätehersteller. Diese Abfragen sind für Szenarien KMU und Franchise-Unternehmer wichtig. Weiterhin sollten sie aus Gründen der Performance nicht in der Anwendungsschicht der Architektur, sondern in der Infrastruktur, durchgeführt werden. Für einige Analysen in der Infrastruktur ist festzulegen welche Auswertungen durchgeführt werden sollen. Diese Auswertungen sind Abfragen, die über die Mongo-Datenbank ausgeführt und danach zurück

in eine andere Collection geschrieben werden. Aufgrund der Einfachheit von Java, wird das Script als Java-Anwendung entwickelt. Die Software soll demnach auf die Mongo-Datenbank zugreifen, Analysen ausführen und die Ergebnisse zurück in die Datenbank schreiben.

In der Pentaho Community Edition fehlt die Möglichkeit der direkten Datenbankverbindung zu der MongoDB. Für die Verwendung von Datenbanken, die diese Version nicht beinhaltet, bietet Pentaho auch eine Lösung an. Alle Funktionen für einen etwaigen ETL-Prozess werden in der Software *Data Integration* zusammengefasst. Diese Software kann kostenfrei und auf vielen Betriebssystemen verwendet werden. Aus einer Großzahl von Datenbanken können Daten extrahiert, transformiert und in eine Zieldatenbank, CSV oder andere Art von Datenhaltung importiert werden. Als Datenbankverbindung bietet die Pentaho Community Edition für das Web-Dashboard mehrere Möglichkeiten an, u.a. MySQL, PostgreSQL, MonetDB und MS SQL Server und als Zugriff auf diese Datenbanken bietet sie die ODBC, JNDI oder die native Lösung mit JDBC an. In diesem Fall wurde sich für die Nutzung der PostgreSQL-Datenbank in Verbindung mit JDBC entschieden. Somit sollte der ETL-Prozess über die Data Integration Anwendung von Pentaho die Daten aus der MongoDB extrahieren, möglicherweise transformieren, und in die PostgreSQL-Datenbank schreiben.

4.2 Realisierung und Test

Dieses Kapitel beschreibt die Umsetzung des Konzepts. Begonnen wird mit der Erfassung der Daten, gefolgt von der Säuberung, der Analyse und der Auswertung mit Hilfe eines Web-Dashboards.

4.2.1 Erhebung der Daten

Wie im Konzept vorab definiert wurde die Datenbank mit dem Namen *wifidiscovery* realisiert. Während der Installation auf dem Raspberry Pi trat ein Problem mit dem Mongo-Datenbankmanager auf. Dieser unterstützt zum Zeitpunkt dieser Arbeit noch keine ARM-Prozessoren. Aus diesem Grund wurde eine Version benutzt, die auf dem Raspberry Pi kompiliert und von Barry Rowe, (vgl.Rowe (2015)), im Git-Hub zur Verfügung gestellt wurde.

In den folgenden Visualisierungen werden Beispieleinträge zum besseren Verständnis angezeigt (siehe Abbildung 4.3, 4.4 und 4.5).

	_id	timestamp	message
1	ObjectId("5...	2015-10-23 14:45:23.442Z	network-iface ausgewählt: wlan0 - 74:da:38:09:fb:35 - 141.22.88.176
2	ObjectId("5...	2015-10-23 14:45:37.136Z	Preparetime: 15891 ms
3	ObjectId("5...	2015-10-23 14:45:37.146Z	Sending Request between: 141.22.64.1 - 141.22.95.254
4	ObjectId("5...	2015-10-23 14:46:10.427Z	Runtime: 33281 ms
5	ObjectId("5...	2015-10-23 14:46:11.009Z	Used memory is bytes: 13535128 / >12 MBytes

Abbildung 4.3: MongoDB-Collection logs

	_id	mac	vendor	ip	hostname	destip	destma	iteration	timestamp
1	ObjectId("5...	776b6677dd598f2affe04066e0c71...	Apple,Inc.	169.254.88...	169.254.88....	169.254.88.162		1	2015-10-23 14:47:10.267Z
2	ObjectId("5...	776b6677dd598f2affe04066e0c71...	Apple,Inc.	169.254.88...	169.254.88....	169.254.88.162		1	2015-10-23 14:47:10.284Z
3	ObjectId("5...	776b6677dd598f2affe04066e0c71...	Apple,Inc.	169.254.88...	169.254.88....	169.254.88.162		1	2015-10-23 14:47:10.291Z
4	ObjectId("5...	776b6677dd598f2affe04066e0c71...	Apple,Inc.	169.254.88...	169.254.88....	169.254.88.162		1	2015-10-23 14:47:10.298Z
5	ObjectId("5...	81112fa7979173e037d680928d17...	Apple,Inc.	169.254.51...	169.254.51.26	169.254.51.26		5	2015-10-23 14:54:07.056Z
6	ObjectId("5...	a5339d4cd68effb156740e02c2594...	Apple,Inc.	169.254.13...	169.254.13...	169.254.138.239		8	2015-10-23 14:59:22.903Z
7	ObjectId("5...	30010999f7b3e8d7087a02b9018d...	IntelCorpor...	141.22.79.9	ws-79-9.ha...	141.22.88.176		9	2015-10-23 15:01:07.932Z

Abbildung 4.4: MongoDB-Collection requests

	_id	mac	vendor	ip	hostname	iteration	timestamp
1	ObjectId("5...	ebc4cede8fad53...	SamsungElectroMech...	141.22.89...	ws-8...	1	2015-10-23 14:47:06.124Z
2	ObjectId("5...	6ade11f9569f3f...	Apple,Inc.	141.22.89...	ws-8...	1	2015-10-23 14:47:06.157Z
3	ObjectId("5...	05020d1dd242cf...	SonyMobileCommuni...	141.22.89...	ws-8...	1	2015-10-23 14:47:06.182Z
4	ObjectId("5...	54c1ddd4279d2...	Apple,Inc.	141.22.89...	ws-8...	1	2015-10-23 14:47:06.199Z
5	ObjectId("5...	2aecb5d878abf1...	AzurewaveTechnolog...	141.22.89...	ws-8...	1	2015-10-23 14:47:06.217Z
6	ObjectId("5...	27de5aba5b096...	XiaomiCommunicatio...	141.22.89...	ws-8...	1	2015-10-23 14:47:06.237Z
7	ObjectId("5...	b7f9823cad36b1...	SamsungElectronicsC...	141.22.89...	ws-8...	1	2015-10-23 14:47:06.254Z

Abbildung 4.5: MongoDB-Collection responses

Die virtuelle Maschine (nachfolgend VM genannt) und der Raspberry Pi wurde gleichzeitig mit den vorher definierten Betriebssystemen installiert. Im Anschluss wurde die Entwicklungsumgebung Eclipse und Java installiert, damit schnellstmöglich mit der Entwicklung der Datenakquise begonnen werden konnte.

In dem Raspbian Jessie wurden wichtige Performance und Sicherheitsänderungen durchgeführt (in der Installationsanweisung erläutert), damit genügend Ressourcen für die Erfassungssoftware zur Verfügung stehen und keine unnötigen sicherheitsrelevante Protokolle und Dienste ausgeführt werden. Zusätzlich wurde der SSH-Zugriff und der *Watchdog* (in den Grundlagen erläutert) konfiguriert. Da die Software durchgängig laufen soll, wurde ein Autostart-Eintrag in der */etc/rc.local* eingefügt, damit sobald das Betriebssystem neu gestartet wird, nach Stromverlust oder Watchdog-Reset, auch umgehend die Erfassung fortgeführt werden kann.

Für die Java-Anwendung der Datenakquise wurde der Namen *CompDiscovery* vergeben und daraufhin mit der Realisierung auf Grundlage des vorher definierten Aktivitätsdiagramm (siehe 4.2) begonnen. Um die nötigen Bibliotheken, sowie verfügbaren Updates schnellstmöglich einzubinden und sobald ein Update verfügbar ist, dieses auch zu installieren, wurde die Java-Anwendung in Form eines Maven-Projektes umgesetzt. Zur Gewährleistung einer große Fehlerfreiheit wird nach jedem Integrieren neuer Funktionen ein neue Prototyp erstellt und auf den Raspberry Pi ausgeführt. Somit wird sowohl in der Entwicklungsphase, als auch beim Einsatz des Prototyps auf dem Zielsystem, getestet.

Um während der Erfassung der Antworten direkt bestimmen zu können um welchen Hersteller der Komponente es sich handelt, wurde die beschriebene OUI-Liste von der IEEE integriert. Dieser Schritt schloss das Prüfen auf Fehlerfreiheit mit ein und gehört zu der Datensäuberung. Die Text-Datei wurde ausgelesen und mit den ersten 24 Bits der MAC-Adresse sowie des Herstellernamens in eine Liste im Arbeitsspeicher geschrieben. Diese Prüfung wird bei jedem Programmstart durchgeführt, sollte es fehlschlagen, wird die Anwendung umgehend beendet. Nach dem Empfangen der ARP-Antwort wird der Anfangsbereich der MAC-Adresse in der Liste gesucht, um den Hersteller mit der Antwort in der Datenbank zu speichern.

Die Funktions- und Lauffähigkeitstests wurde gleichzeitig in einem kleineren Netzwerk (192.168.10.0/24) mit maximal sieben Komponenten durchgeführt und erfolgreich abgeschlossen. Aus der Anwendung und der OUI-Liste wird eine Jar-Datei erstellt, diese auf den Raspberry Pi transferiert und dort ausgeführt. Ab diesem Zeitpunkt konnten die ersten Langzeittests durchgeführt werden. Ziemlich zeitnah fiel auf, dass jüngere Geräte, wie z.B. neue Smartphones mit Android der Version 4 bis 6 oder neuere Mainboards, die interne Netzwerkadapter beinhalten, nicht jedes Mal auf eine Anfragen antworten. Erst nach Ablauf einer definierten Zeit oder nach eine hohen Anzahl von Anfragen wird auf die ARP-Broadcast-Anfragen geantwortet. Der

Grund für dieses Verhalten wird nicht weiter analysiert, da es sich höchstwahrscheinlich um einen Sicherheitsmechanismus handelt. Aufgrund dieser Tatsache mussten mehrere Anfragen nacheinander verschickt werden, damit die entsprechende Komponente antwortet. Da allerdings die Software für die Benutzung in einem großen Netzwerk mit möglichen Subnetzen ausgelegt ist, kann sie nicht durchgängig das Netzwerk mit ARP-Anfragen belasten. Weitere durchgeführte Tests waren nötig, um zu bestimmen, wann die Anfrage stets beantwortet wird. Sobald die ARP-Anfrage nach der IP-Adresse nur an die dazugehörige MAC-Adresse geschickt wird, anstatt an die Broadcast-Adresse, antwortet die Netzwerk-Komponente auch umgehend. Hierfür musste eine MAC-IP-Adresse-Zugehörigkeitsliste, in Form einer *ArrayList*, in dem Prototypen umgesetzt werden. Aus Datenschutzbedenken wird sie nur im Arbeitsspeicher gehalten und niemals fest auf nicht-flüchtigen Speicher geschrieben. Somit werden die Anfragen immer an die Broadcast-Adresse geschickt und sobald eine Antwort eintrifft wird in der Zugehörigkeitstabelle die MAC-Adresse des Senders eingetragen. Alle weiteren Anfragen werden ab dem Zeitpunkt nicht mehr an die Broadcastadresse, sondern direkt an die Sender-MAC-Adresse geschickt. Die Erfassungssoftware wird somit, solange bis eine Antwort der zugehörigen IP-Adresse kommt, alle Anfragen an die Broadcastadresse schicken. Da größtenteils in Netzwerken ein DHCP-Server die IP-Adressvergabe regelt kann es häufiger vorkommen, dass sich die IP-Adresse eines Gerätes ändert. Aus diesem Grund wird die Client-MAC-Adresse, aus der eben beschrieben MAC-IP-Adresse-Zugehörigkeitsliste, nach fünf versuchten Anfragen ohne eine Antwort zurückgesetzt und alle weiteren Anfragen an die IP-Adresse werden wieder an die Broadcast-Adresse geschickt.

In der Hinsicht, dass die Software in einem Netzwerk ausgeführt werden soll, das nicht unbedingt nur ein Class-C-Netzwerk mit maximal 254 Hosts ist, wurden die ersten Tests in dem WLAN der *Hochschule für Angewandte Wissenschaften Hamburg* durchgeführt. Nach dem erstem Start der Datenakquise war es nicht mehr möglich per SSH zu verbinden, da die Software die komplette Systemleistung in Anspruch nahm. Das Abarbeiten der Antworten war sehr langsam, sodass der Puffer der Netzwerkkarte vollgelaufen ist und es wurden Pakete, wie z.B. zum Verbinden per SSH, gedropt. Aufgrund dieser Problematik wurde die Programmierung der Java-Anwendung in einem neuen Prototypen komplett geändert, damit alle Antworten in den Arbeitsspeicher geschrieben und nach einem bestimmten Zeitraum in die Datenbank übernommen werden. Mit dieser Änderung lief die Erfassungssoftware ununterbrochen und es war eine Live-Verfolgung per SSH möglich.

Für die bessere Visualisierung des Prototyps für die Datenakquise wurde das Klassendiagramm (siehe Abbildung 4.6) erstellt. Es wird ein Objekt aus der Klasse *Startcontroller* kreiert, dies stellt die Verwaltung für die Suche und Speicherung dar. Für jeden Durchlauf der Suche wird ein Objekt der Klasse *Sniffcontroller* erstellt und folglich nach der Abarbeitung wieder freigegeben. Das Objekt vom Typ *Sniffcontroller* startet vor der Suche eine Task um alle eintreffenden ARP-Pakete im Arbeitsspeicher abzulegen.

Abbildung 4.6: CompDiscovery Klassendiagramm

4.2.2 Säuberung

Damit nicht fälschlich fehlerhafte Herstellerangaben abgespeichert werden, musste bei der Erfassung die IEEE-OUI-MAC-Liste beim Einlesen auf Fehlerfreiheit geprüft werden. Dies geschieht in der Klasse *MacConverter* (siehe Abbildung 4.6). In jeder vierten Zeile der Text-Datei stehen die ersten 24 Bits der MAC-Adresse und direkt eine Zeile darunter der Herstellername. Die MAC-Liste wird am Programmstart der Erfassung in eine Liste im Arbeitsspeicher geladen, beim Einlesen wird auf Richtigkeit der beinhalteten Werte geprüft. Falls ein Fehler beim Einlesen auftreten sollte, führt dieser direkt zum Abbruch der Software. Das hat den Hintergrund, dass durch das Hashen der MAC-Adresse nicht mehr an die herstellerspezifischen ersten drei Bytes herangekommen werden kann.

4.2.3 Analyse und Auswertung

4.2.3.1 Analyse

Für die Analyse und die Auswertung war es von Nöten eine Live-Datenbank mit aktuellen Aufzeichnung von dem Raspberry Pi zu verwenden. Hierfür wurde von dem Erfassungssystem die Mongo-Datenbank über die dazugehörige *Mongodump*-Funktion exportiert. Es wurde per SSH auf das System zugegriffen, um diese Funktion für die *Wifidiscovery*-Datenbank auszuführen. Nach Abschluss des Dumps wurden die Daten gepackt, überspielt, entpackt und per Mongo-Wiederherstellung *mongorestore* in die Mongo-Instanz auf der VM eingespielt.

Somit war die Datenbank für die Analyse lokal verfügbar. Für den Zugriff auf die Datenbank und die Collections konnte die Software *Robomongo* (vgl. Paralect (2016)), genutzt werden. Um die fehlerfreie Übernahme der Datenbank sicherzustellen wurden Abfragen in Form von Mongo-Scripten erstellt und ausgeführt. Einige dieser Abfragen sind nachfolgend aufgelistet.

```
1 db.getCollection('logs').count();
2 db.getCollection('requests').count();
3 db.getCollection('responses').count();
```

Listing 4.1: Scripte zur Anzeige der Anzahl der Dokumente in Collections

```
4 db.getCollection('responses').distinct("vendor")
```

Listing 4.2: Script zum Anzeigen alle Hersteller

```
5 db.getCollection('responses').aggregate([
```

```
6   { $group: { _id: '$mac',  anzahl: { $sum : 1} } },
7   { $sort: { anzahl: -1 } }
8 ])
```

Listing 4.3: Zeigt die Anzahl ARP-Antworten per gehaster MAC-Adresse

Für die vorab angesprochenen szenarien- bzw. anwendungsübergreifenden Auswertungen der Erfassungsdatenbank wurde ein Java-Programm in Form eines Maven-Projektes erstellt. Diese kann als Jar-Datei exportiert werden, damit keine Entwicklungsumgebung auf dem Analysesystem ausgeführt werden muss. Diese Anwendung beinhaltet drei Klassen und dient als Script für das Ausführen der Analysen.

Um keine doppelte Arbeit zu leisten wurden die Kennzahlen (siehe Abschnitt 3.4.1) für alle Szenarien zusammen in der Tabelle 4.1 betrachtet, um einen besseren Überblick zu erhalten welche Analysen szenarienübergreifend durchgeführt werden können. Sobald die Kennzahl in einem Szenario benötigt wird steht ein “X” in der Spalte, andernfalls ein “O”. Weiterhin ist zu prüfen ob mit Analysen, die eine Kennzahl zurückliefern, mehrere Antworten beantwortet werden können. Beispiel wäre dafür, welche IP-Adressen fest vergeben sind. Zum Zeitpunkt der Datenakquise wird festgelegt, in welchem Bereich gesucht wird. Diese Auswertung der IP-Adressen in Form einer Liste kann mit einem Benutzungskennzeichen erweitert werden. Dieses Kennzeichen beschreibt die Verwendungsanzahl der IP-Adresse in einem Netzwerk. Alle Datensätze oder IP-Adressen ohne Benutzung wären Adressen die nicht vergeben sind.

Tabelle 4.1: Kennzahlen für die Szenarien

Kennzahl	Szenario 1	Szenario 2	Szenario 3
Übersicht			
Antworten gesamt	X	X	X
Suchdurchläufe	X	X	X
Anzahl der erfassten Geräte	X	X	X
Anzahl der Gerätehersteller	X	X	X
Szenarien 1,2 und 3			
Anzahl der Geräte pro Zeitspanne	X	X	X
Anzahl der IP-Adressen, die fest vergeben sind	X	X	O
Anzahl der nicht vergeben IP-Adressen	X	X	O
Geräte außerhalb der Arbeitszeiten	X	O	O
Geräte, die sequenziell das Netzwerk durchsuchen	O	O	X
Geräte, die öfter/durchgehend im Netzwerk sind	O	O	X
Abgleich der IP-Adresse und MAC-Hash	X	X	X

Wie vorab erläutert werden die erstellten Analysen direkt auf der Mongo-Datenbank ausgeführt; Hierfür ist es notwendig, dass Mongo-Scripte erstellt, diese in die Java-Anwendung integriert und für die Ergebnisse neue Collections angelegt werden. Durch diese Vorgehensweise kann die spätere Webanwendung direkt auf die schon analysierten Daten zugreifen und es ist nicht nötig den großen Grundstamm an akquirierten Daten zu durchsuchen. Im nachfolgenden Abschnitt (siehe Tabelle 4.2) werden einige dieser neuen Collections näher beschrieben.

Tabelle 4.2: Erläuterung einzelner Analysen

Collection	Erläuterung der Collection
stats_overview	Auflistung der Werte aus Tabelle 4.1 - Übersicht
stats_vendors	Pro Dokument - Herstellername und Anzahl der gefundenen Geräte
stats_mac_vendor_ips	Gehashten Mac-Adressen, der Hersteller und die IP-Adressen
stats_ips_used	IP-Adressen und Anzahl der Benutzung
stats_time_components_sum_daily	Alle Erfassungstage und der Anzahl der Geräte
stats_time_comps	Pro MAC-Hash ein Dokument und ein Array mit gefundenen Stunden/Tagen
stats_time_comps_res_cnt	stats_time_comps gruppiert nach Anzahl der Stunden (Aufenthaltsdauer in Stunden)

Zwei dieser interessanteren Analysen werden im Nachfolgenden aufgelistete und erläutert.

```
db.responses.aggregate([
    {$project: {
         mac : '$mac'
        ,date: {hour: {$hour : '$timestamp'},
                day: {$dayOfMonth: '$timestamp'},
                month: {$month: '$timestamp'},
                year: {$year: '$timestamp'}}
    }},
    {$group: {_id: {"mac":'$mac'},
                zeiten: {$addToSet:{"datum":"$date" }}}},
    {$unwind: '$zeiten'},
    {$sort : { 'zeiten.datum.year' : 1,
                'zeiten.datum.month': 1,
               'zeiten.datum.day': 1,
               'zeiten.datum.hour': 1}},
    {$unwind: '$zeiten'},
    {$group: {_id: {"mac":'$_id.mac'}, zeiten: {$push:"$zeiten"}}},
    {$out:"stats_time_comps"}
],{"allowDiskUse":true})
```

Listing 4.4: MAC-Hash und stundenweises Zeiten-Array

Das in Listing 4.4 betrachtete MongoDB-Script gruppiert alle Dokumente aus der Collection Responses nach dem MAC-Hash und der Stunde, in der die Antwort eingetroffen ist. Es werden die beiden group-Bereiche verwendet, damit die Werte in einem Array sortiert gespeichert werden können. Durch diese Art von Speicherung der Daten kann die Größe des Zeiten-Arrays abgefragt werden. Wenn es nach der Größe absteigend sortiert wird, werden alle Geräte oberhalb angezeigt, die fast immer im Netzwerk verfügbar sind und geantwortet haben. Es entsteht somit eine Liste mit den Gesamtstunden im Netzwerk, in Verbindung mit der Anzahl der Geräte für diesen Zeitraum.

```
28 db.responses.aggregate([
29    { $match : { "timestamp" : {
30                 $gt : ISODate("2015-10-26T00:00:00.0Z"),
31                 $lt : ISODate("2015-10-30T23:59:59.0Z")}}},
32    { $project : { "date" : {"hour" : { "$hour" : "$timestamp"} ,
33                               "day" : { "$dayOfMonth" : "$timestamp"} ,
34                               "month" : { "$month" : "$timestamp"} ,
35                               "year" : { "$year" : "$timestamp"}} ,
36                   "vendor" : "$vendor" ,
37                   "mac" : "$mac"}} ,
38    { $group : { "_id" : { "date" : "$date" , "mac" : "$mac" ,
39                               "vendor" : "$vendor"}}} ,
40    { $group : { "_id" : { "date" : "$_id.date" ,
41                               "vendor" : "$_id.vendor"},
42                 "anzahl" : { "$sum" : 1}}},
43    { $sort : { "_id.date.year" : 1 ,
44                "_id.date.month" : 1 ,
45                "_id.date.day" : 1 ,
46                "_id.date.hour" : 1 ,
47                "anzahl" : -1}}], {"allowDiskUse" : true})
```

Listing 4.5: Stundenweise Anzahl Geräte pro Hersteller

Für die stunden- und tagesweise Anzeige der Summe an Geräten eines Hersteller wurde das Mongo-Script (siehe Listing 4.5) erstellt. Der Auswertungsbereich der Analyse beträgt eine Woche. Die Ausgabe dieses Scriptes schreibt in das Feld *_id* sowohl das Datum mit Stunde als auch den jeweiligen Hersteller. Zusätzlich wird ein weiteres Feld für die Anzahl der Geräte ausgegeben. Um eine eindeutige Anzahl an Geräten zu bekommen wurde nach der MAC-Adresse gruppiert. In der Java-Anwendung wird dieses Skript ausgeführt, dokumentenweise

durchsprungen und in der Collection *stats_time_vendors_hourly_res* gespeichert.

Damit die analysierten Daten dem Web-Dashboard zur Verfügung gestellt werden können, musste der im Konzept erwähnte ETL-Prozess durchgeführt werden. Dafür wurde die Anwendung Pentaho Data Integration genutzt. Eine neue Transformation beinhaltet alle MongoDB-Collections, die in die PostgreSQL-Datenbank übertragen werden sollen. Für jede Collection wurde ein MongoDB Input- und eine Insert / Updatekomponente angelegt und miteinander verbunden. Nachdem sowohl in beiden Komponenten die Einstellungen gesetzt wurde, konnte die Transformation fehlerfrei ausgeführt werden. Hierbei wurden alle Analysetabellen in die PostgreSQL-Datenbank integriert.

4.2.3.2 Anwendungen

Um das Web-Dashboard zu realisieren, wurde die in Abschnitt 3.4.2.1 vorgestellte Pentaho Community Edition genutzt. Nach dem Ausführen der Start-Anweisung konnte mit einem Browser auf die Web-Oberfläche zugegriffen werden. Der vorinstallierte Login-Bildschirm stellt einen Administrator bereit um schneller mit der Entwicklung beginnen zu können. Damit die Verbindung zur Datenbank realisiert werden kann, wurde unter *Datenquellen verwalten* eine neue Verbindung zu der dafür angelegten PostgreSQL-Datenbank mit Zugriff über JDBC erstellt.

Wie vorab im Konzept definiert, wurde ein Web-Dashboard pro Szenario angelegt. Aufgrund der Verwendungen von Auswertungen, die auf jedem Web-Dashboard angezeigt werden sollen, wird vorab eins angelegt. Nachdem dieses die nötigen Anzeigen und Auswertungen erhalten hat, wird durch eine interne Funktion das erstellte Web-Dashboard für die anderen beiden Szenarien kopiert.

Die Bearbeitung des Dashboards ist in drei Gruppen, hier Panel genannt, aufgeteilt. Das Layout Panel wird für die visuelle Verteilung der Komponenten verwendet, hier können auch eigene HTML-Bereiche mit HTML-Tags hinzugefügt werden. Für die schnelle Erstellung eines Layouts sind mehrere Vorlage-Templates vorgesehen und können verwendet werden. Alle Komponenten beinhalten das *Component Panel*, u.a. Text-Felder, Eingabe-Felder, Tabellen und Diagramme können anlegt und in den Einstellungen einer Spalte oder einer Zeile vom *Layout Panel* zugeordnet werden. Dieses Panel (vgl. 4.7) ist in drei Spalten aufgeteilt. Links ist eine Auflistung aller möglichen Komponenten, in der Mitte die bereits verwendeten Komponenten und rechts die Einstellungen sowie die erweiterten Einstellungen zu der, in der Mitte

ausgewählten Komponente zusehen. Um diese mit Daten zu füllen, ist es Nötig auf dem Data-Source Panel ein SQL Query anzulegen. Pentaho Community Edition bietet eine Vielzahl von Datasources an. Für den Zugriff über die Datenbankverbindung auf die Datenstruktur der Datenbank, sind unter anderem OLAP Selektoren, MDX, OLAP4J und SQL Querys vorgesehen. Viele der Einstellungen zu den Komponenten oder Querys besitzen eine Dropdown-Box oder auch Autovervollständigung, so werden beispielsweise bei der Einstellung "Datasource" einer Komponente die vorab erstellten Querys angezeigt.

Für die erste Anwendung, Szenario KMU, wurde unter Layout Structure ein Layout-Template ausgewählt. Das verwendete Layout wurde mit fortschreitender Entwicklung immer wieder angepasst, damit alle Komponenten darauf Platz finden. Zu jeder Zeit der Bearbeitung des Layouts, der Komponenten oder SQL Querys konnte über die Anzeige eines Vorschau, hier Preview genannt, der Funktionstest durchgeführt werden. Das Erstellen und Anzeigen des Preview war innerhalb weniger Sekunden abgeschlossen. Somit konnte ein stetiger Test der Anwendung durchgeführt werden. Mit fortschreitender Erfassung blieb die Performance des Previews homogen, da die SQL Querys ihre Daten cacht, um eine schnelle und performante Anzeige zu garantieren.

Um beispielsweise ein Balkendiagramm zu realisieren (siehe Abbildung 4.7) wurde auf dem *Components Panel* ein "CCC Bar Chart" hinzugefügt. Die nötigen Einstellungen für die Komponente sind der Name, die Überschrift, die Auswahl eines HTMLObjektes und einer Datasource. Das HTMLObjekt beschreibt den Punkt im Layout in dem das Diagramm angezeigt werden soll. Jede Spalte und Zeile hat eine Bezeichnung, die per Dropdown-Liste in den Einstellungen ausgewählt werden kann. Als Datenzugriff wird ein SQL Query ("sql over sqlJndi") angelegt und der Komponente in der Einstellung Datasource zugeordnet. In dem SQL Query werden die Einstellungen für den JDBC-Treiber, u.a. der Name zu der oben definierten Datenbankverbindung zur PostgreSQL-DB, gesetzt. Nachfolgend wird eine SQL-Abfrage erstellt und diese im SQL Query gespeichert. Im Anschluss daran kann das gefüllte Preview ausgeführt und getestet werden. Nach diesem Prinzip sind alle weiteren Auswertungen in Form von Tabellen und Diagrammen angelegt und gleichzeitig getestet worden. Viele Diagrammarten beinhalten weiterreichende Funktionen zum Anzeigen von zusätzlichen Informationen, die aus Platzgründen nicht unbedingt in der Grafik erscheinen sollen. Beispielsweise wird sobald sich der Mauscursor über einem Kreisdiagramm befindet, eine kleine Anzeige mit näheren Werten zu dem unterliegendem Segment angezeigt.

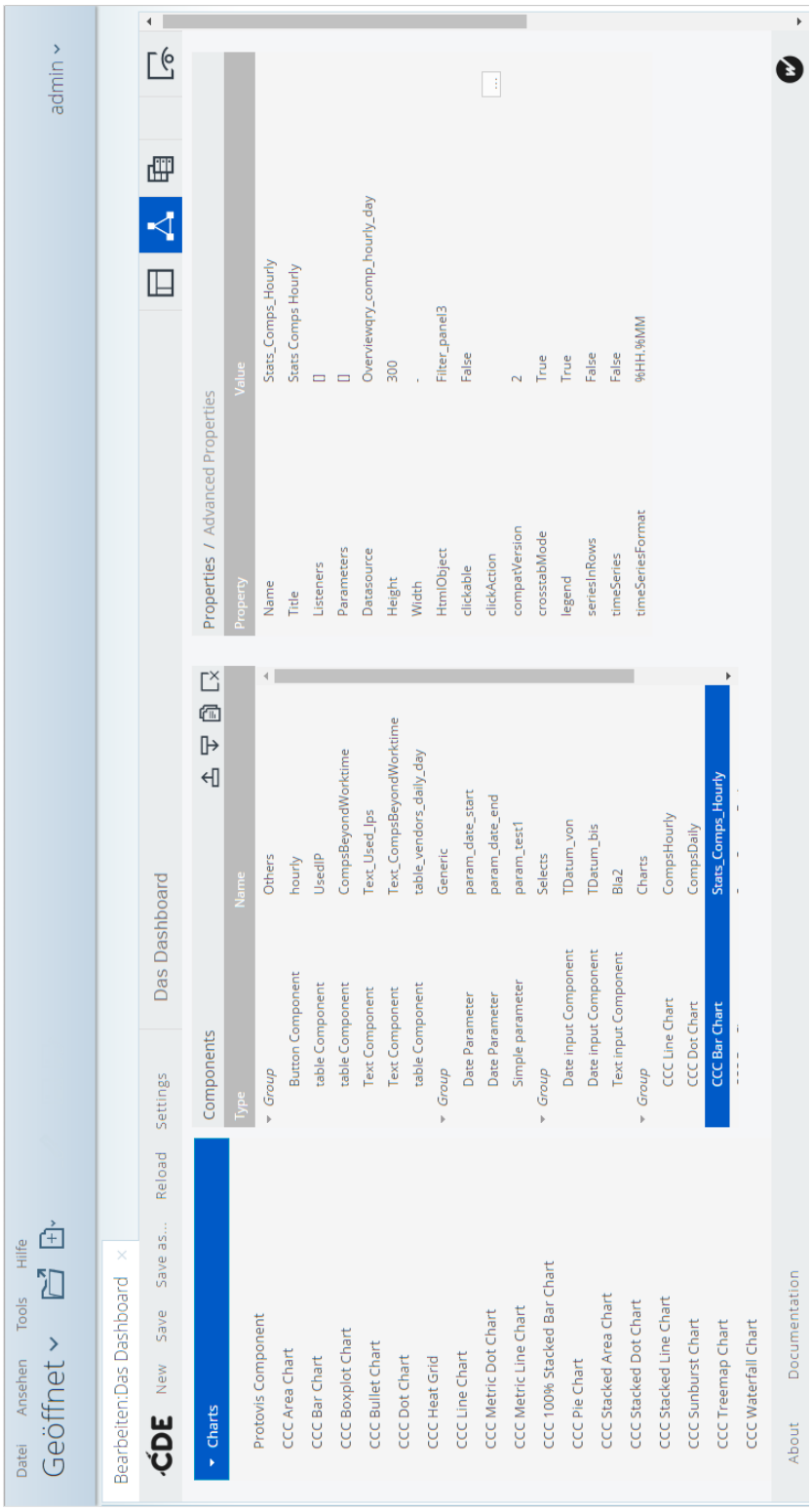


Abbildung 4.7: Dashboard - Component Panel

In dem Web-Dashboard (siehe Abbildung 5.2) für das Szenario IT-Dienstleister werden wie in allen anderen auch die Eckdaten zu der Datenakquise angezeigt. Um nicht mit vielen Anzeige-Komponenten (Text-Komponenten) arbeiten zu müssen, wurde sich für die Anzeige der Eckdaten in einer Tabelle entschieden. Die Daten wurden bereits in der Analyse der Daten erfasst und in der Tabelle *stats_overview* gespeichert. Um dem Anwender einen schnellen Überblick über die erfassten Komponenten pro Tag zu zeigen, wurde ein Balkendiagramm verwendet werden. In diesem Balkendiagramm stellen die Balken die Tage dar, die Höhe der Balken spiegelt die Anzahl der Komponenten wieder. Weiterhin war es in diesem Szenario nötig, die fest vergebenen IP-Adressen anzuzeigen. Wie in der Analyse beschrieben, beinhaltet die Tabelle *stats_ips_used* alle verwendeten IP-Adressen mit der Anzahl der Benutzung. Es war hierbei aus Sicherheitsgründen zusätzlich erforderlich das Abfragestatement auf die Benutzungsanzahl 2 abzuprüfen. Für die schnelle Sichtung, ob eine IP-Adresse möglicherweise von zwei Geräten geteilt wird. Die vierte Anzeige ist ein Kreisdiagramm, die Segmente stellen die Hersteller und die Größe stellt die Anzahl der gefundener Geräte dar. Die danebenliegende Tabelle beschreibt genauer welche Hersteller wie oft vertreten sind. Für die schnelle Suche von Herstellern wurde das Such-Feld der Tabelle aktiviert. Dieses Szenario beinhaltet auch die Prüfung auf Sicherheitsprobleme. Für die schnelle Prüfung von Sicherheitsproblemen wurde, wie oberhalb beschrieben, eine Liste mit allen Anfragen an das Erfassungssystem erstellt, diese wurde benutzt um eine Tabelle und ein Balkendiagramm auf dem Sicherheitsbereich zu füllen. Das Balkendiagramm zeigt die Summe aller Anfragen pro Tag, hingegen zeigt die Tabelle eine Liste gruppiert nach MAC-Hash mit Anzahl der Anfragen durch die Anzahl der Erfassungstage. Die nötigen Berechnung für das Durchschnittsfeld stehen in der SQL-Abfrage im SQL Query.

Die Szenarien KMU und Frenchise-Unternehmer bestehen zum großen Teil aus den Analysen und Auswertungen des Szenario IT-Dienstleister. Aus diesem Grund wurde zuerst das Web-Dashboard für den Dienstleister erstellt und dieses folglich für die anderen beiden Szenarien kopiert. Im Anschluss daran wurden einige Auswertungen für das Szenario KMU aus dem Dashboard verändert und entfernt.

Auch für das Szenario Franchise-Unternehmer wurde das Web-Dashboard, wie eben erläutert, aus dem Szenario IT-Dienstleister kopiert und weiter bearbeitet (siehe Abbildung 5.3). Der Kopiervorgang schließt sowohl das Layout, als auch die Komponenten und die Querys mit ein. Für die Auswertung pro Tag wurde zwischen die Gruppen Gesamte Auswertung und Sicherheit eine weitere Gruppe für die Tagesauswertung integriert. Um den Tag aus den Analysen herauszufiltern wurde eine Selector-Komponente verwendet, die alle aufgezeichneten

Tage anzeigt. Diese füllt auch einen definierten Parameter, der in dem Komponenten Panel hinzugefügt wurde. Alle folgenden SQL Querys beinhalten in ihren SQL-Abfragen diesen Parameter im Filterbereich. Durch eine weitere Einstellung der Komponenten wird für diesen Parameter ein sogenannter *Listener* erstellt, sobald dieser Parameter geändert wird, gibt die Komponente die Änderung an das SQL Query weiter, welches daraufhin die neue Daten lädt. Für die stundenweise Auswertung wird dieser Schritt wiederholt. Für die Tagesauswertung wurde eine weiteres Balkendiagramm erstellt, um die Anzahl der Geräte pro Stunde zu zeigen. Für die herstellerspezifische Auswertung konnte ein gestapeltes Flächendiagramm verwendet werden. Dadurch kann schnell überprüft werden, ob Geräte eines Herstellers früher oder später im Netzwerk erscheinen im Gegensatz zu anderen Anbietern. Für dieses Szenario ist es angebracht auch die Aufenthaltsdauer der einzelnen Geräte anzuzeigen. Für diese Daten sind sowohl ein Linien-, als auch Kreisdiagramm verwendet worden. In diesen Diagrammen werden die Geräte, gruppiert nach den aufgehaltenen Stunden aus dem gesamten Zeitraum der Erfassung angezeigt.

5 Fallstudie HAW.1X

Dieses Kapitel beschreibt die Benutzung der realisierten Architektur dieser Arbeit in einem WLAN-Netzwerk. Damit dieses Projekt auch für Großkunden interessant wird und mit einem großen Grundstamm an Daten arbeiten kann, wurde sich dazu entschlossen, das *HAW.1X* (das WLAN-Netzwerk der *Hochschule für angewandte Wissenschaften Hamburg*, kurz HAW Hamburg) als Erfassungsgrundlage zu verwenden.

5.1 Erfassung

Dieses Netzwerk ist in mehreren Standorten verfügbar, berücksichtigt werden hier die Campus-Standorte *Berliner Tor* und *Bergedorf*. Zwei WLAN-Benutzer in unterschiedlichen Standorten können somit untereinander über die zugewiesene WLAN-IP-Adresse kommunizieren und darüber hinaus auch Daten austauschen, wenn die Netzwerkports geöffnet sind. Damit stellt dieses WLAN eine gute Anwendung bzgl. aller durchgesprochenen Szenarien dar. Die Zielsetzung, eine Anwendung für einen unprivilegierten Benutzer zu schaffen, konnte mit Hilfe eines einfachen Studenten-Accounts im WLAN getestet werden. Es ist nicht möglich auf die Geräte zuzugreifen, wenn von außen, per VPN, ein Zugang zum internen HAW-Netz geschaffen wird. Daraus ergibt sich, dass das Netzwerk *HAW.1X* nur für WLAN-Geräte in der Hochschule genutzt wird.

5.1.1 Standort der Erfassung

Das WLAN-Netz der HAW Hamburg am Campus-Standort *Berliner Tor* ist als Erfassungsstandort prädestiniert, da auf jeder Etage mehrere Access Points zur verfügbar stehen und somit ein Höchstmaß an Verfügbarkeit des WLANs bereitgestellt wird. Somit konnte ein Standort gefunden werden, an dem kaum Benutzer einen Zugriff auf den Raspberry Pi haben. Eine ununterbrochene Lauffähigkeit und Sicherheit gegen einen Zugriff von außen wurde so sichergestellt.

5.1.2 IP-Adressbereich

Das *HAW.1X* ist ein Class-B-Netzwerk, das technisch getrennt von den Arbeitsplätzen und der Server in der HAW Hamburg verwendet wird.

IP-Bereiche	141.22.64.1 - 141.22.95.254
CIDR-Suffix	19
Subnetzmaske	255.255.224.0
Inverse Subnetzmaske	0.0.31.255
Broadcast	141.22.95.255
Standard-Gateway	141.22.80.1
DNS-Server	141.22.192.100 u. 141.22.192.101
Anzahl Hosts	8190

Es wird ein DHCP-Server eingesetzt, der die freien IP-Adressen den WLAN-Komponenten zu teilt. Durch Tests wurde festgestellt, dass die *Lease Time* abhängig von der eingesetzten WLAN-Verschlüsselung des Gerätes ist. Für den Zugriff auf den Raspberry Pi von einer anderen Komponente aus dem WLAN wird hier die Verschlüsselung gewählt, bei der die Lease Time so hoch ist, dass nach einem Neustart des Erfassungssystems die gleiche IP-Adresse wieder wird. Vorteile hat die Verwendung von nur einer IP-Adresse auch auf die Auswertungen. Die IP-Adressen zu unterschiedlichen Zeiten müssten einzeln geprüft und aus den Auswertungen herausgefiltert werden.

5.1.3 Datenakquise

Nach dem erfolgreichen Aufbau der WLAN-Verbindung am Raspberry Pi begann dieser mit der Suche. Aufgrund der Masse an Geräten im Netzwerk wurden einige Änderungen am Prototyp durchgeführt. Aufgrund der Tatsache, dass die Suche nicht zu viel Netzwerklast verursachen sollte, ist der Durchlauf der Suche auf 1:40 Minuten begrenzt worden. Da zu dieser Zeit auch noch nicht geklärt war, ob ein Sicherheitssystem im WLAN die ARP-Anfragen blockieren wird, wurde eine weitere Bremse zischen dem Versand der einzelnen Anfragen integriert. Für die Erfassung an einem Werktag ist der Zeitraum von 7 bis 21 Uhr vorgesehen worden und die Wochenenden werden nicht berücksichtigt, um das Netzwerk nicht in unnützen Zeiten in Anspruch zu nehmen.

Diese Erfassung lief einige Wochen. Jede Woche wurde mit einigen Abfragen geprüft, ob die Erfassung weiterhin funktioniert und aufzeichnet. In der dritten Wochen ist das in der

Tabelle 5.1: Zeitraum der Erfassung

Zeitraum	Log-Einträge	ARP-Anfragen	ARP-Antworten
23.10.2015 - 12.11.2015	49.741	19.184	7.973.751
20.11.2015 - 08.12.2015	45.880	20.289	7.970.225
10.12.2015 - 11.01.2016	85.543	23.051	7.976.163
Summe	**181.164**	**62.524**	**23.920.139**

Realisierung angesprochene Problem bzgl. der maximalen Datenbankgröße in Verbindung mit der 32-Bit-Version von MongoDB aufgetreten. Es wurden keine ARP-Antworten mehr gespeichert. Somit mussten die Datenbank ausgelesen, auf eine andere Maschine kopiert und danach vom Raspberry Pi gelöscht werden. Als Grunddaten für die Analyse und Auswertung wurde auf knapp 24 Millionen Datensätze (siehe Tabelle 5.1) zugegriffen. Die Tabelle beschreibt die Anzahl der Dokumente in den Collections. Wie in der Realisierung ausgiebig erläutert, stehen in den Log-Einträgen alle allgemeinen Nachrichten von der Erfassungssoftware inklusive der Fehler. Bei den ARP-Anfragen handelt es sich um Anfragen an den Raspberry Pi, falls eine Komponente eine Verbindung mit ihm aufnehmen möchte. Die ARP-Antworten beinhalten alle Antworten zu den generierten Anfragen, an alle IP-Adressen im Netzwerk.

5.2 Säuberung

Wie schon in der Analyse kurz erläutert, gibt es keine unstrukturierten Daten oder fehlerhafte Einträge in der Datenbank, da die Erfassungssoftware eine Eigenentwicklung ist. Aufgrund der Tatsache, dass die Datenbank alle 2,5 Wochen ausgelesen wurde und dies nicht immer zeitnah geschehen ist, müssen hier in der Säuberung diese Bereiche ausgegrenzt werden, damit es nicht zu Abweichungen bei den Analysen kommt. Mitunter ist tagsüber die maximale Datenbankgröße erreicht worden, deshalb ist es angebracht, diesen Tag als Abschlusstag zu markieren und ihn, bei bestimmten Analysen, nicht in die Ergebnisse einfließen zu lassen.

5.3 Auswertung

Die, in Abschnitt 4.2.3.2, entwickelten Web-Dashboards wurde mit den analysierten Daten aus der Fallstudie gefüllt. Dies wurde mit dem zuvor in der Realisierung erstellten Java-Anwendung für die Analyse der Daten durchgeführt. Aufgrund der großen Anzahl von akquirierten Daten benötigte die Java-Anwendung, die die Analyse beinhaltet, 23 Minuten. Der Schritt, die

Analysedaten per ETL-Prozess in die dafür erstellt relationale Datenbank zu übernehmen dauerte zwei Minuten. Das darauffolgende Starten des Web-Dashboards war innerhalb von fünf Sekunden abgeschlossen. In allen drei Web-Dashboards werden die Eckdaten (siehe Tabelle 5.2) zu dem analysierten Netzwerk aufgelistet. Die beiden letzten Kennzahlen beschreiben die Ergebnisse der Herstellersuche mit Hilfe der OUI-Liste von der IEEE. "Erfasste Geräte ohne Hersteller" sind Geräte, bei denen die MAC-Adresse per Treiber zu einer andere MAC-Adresse geändert worden ist oder deren OUI noch nicht von der IEEE in die Liste mit aufgenommen wurde. Hersteller können Ihren OUI-Bereich von der IEEE geheim halten lassen, dafür steht die Kennzahl "Erfasste Geräte geheimer Hersteller".

Tabelle 5.2: Eckdaten der Erfassung

Bezeichnung	**Anzahl**
Erfasste Antworten	23.920.139
Suchdurchläufe	22.096
Erfasste Tage	53
Erfassungszeitraum	Mo.-Fr. von 7:00 - 21:00 Uhr
Erfasste Geräte	25.710
Erfasste Hersteller	190
Erfasste Geräte ohne Hersteller	70
Erfasste Geräte geheimer Hersteller	65

In allen Szenarien ist die Auswertung der Anzahl der Geräte, die täglich im WLAN angemeldet sind gleich. Hier gibt es eine große Auffälligkeit, sobald Tage einer Woche mit anderen Wochen verglich werden. Von Montag bis Mittwoch steigt die Geräteanzahl stetig, wobei die Anzahl am Mittwoch am höchsten ist. Donnerstags sind ähnlich viele Geräte anwesend, wie am Montag. Allerdings gibt es einen großen Einbruch im Bereich von 27% bis 33% am Freitag zum Vortag. Das kann mehrere Gründe haben und u.a. daran liegen, dass in dem Erfassungszeitraum kaum Vorlesungen an diesem Tag durchgeführt wurden und deshalb weniger Studenten in der Hochschule anwesend waren. Auf die genauen Gründe dieser Muster wird hier in der Arbeit nicht weiter eingegangen.
In den nachfolgenden Abschnitten werden die Kennzahlen zu den Szenarien genauer beschrieben und bewertet. Dazu sind die Web-Dashboards zu den Szenarien (siehe Abbildungen 5.1, 5.2 und 5.3) geöffnet und die Auswertung interpretiert.

5.3.1 Szenario KMU

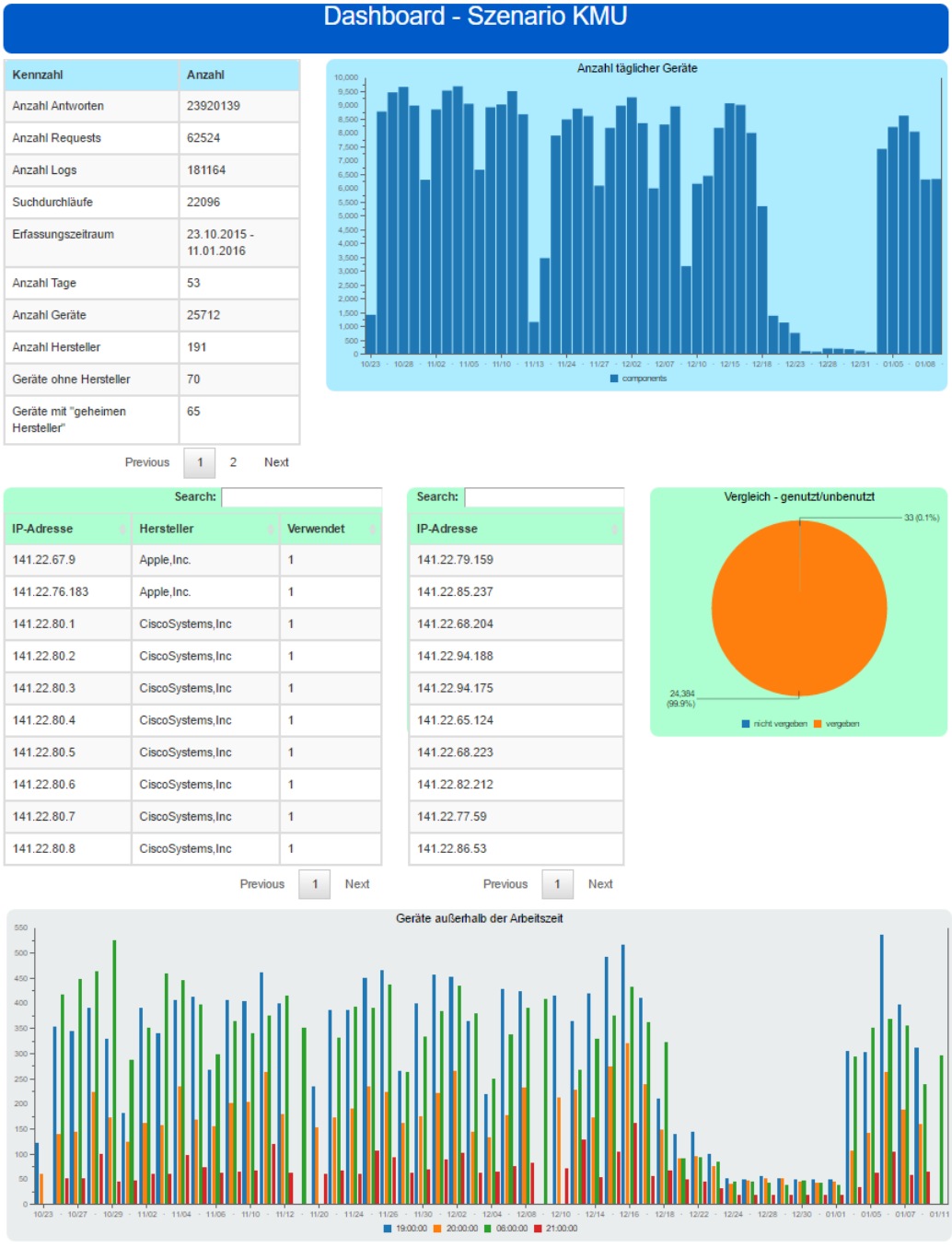


Abbildung 5.1: Dashboard - Szenario KMU

Dieses Szenario wurde für die Nutzung in einer kleinen bis mittelständischen Firma entwickelt und somit brauchen nur einige wichtige Kennzahlen angezeigt werden. Durch die Analyse der fest vergebene IP-Adressen sind zwei Auffälligkeiten beobachtet worden. Zwei Geräte des Herstellers "Apple, Inc." haben durchgängig die gleiche IP-Adresse, daraus kann geschlossen werden, dass diese Geräte Tag und Nacht laufen oder die IP-Adresse fest zugeteilt wurde, da diese auch in dem von dem DHCP-Server liegender IP-Range sind.

Elf IP-Adressen sind in dem Zeitraum nicht vergeben worden, daraus ergibt sich eine Benutzung von 99,9% aller verfügbaren IP-Adressen. Diese Adressen könnten bei dem DHCP-Server gesperrt worden sein oder sie wurden an Geräte vergeben, die nicht auf ARP-Anfragen antworten.

Der Zeitraum von 08:00 bis 18:00 Uhr ist als Arbeitszeit definiert worden. Die Anwendung für eine Auswertung von Geräten außerhalb der Arbeitszeit ist für dieses Netzwerk nicht ganz gegeben, allerdings konnten doch einige Auffälligkeiten beobachtet werden. Da die Datenakquise nur von 7:00 bis 21:00 Uhr ausgeführt wurde, konnten die Geräte, die die ganze Nacht eingeschaltet sind, nicht ausgeblendet werden. Wenn allerdings Nachts von einer Geräteanzahl von 50 angeschalteten Geräten ausgegangen wird, ist morgens, im Zeitraum 07:00 bis 08:00 Uhr, ein Anstieg von durchschnittlich 300 bis 400 Geräten zu beobachten.

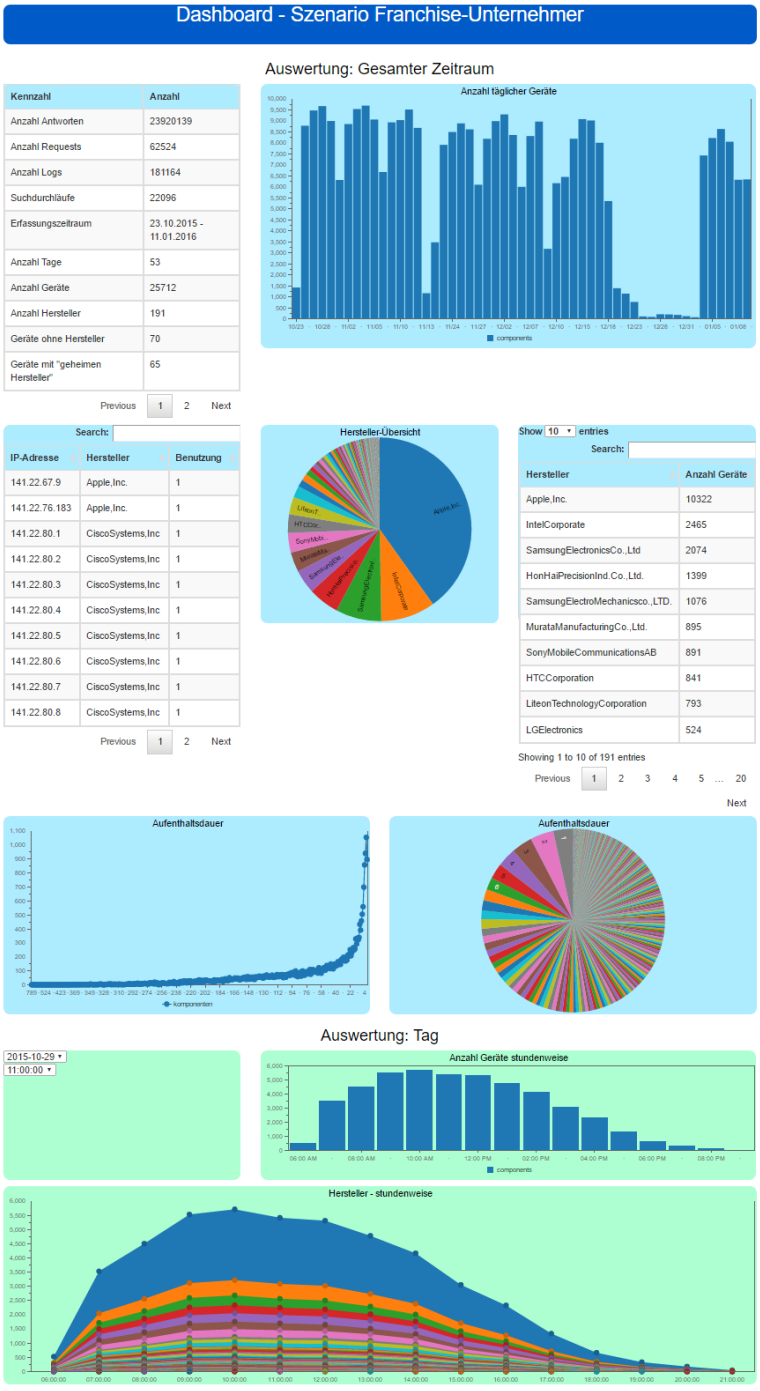
5.3.2 Szenario IT-Dienstleister

Abbildung 5.2: Dashboard - Szenario IT-Dienstleister

Für das zweite Szenario ist der Sicherheitsaspekt mit aufgenommen worden. Die Tabelle mit den fest vergeben IP-Adressen zeigt, wie in der Realisierung beschrieben, sowohl IP-Adressen an, die ein oder zwei Mal vergeben wurde (Anzeige der festen IP-Adressen). In dem IP-Bereich von 141.22.80.1 - 141.22.80.9 sind Geräte des Herstellers “CiscoSystems, Inc”, das lässt darauf schließen, dass diese Geräte z.B. Router im Netzwerk sind. Bei der IP-Adresse 141.22.80.9 wurden im Laufe der Erfassung allerdings zwei verschiedene MAC-Hashs festgestellt. Aufgrund dessen tauchte diese IP-Adresse auch nicht in der Liste der fest vergebenen Adressen im Szenario KMU auf. Grund hierfür wäre z.B. ein Defekt und der daraus resultierende Austausch eines Routers, falls dies allerdings nicht der Fall war, wurde hierdurch ein Sicherheitsproblem aufgedeckt.

Wie in der Realisierung schon näher beschrieben, wurden die zehn Geräte mit den meisten Anfragen pro Tag sortiert angezeigt. Nach gründlicher Prüfung sind keine Auffälligkeiten aufgetreten. Das Gerät mit den meisten Anfragen an das Erfassungssystem hat weniger als eine pro Tag gestellt. Daraus wird geschlossen, dass es nur standardmäßige ARP-Anfragen sind und keine, die eine Aufzeichnung in der Hinsicht dieser Arbeit darstellen. Würde das der Fall sein, wäre der Wert mindestens um das Zehnfache höher.

5.3.3 Szenario Franchise-Unternehmer



Kennzahl	Anzahl
Anzahl Antworten	23920139
Anzahl Requests	62524
Anzahl Logs	181164
Suchdurchläufe	22096
Erfassungszeitraum	23.10.2015 - 11.01.2016
Anzahl Tage	53
Anzahl Geräte	25712
Anzahl Hersteller	191
Geräte ohne Hersteller	70
Geräte mit "geheimen Hersteller"	65

Previous 1 2 Next

Search:

IP-Adresse	Hersteller	Benutzung
141.22.67.9	Apple,Inc.	1
141.22.76.183	Apple,Inc.	1
141.22.80.1	CiscoSystems,Inc	1
141.22.80.2	CiscoSystems,Inc	1
141.22.80.3	CiscoSystems,Inc	1
141.22.80.4	CiscoSystems,Inc	1
141.22.80.5	CiscoSystems,Inc	1
141.22.80.6	CiscoSystems,Inc	1
141.22.80.7	CiscoSystems,Inc	1
141.22.80.8	CiscoSystems,Inc	1

Previous 1 2 Next

Show 10 entries
Search:

Hersteller	Anzahl Geräte
Apple,Inc.	10322
IntelCorporate	2465
SamsungElectronicsCo.,Ltd	2074
HonHaiPrecisionInd.Co.,Ltd.	1399
SamsungElectroMechanicsco.,LTD.	1076
MurataManufacturingCo.,Ltd.	895
SonyMobileCommunicationsAB	891
HTCCorporation	841
LiteonTechnologyCorporation	793
LGElectronics	524

Showing 1 to 10 of 191 entries
Previous 1 2 3 4 5 ... 20 Next

Abbildung 5.3: Dashboard - Szenario Franchise-Unternehmer

Für den Franchise-Unternehmer, dem dritten Szenario, ist eine Auswertung bzgl. der Hersteller, der stündlichen Benutzung und der Aufenthaltsdauer durchgeführt worden. Von den erfassten 25000 Geräten sind über 10000 davon Geräte des Herstellers “Apple, Inc.”. “Samsung” hat unter verschiedenen Namen OUI-Bereiche der MAC-Adresse reserviert, deshalb sind mehrere “Samsung”-Hersteller in der Liste vertreten und diese kommen auf eine Gesamtanzahl von 3977 Geräten. Somit stellen Apple und Samsung mehr als die Hälfte aller Komponenten im WLAN der HAW Hamburg.

Die Aufenthaltsdauer, verdeutlicht in Tabelle 5.3, unterscheidet sich sehr massiv. Knapp 25% der Geräte sind in dem Aufzeichnungszeitraum von 52 Werktagen nur zwischen einer und neun Stunden im dem WLAN angemeldet. 1/4 der Komponenten sind somit durchschnittlich nur 4,5 Stunden im WLAN an der Hochschule angemeldet. Für die Anzahl der Geräte mit mehr als 500 Stunden kann davon ausgegangen werden, dass es Komponenten sind, die durchgängig betrieben werden.

Tabelle 5.3: Aufenthaltsdauer

Zeitraum in Stunden	**Durchschnitt**	**Summiert in Tage**
1 bis 9	25%	1
10 bis 39	25%	1 - 5
40 bis 103	25%	5 - 13
104 bis <500	25%	13 - 78

Die stundenweise Analyse der Komponenten brachte, wie vorab vermutet, zum Vorschein, dass die größte Anzahl von Geräten in der Zeit von 10:00 bis 13:00 Uhr im Netzwerk vorhanden ist. Die Abschaltung des WLANs zu den Hauptessenszeiten in der Mensa hat darauf kaum eine Auswirkung, da mehrfach in der Stunde nach Komponenten im WLAN gesucht wurde.

Bei dem Großteil der Tage die erfasst worden sind erscheinen Geräte des Herstellers “Apple’, Inc.” proportional gleichzeitig mit anderen Geräten, allerdings ist es so, dass die Personen mit diesen Geräten länger im WLAN bleiben. Dieses Ergebnis konnte aus dem gestapelten Flächendiagramm (siehe Abbildung 5.3) “Hersteller stundenweise” extrahiert werden. Um diese Kennzahl ein wenig zu verdeutlichen, wurde die Tabelle 5.4 erstellt. Der Berechnungszeitraum ist stundenweise, wobei die Ergebnisse eines Tages in drei Stundenschritten angegeben sind.

Um das Verringern der Geräte anderer Hersteller am Ende des Tage zu verdeutlichen, wurde zusätzlich die Prüfung um 20 Uhr integriert.

Tabelle 5.4: Ergebnis - Vergleich Apple-Geräte mit Geräten anderer Hersteller - 28.10.2016

Hersteller	**7 Uhr**	**10 Uhr**	**13 Uhr**	**16 Uhr**	**19 Uhr - 20 Uhr**
Apple, Inc.	42,4%	43%	42,8%	45,5%	50,1% 53,6%
Andere Hersteller	57,6%	57%	57,2%	54,5%	49,9% 46,4%
Differenz	-15,2	-14	-14,4	-9	**0,2** **7,2**

5.4 Schlussbetrachtung

Mit dieser Art von Aufzeichnung können somit eine Vielzahl an Anwendungsmöglichkeiten geschaffen und daraus sehr interessante Ergebnisse gewonnen werden. Auf einige Schlussfolgerungen wird in den folgenden Abschnitten näher eingegangen.

5.4.1 Sicherheit

In kleineren Netzwerken wird kaum ein Sicherheitssystem eingesetzt, um dieses mögliche Sicherheitsproblem der Langzeitaufzeichnung von Netzwerkgeräten aufzuzeichnen und zu unterdrücken. Vor und während der Benutzung der Software für die Datenakquise ist keine Verbindung mit dem Administrator des Netzwerkes aufgenommen worden und die Aufzeichnung lief nur mit Hilfe eines unprivilegierten Studentenzuganges. Da keinerlei WLAN-Abbrüche oder Mitteilungen über andere Wege aufgetaucht sind, muss davon ausgegangen werden, dass das Durchsuchen de Netzwerks nicht aufgefallen ist. Für den Administrator des Netzwerkes hätten die Verbindungsdaten ein Grund zur Sorgen sein müssen. Die 23 Millionen Antworten mit den dazugehörigen verschickten Anfragen an 8120 IP-Adressen, alle zwei Minuten, 15 Stunden am Tag und das 52 Tage, summiert sich auf elf Gigabyte an ARP-Paketen, die in diesem Zeitraum verschickt wurden. Dies stellt ein ganz wesentliches Sicherheitsproblem dar, jedem Benutzer des WLAN wäre es möglich, diese Art von Aufzeichnung durchzuführen. Wenn auch nicht alle ARP-Pakete gespeichert werden, stellt doch das Herauskristallisieren einiger weniger Geräte, zum Zwecke von Überwachungsmaßnahmen und im schlimmsten Fall Stalking, ein Problem dar. Dies wurde im Zeitraum der Erfassung allerdings nicht durch einen anderen Benutzer durchgeführt. Es könnte davon ausgegangen werden, dass keine intelligente Software die ARP-Pakete im Netzwerk prüft. Mit voller Sicherheit kann diese Behauptung nicht aufgestellt werden, da die Erfassungssoftware nur die ARP-Anfragen und Antworten für

die eigene MAC-Adresse speichert und nicht alle aus dem Netzwerk.

5.4.2 Kapazität

Die Netzwerkprüfung vorab ergab eine maximale Anzahl von 8190 Hosts in dem Netzwerk. Zusätzlich wurde die Zahl der Studenten, Professoren, Lehrbeauftragte und wissenschaftliche Mitarbeiter mit Hilfe von HAW Hamburg (2016) auf rund 18000 WLAN-Accounts geschätzt. Mit Überraschung wurde festgestellt, dass über 25000 Geräte in dem Aufzeichnungszeitraum gefunden worden sind. Die Geräte haben sich zu unterschiedlichsten Zeiten im WLAN angemeldet und somit reicht die Netzwerkgröße für 8190 Hosts auch ohne Probleme aus, da zu Höchstzeiten maximal 6900 Geräte mit dem WLAN verbunden sind.

5.4.3 Weitere Auswertungen

Für die mögliche späterer Nutzung können Standardauswertungen für den Anwender vorgesehen werden. Diese stehen als Grundlage für spezielle, möglicherweise nur kurzfristige, Auswertungen zur Verfügung. Möglich wären hier auch geschachtelte Analysen. Hierfür müsste die Analyse soweit verändert und erweitert werden, damit z.B. die Gruppierung in der Anwendung durchgeführt wird.

6 Abschluss

Im folgenden Kapitel werden die Ergebnisse der vorangegangenen Kapitel dieser Arbeit zusammengefasst. Einige interessante Ergebnisse der Fallstudie werden vermittelt und zusätzlich werden mögliche Erweiterungen, Veränderungen und Anmerkungen im Ausblick aufgenommen, um Verbesserungen für die Zukunft der bestehenden Architektur aufzuzeigen.

6.1 Zusammenfassung

In dieser Bachelorarbeit wurde eine Big Data Anwendung auf mit Hilfe des Big Data Life Cycle entwickelt. Es wurde eine Architektur entworfen, die konzeptionell die Infrastruktur- mit der Anwendungsschicht, den unterschiedlichen Szenarien, verbindet. Auf Grundlage des Konzeptes wurde diese Architektur realisiert. Die Realisierung wurde an einem Fallbeispiel in dem WLAN-Netz der *Hochschule für Angewandte Wissenschaften Hamburg* durchgeführt und ausgewertet.

Das definierte Ziel, eine Big Data-Anwendung zu erstellen, die für unprivilegierte Nutzer sowohl Aktivitätsmuster als auch Sicherheitsprobleme in einem potenziell fremden, WLAN-basierten Netzwerk anzeigt, wurde erreicht. Allerdings sind in einigen Teilbereichen nicht die gewünschten Teilergebnisse erzielt worden. Die verwendete BI-Suite hatte nicht die nötige Datenbankverbindung, um eine durchgängige Infrastruktur zu realisieren.

Zunächst wurden die Anforderungen aller Bestandteile der Arbeit analysiert und erläutert. Es wurden spezifische Szenarien für Anwender erstellt, die in unterschiedlichen Branchen arbeiten, um ein großes Anwendungsgebiet abzudecken. Für die Verwendung eines Web-Dashboards wurden zwei Business Intelligence Suites verglichen und evaluiert. Aus diesen Anforderungen wurde ein Konzept entwickelt. Es beinhaltet eine Architektur, in der die beiden Schichten Infrastruktur und Anwendungen konzeptionell zusammengefügt werden. Aufgrund dieses Schrittes sollte es möglich sein, viele weitere Anwendungen und Szenarien auf die bestehende Infrastruktur aufzusetzen. Die Infrastruktur steht für die Haltung der Daten, dies beinhaltet die Datenakquise, einen Großteil der Analyse und die Datenhaltung in einer MongoDB-Datenbank.

Die darüberliegende Schicht, die Anwendungsschicht, beinhaltet die Anwendungen, die auf die Infrastruktur zugreifen. Aufgrund dieser Vorgehensweise können beliebig viele Anwendung oder auch Szenarien eingesetzt werden.

Das Konzept wurde als Prototyp realisiert und in kleineren Netzwerken getestet. Um auch die realisierte Anwendung für Kunden mit größeren Netzwerken interessant zu machen, wurde ein Fallbeispiel in dem WLAN der *HAW Hamburg* durchgeführt. Im Anschluss folgte die Analyse der Daten mit Hilfe einer Eigenentwicklung. Auf dem Analysesystem wurden die analysierten Daten in Form eines Web-Dashboard zum Interpretieren bereitgestellt, dazu mussten die analysierten Daten mit einem ETL-Prozess in eine andere Datenbank kopiert werden. Auf Grundlage dieser Daten wurde das Web-Dashboard erstellt, mit dem weitere Auswertungen durchgeführt wurden. Diese Auswertungen sollten möglichst einfach und visuell gut dem spezifischen Anwender die verlangten Kennzahlen vermitteln.

6.2 Ergebnisse

Viele Erkenntnisse wurden durch die Analyse der akquirierten Daten aus dem *HAW.1X* erlangt, einige sehr interessante werden in den folgenden Abschnitten kurz erläutert.

Vorab wurde der IP-Adressbereich geprüft, wobei der Bereich eine Anzahl von 8129 Hosts zulässt. In dem definierten Erfassungszeitraum wurden 25712 Geräte gefunden, die zu 190 verschiedenen Herstellern gehören. Mit 40%, der 25712 aufgezeichneten Geräten, ist “Apple, Inc.” der Hersteller, mit den meisten Geräte im WLAN, zusammen mit den Geräten des Herstellers “Samsung” stellen sie sogar über 50% aller Komponenten.

Ein Viertel der Komponenten, die möglicherweise Smartphone, Tablets, Laptops oder weitere WLAN-fähige Geräte sind, wurden in dem aufgezeichnetem Zeitraum von 53 Tagen nur zwischen einer und neun Stunden erkannt. Daraus ergibt sich für 25% der Geräte einen durchschnittlichen WLAN-Aufenthalt von 4,5 Stunden.

Die Geräte des Herstellers “Apple, Inc.” erscheinen morgens im WLAN proportional mit den Geräten anderen Hersteller. Diese Proportionen (42 %“Apple”/ 58% andere Hersteller) bleiben über den Tag symmetrisch. Am Ende des Tages, zwischen 17:00 und 21:00 Uhr, ändert sich das Verhältnis um mehr als 10% zu einem Anteil von 54% an *Apple, Inc.*-Geräten, die länger im

WLAN angemeldet bleiben.

Anhand der erfassten Daten aus dem WLAN sind keinerlei Sicherheitsprobleme, im Bezug auf, die in dieser Arbeit benutzten Technik der Erfassung von Geräten, festgestellt worden. Allerdings wurde eine Änderung des MAC-Hashs bei einem der möglichen Router festgestellt, was möglicherweise darauf schließen lässt, dass dieser getauscht wurde.

6.3 Ausblick

Der nötige ETL-Prozess, der analysierten Daten von der MongoDB in die Postgre-Datenbank kopiert, war unausweichlich, da es leider zu dem Zeitpunkt der Arbeit noch keine direkte Verbindung zwischen dem Web-Dashboard von der Pentaho Community Edition und MongoDB gab. Für die bessere Nutzung und im Hinblick auf eine Geräteanalyse in Echtzeit, wäre eine durchgehende geschachtelte Infrastruktur, mit nur einer Datenbank als Datenhaltung, einer der wichtigsten Verbesserungen für die Zukunft. Weiterhin könnte überprüft werden, ob MongoDB bereits ARM-Prozessoren unterstützt, zum Zeitpunkt der Arbeit war dies bereits in der Planung, wurde allerdings noch nicht umgesetzt. Durch die hervorragenden Eigenschaften im Bereich der Replikation, wäre es möglich, die erfassten Datum umgehend auf das Analysesystem zu übernehmen, um sie dort live zu analysieren.

Da die Erfassung auf einen bestimmten Zeitrahmen begrenzt ist, in diesem Fall, Montags bis Freitags von 7:00 bis 21:00 Uhr, wäre es für die bessere Visualisierung in den Diagrammen, Tabellen und Grafiken wichtig, die Wochenenden und Feiertage zu definieren. Damit eindeutig und schnell zu erkennen ist, warum bestimmte Werte vom allgemeinen Durchschnitt abweichen.

In der Entwicklung des Prototypen für die Datenakquise ist aufgefallen, dass die OUI-Liste von der IEEE nicht nur erweitert, sondern auch verändert worden ist. Vorab gab es beispielsweise für die MAC-Bereiche für den Hersteller “Apple, Inc.” unterschiedliche Namen, so wie es zum Zeitpunkt der Arbeit bei “Samsung” noch der Fall ist. Die OUI-Liste sollte deshalb aus der Datenakquise entfernt und in der Analyse hinzugefügt werden, dazu müsste der OUI-Bereich der MAC-Adresse in der Datenhaltung in Klartext abgelegt werden, anstatt der Hersteller.

Um die Sicherheit und Datenschutz für die Zukunft zu gewährleisten, sollte überprüft werden, ob das Hash-Verfahren SHA3 auch immer noch sicher genug ist, damit kein Missbrauch geschieht und jemand womöglich den Hash zurück in eine MAC-Adresse rechnen kann.

Inhalt der CD-ROM

Im der nachfolgenden Aufzählung wird der Inhalt der CD-ROM aufgelistet.

- Ausarbeitung im PDF-Format
- Quellcode + realisierte Anwendungen in Jar-Format
  - Datenakquise (WifiDiscovery)
  - Analyse (CompAnalysis)
- Installationsanweisung für Raspberry Pi und Erfassungssoftware
- Skripte und Dateien
  - Script - Dump / Restore von MongoDB-Datenbank
  - Script - Prüfung auf Lauffähigkeit der Datenakquise
  - Transformations-Projekt der Pentaho Data Integration
  - SQL-Scripte für PostgreSQL
    - Datenbank-Dump von PostgreSQL
    - Löschen von Tabelleninhalten
  - Transformationsprojekt von Pentaho Data Integration
  - Dashboards - Pentaho BIServer Backup

Literaturverzeichnis

[Bundesregierung 2015] Bundesregierung: *Entwurf eines Zweiten Gesetzes zur Änderung des Telemediengesetzes*. Website http://dip21.bundestag.de/dip21/btd/18/067/1806745.pdf. 11 2015

[Edimax 2015] Edimax: *EW-7612UAn V2 Datasheet*. http://www.edimax.com/edimax/mw/cufiles/files/download/edimaxDE/transfer/products/EW-7612UAnV2/EW-7612UAn_V2_Datasheet.zip. 2015. – Accessed: 2015-09-21

[HAW Hamburg 2016] HAW Hamburg: *Daten und Fakten*. Website https://www.haw-hamburg.de/daten-und-fakten.html. Februar 2016

[IEEE 2015] IEEE: *IEEE-SA - Registration Authorityt*. Website http://standards.ieee.org/develop/regauth/index.html. 2015. – Accessed: 2015-09-21

[Jagadish u. a. 2014] Jagadish, H. V. ; Gehrke, Johannes ; Labrinidis, Alexandros ; Papakonstantinou, Yannis ; Patel, Jignesh M. ; Ramakrishnan, Raghu ; Shahabi, Cyrus: Big Data and Its Technical Challenges. In: *Commun. ACM* 57 (2014), Juli, Nr. 7, S. 86–94. – URL http://doi.acm.org/10.1145/2611567. – ISSN 0001-0782

[MongoDB 2015] MongoDB, Inc.: *Document Databases | MongoDB*. Website https://www.mongodb.com/document-databases. 2015

[Paralect 2016] Paralect, Inc.: *Robomongo - native MongoDB management tool (Admin UI)*. Website https://robomongo.org/. Februar 2016

[Plate 2015] Plate, Prof. J.: *Grundlagen Computernetze*. Website http://www.netzmafia.de/skripten/netze/netz8.html. April 2015

[Plummer 1982] Plummer, David C.: *RFC 826 - Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*. Website https://tools.ietf.org/html/rfc826. November 1982

[RASPBERRY PI FOUNDATION 2015a] RASPBERRY PI FOUNDATION: *Datasheet*. Website https://www.raspberrypi.org/products/model-b-plus/. 2015

[RASPBERRY PI FOUNDATION 2015b] RASPBERRY PI FOUNDATION: *Raspberry Pi Hardware*. Website https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md. 2015

[Rowe 2015] Rowe, Barry: *MongoDB for ARM*. Website https://github.com/Barryrowe/mongo-arm. Januar 2015

[Scofield 2010] Scofield, B.: *NoSQL @ CodeMash 2010*. Website http://de.slideshare.net/bscofield/nosql-codemash-2010/. Januar 2010. – URL http://de.slideshare.net/bscofield/nosql-codemash-2010/

[Yamada 2015] Yamada, Kaito: *kaitoy (Kaito Yamada) · GitHub*. Website https://github.com/kaitoy. 11 2015

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 29. April 2016 Simon Kallweit