Bachelorarbeit

Studiengang Informatik

Hochschule Luzern

Coaxial Tree Map

Adhurim Krasniqi

Breuer: Markus Waldmann
Experte: Konrad Durrer
Bearbeitungsreitraum: 23.09.2022 – 03.01.2023

Bachelorarbeit an der Hochschule Luzern - Informatik

Titel: Coaxial Tree Map

Studentin/Student: Adhurim Krasniqi

Studiengang: BSc Informatik.

Abschlussjahr: 2023

Betreuungsperson: Markus Waldmann

Expertin/Experte: Konrad Durrer

Auftraggeberin/Auftraggeber: Evren Yilmaz

Codierung / Klassifizierung der Arbeit:
☒ Öffentlich (Normalfall)
☐ Vertraulich

Eidesstattliche Erklärung
Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt habe, alle verwendeten Quellen, Literatur und andere Hilfsmittel angegeben habe, wörtlich oder inhaltlich entnommene Stellen als solche kenntlich gemacht habe das Vertraulichkeitsinteresse des Auftraggebers wahren und die Urheberrechtsbestimmungen der
Hochschule Luzern respektieren werde.

Ort/Datum, Unterschrift _____

Abgabe der Arbeit auf der Portfolio Datenbank:
Bestätigungsvisum Studentin/Student
Ich bestätige, dass ich die Bachelorarbeit korrekt gemäss Merkblatt auf der Portfolio Datenbank ablege. Die Verantwortlichkeit sowie die Berechtigungen gebe ich ab, so dass ich keine Änderungen mehr vornehmen oder weitere Dateien hochladen kann.

Ort/Datum, Unterschrift _____

Verdankung
Hiermit bedanke ich mich herzlich bei meinem Betreuer für die Beratung, meinem geschätzten Arbeitskollegen für die spannenden Reviews und meinen Bekannten für die Unterstützung.

Geistiges Eigentum gemäss der Studienordnung für die Ausbildung an der Hochschule Luzern, FH Zentralschweiz

Management Summary

Das Ziel der nachfolgenden Bachelorarbeit war es, eine Komponente zu erstellen, welche das HF-Netzes in einer Grafik abbildet. Durch die Aggregation und visuellen Bereitstellung dieser Daten, würden Zusammenhänge im HF-Netz sichtbar. Diese Übersicht könnte den Technischen Angestellten helfen, Rückschlüsse auf Fehlerhafte Installationen zu tätigen.

Da es sich bei der Komponente um eine Erweiterung handelt, wurden bestehende Komponente durch Software-Patterns, wie dem Adapter oder der Fassade, kombiniert.

So würden die Störungen bei den Kunden verkürzt werden, die Fehler im HF-Netz früher gefunden und eine proaktive Lösungsfindungen, sind zukünftig möglich.

The goal of the following bachelor thesis was to create a component that depicts the RF network in a graphic. Through the aggregation and visual provision of this data, correlations in the RF network would become visible. This overview could help the technical staff to draw conclusions about faulty installations.

Since the component is an extension, existing components were combined by software patterns, such as the adapter or the facade.

This would shorten the disruptions at the customers, the faults in the RF network would be found earlier and a proactive solution finding, are possible in the future.

Inhaltsverzeichnis

Management Summary **1**

1 Einleitung **7**

1.1 Problem *7*

1.2 Ausgangslage *8*

1.3 Fragestellung *9*

1.4 Abgrenzung *10*

1.5 Aussicht *11*

2 Stand der Technik **13**

2.1 Kunden *13*

2.2 Monitoring Tools *14*

2.3 Homeoffice *17*

2.4 Visualisierung *18*

2.5 Messwerte *19*

2.6 Software *19*

2.7 Tests *20*

2.8 Projektlücke *20*

3 Ideen und Konzepte **21**

3.1 Roter Faden *21*

3.2 Zusammenhang Netz *21*

3.3 Netzwerk Störungen Koaxial *21*

3.4 Darstellung Baum *22*

4 Methoden **23**

4.1 Projektmanagement *23*

4.2 Nutzwertanalyse *25*

4.3 Bedarfspegel und Einheitspegel *26*

4.4 Tools *26*

4.5 Risikoanalyse *26*

4.6 Softwareentwicklung *28*

4.7 Tests *28*

5 Realisierung **29**

5.1 Gantt Diagramm *29*

5.2 Arbeitsrapport *30*

5.3 Mögliche Variante *32*

5.4 Mögliche Visualisierung *35*

5.5 Analyse IST-Zustand *36*

5.6 Systemkontextdiagramm *36*

5.7 Prozesse und Szenarien *37*
5.8 Messwerte ermitteln *38*
5.9 Pegelkonzept *38*
5.10 Komponentendiagramm backend *39*
5.11 Sequenzdiagramme backend *43*
5.12 Parallelisierung backend *46*
5.13 Einführung Frontend *46*
5.14 Bestandesaufnahme Frontend *46*
5.15 Implementation Frontend *47*
5.16 Integrationstest – GUI *48*
5.17 Funktionstest – GUI *49*
5.18 Resultate *52*

6 Evaluation und Validation **54**
6.1 Erreichen und Erfüllen der Anforderungen *54*
6.2 Reflexion *56*

7 Ausblick **57**

8 Anhänge **58**
8.1 Anforderungskatalog *58*
8.2 Testplan *59*
8.3 Bestandesaufnahme der Techniker *70*

9 Abkürzungs-, Abbildungs-, Tabellen-, Quellen-Verzeichnis **73**
9.1 Abkürzungsverzeichnis *73*
9.2 Abbildungsverzeichnis *73*
9.3 Tabellenverzeichnis *74*
9.4 Literaturverzeichnis *74*

1 Einleitung

Dieses Kapitel gibt die Probleme und die Momentane der Technischen Angestellten, eines Internet Service Providers wieder. Es folgt eine Fragestellung über eine Lösung, welches unter Berücksichtigung der bestehenden Systeme und deren Grenzen, untersucht wird.
Der Mehrwert, der durch das in diesem Projekt realisierten Produktes, wird nach der Durchführung gemessen.

1.1 Problem

Der Telekomanbieter bietet verschiedene Services an. Dazu zählen verschiedene Fernseh-Abos, Internet und Festnetztelefonie. Damit die Services genutzt werden können, werden dies über Zuleitungskabel, siehe Abbildung 1 HF-Netz, übermittelt. Die Zuleitungskabel bestehen aus den beiden Technologien Koaxialkabel oder Glasfaser.
Die Kabel werden, je näher sie zum Telekomanbieter rücken, immer mehr zusammengeführt. Das kann einerseits durch passive Elemente wie zum Beispiel Verteiler, oder andererseits durch aktive Netzkomponente, wie zum Beispiel Verstärker, erfolgen. Falls eines dieser Netzkomponenten ausfällt, kommt es bei vielen Kunden zu Unterbrüchen. Je näher sich die Netzkomponente nämlich beim Telekomanbieter befindet, desto mehr Kunden sind aufgrund der Aggregation der Leitungen betroffen. Die Kunden melden sich in so einem Fall beim Telekomanbieter über verschiedene Kanäle. Das kann über Telefon, über Mail oder kann sogar physisch über einen Besuch im Verkaufsraum erfolgen. Diesen ganzen Meldungen generieren eine Unmenge an Aufwand, die daraus besteht, dass sich die technischen Mitarbeiter jeden Kunden einzeln kontaktieren müssen, um Sie fachlich zu beraten und zu unterstütze. Das kann per Telefon, Fernwartung oder vor Ort erfolgen.
Viele dieser ausgefallenen Komponenten treten jedoch nicht einfach plötzlich aus, sondern fangen früh an, über Ihren schlechten Zustand sich bemerkbar zu machen.

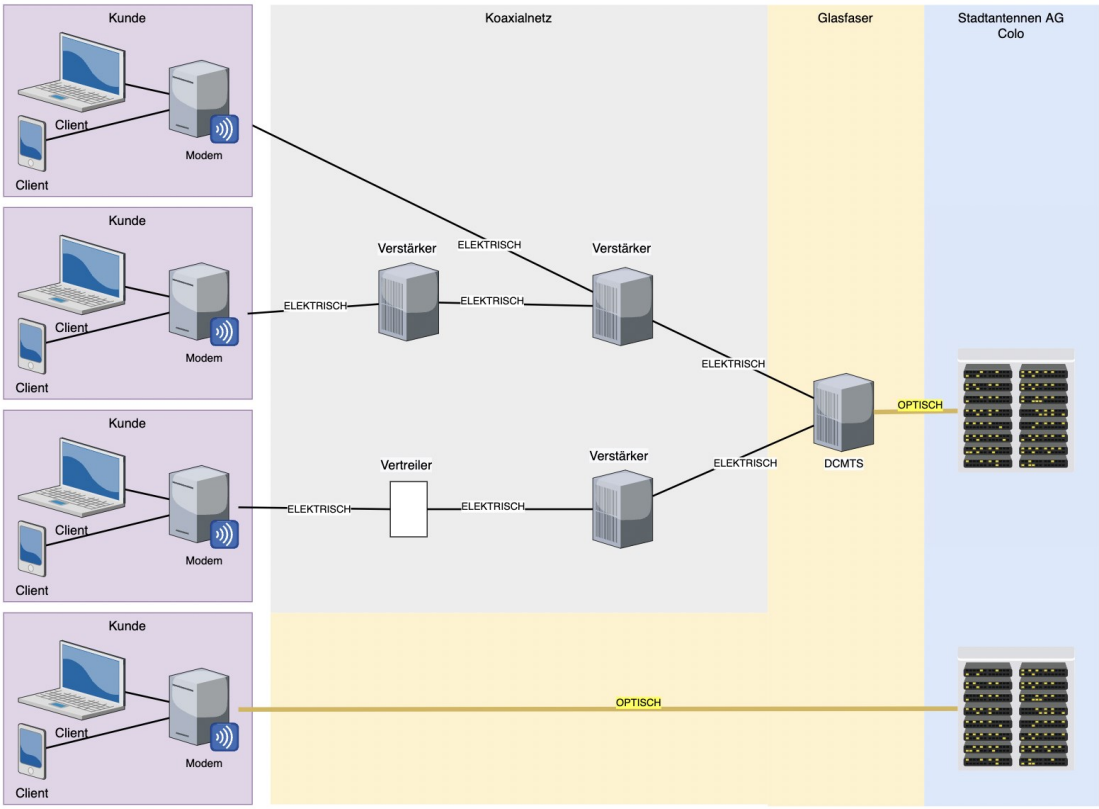


Abbildung 1 HF-Netz

Quelle: Eigene Darstellung

1.2 Ausgangslage

Um den zuvor im Kapitel 1.1 angesprochenen Support der Kunden zu gewährleisten, nutzen die technischen Mitarbeitern das Monitoring-Tool «Watchdog».
Beim Watchdog handelt es sich um eine eigens entwickelte Applikation, die es ermöglicht, Informationen bezüglich dem HF-Netz auszulesen. Dadurch das der «Watchdog» historisch gesehen sehr viele Daten aus verschiedensten Quellen aggregiert, hat er monolithische Züge angenommen.
Heisst, Methoden oder Klassen halten sich nicht an die Clean Code vorgaben. Eines davon ist das «Single-Responsibility-Prinzip», wo die eindeutige Verantwortlichkeit einer Klasse vorschreibt.

In der Abbildung 2 sind die Komponenten vereinfacht dargestellt. Links den einzelnen Layer zusammengefasst. Zuunterst liegt der Layer mit dem Koaxialnetz – das ganze HF-Netz. Dazu zählen Verstärker, Verteiler und Kabelmodems welche in Abbildung 1 abgebildet sind. Die wichtigsten Daten aus diesem Layer wurden im darüberliegenden, dem sogenannten Daten-Layer, in den dafür vorgesehenen Datenbanken aufgenommen. Je nach Art der Information werden Daten für 30 Tage, oder ohne zeitliche Beschränkung persistent abgespeichert. Die 30 Tage dienen dazu, um bei Störungen historisch die Daten auszulesen. Im Aggregations-Layer werden die Daten anschliessend aggregiert. Je nach Bedarf dieser Informationen, besteht die Möglichkeit, dass dieser Layer über mehrere Datenbanken hinweg sich Informationen beschafft. Dieser Layer ist auch für die Serialisierung der Informationen zuständig. Nach der Aggregation der Daten, werden sie weiter an das Frontend geleitet. Das Frontend befindet sich im Layer Ansicht & Aggregation, wo die Daten anschliessend visuell abgebildet werden. Das sind je nach Informationen Tabellen, Werte oder Graphen, die dem Techniker, der sich im Layer darüber befindet, die Informationen sinnvoll bereitstellt.

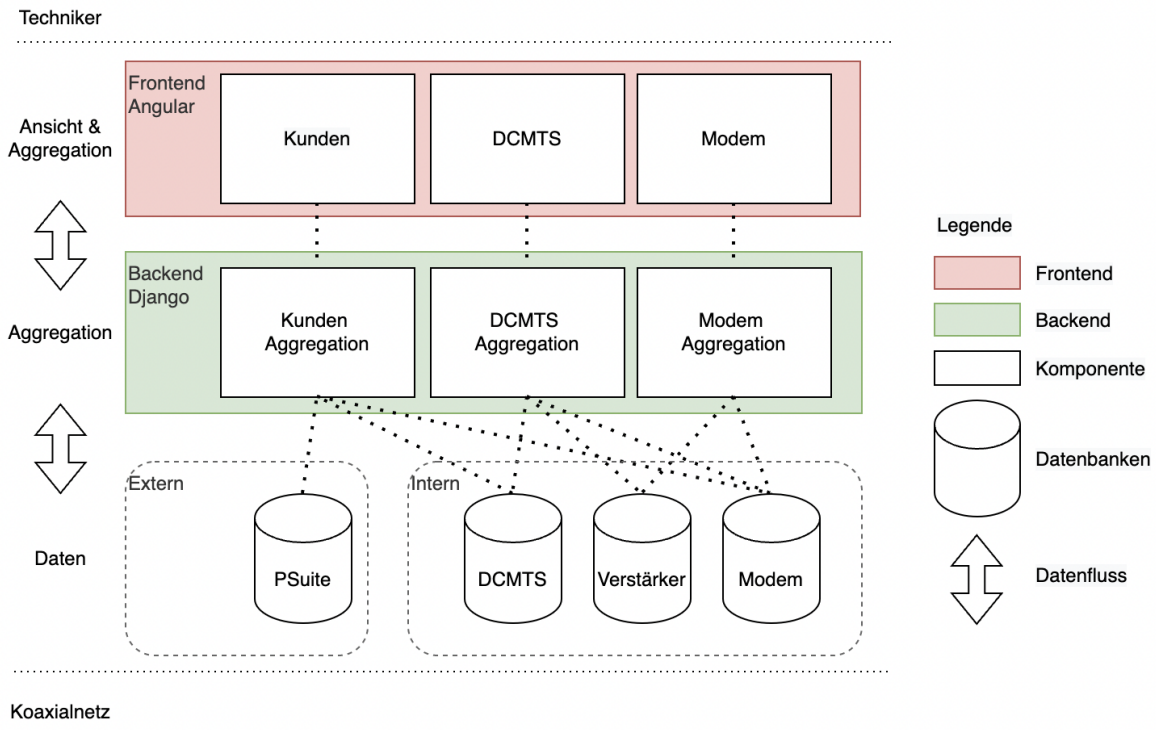


Abbildung 2 Systemkomponente Ausgangslage

Quelle: Eigene Darstellung

1.3 Fragestellung

Beim Coaxial Tree Map Projekt geht es um eine Komponente, die im Rahmen dieser Bachelorarbeit erstellt wird. Diese wird sich voraussichtlich über mehrere Layer, die im Kapitel 1.2 vorgestellt wurden, erstrecken. In Abbildung 3 ist diese Komponente rot umrandet. Für die neu erstellte Komponente werden verschiedene Metriken definiert und voraussichtlich in einem Eignen dafür vorgesehenen Datenbank, abgespeichert.
Diese werden anschliessend von der im Aggregations-Layer liegenden CTM-Aggregation-Komponente, CTM-View, aggregiert.

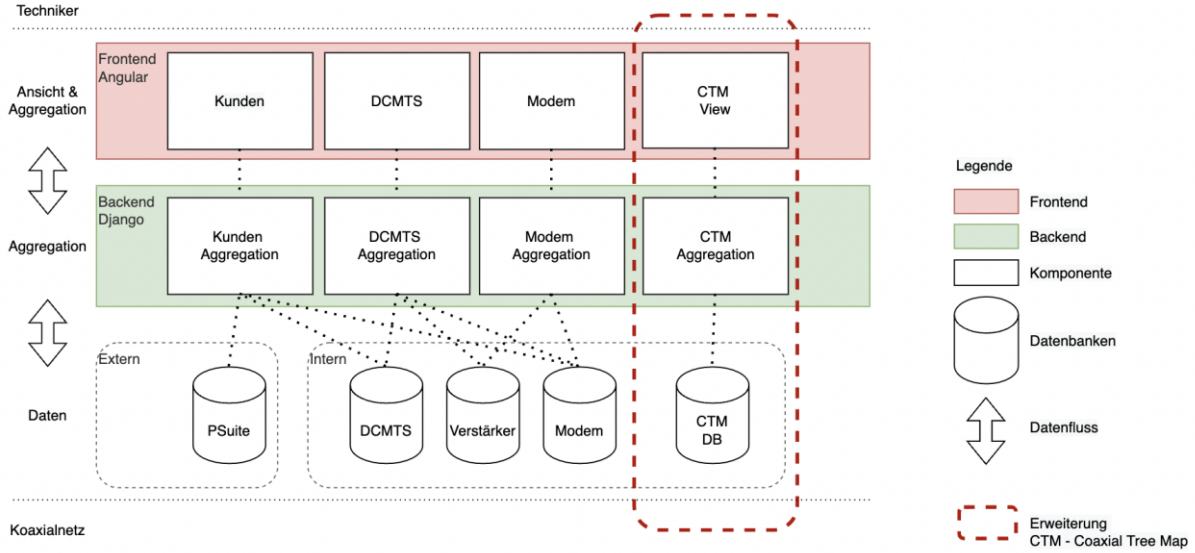


Abbildung 3 Systemkomponente CTM vereinfacht

Quelle: Eigene Darstellung

Es müssen noch weitere Punkte geklärt werden, damit sich die Anforderungen des Arbeitgebers erfüllen lassen:

- Braucht es eine DB oder kann mit Hilfe der bestehenden Datenbanken auf die nötigen Informationen zugegriffen werden?
- Was für eine Technologie ist bei der DB zu wählen? Da sehr viele Daten anfallen werden, muss diese einerseits performant sein und andererseits, darf sie nicht zu viel Speicherplatz verschwenden.
- Wie lange und in welchem Format sollen die Daten gespeichert werden? Welche Daten sollen gespeichert werden?
- Da auf viele Schnittstellen zugegriffen wird, müssen die Berechtigungen definiert und bewilligt werden

Folgende Bedingungen wurden vom Arbeitgeber beschlossen:

- Die Komponente für das Frontend wird mit dem bestehenden Framework Angular realisiert. Die bestehende Version muss beachtet werden, da das Frontend nicht auf die aktuelle Version updatend wurde.
- Das Backend wird mit Django realisiert.
- Für die Visualisierung steht d3.js zur Verfügung.
- Für das Projekt wird ein separater GitHub-Server erstellt.
- Das Frontend wird separat auf einem Server laufen.
- Das Backend wird separat auf einem Server laufen.

1.4 Abgrenzung

Folgende Bereiche sind nicht Teil dieser Arbeit - folgende Vorgaben werden vom Telekommunikationsanbieter definiert, welche die Grenzen dieses Projektes bilden:

- Das Frontend wird später durch den Arbeitgeber ausgebessert bzw. noch erweitert.
- Die Implementation in das bestehende Monitoring-tool wird in einer zweiten Etappe durchgeführt, was nicht Teil dieser Arbeit ist.
- Benötigte Komponenten werden von anderen Teams entwickelt und werden hier nicht weiter erörtert.

1.5 Aussicht

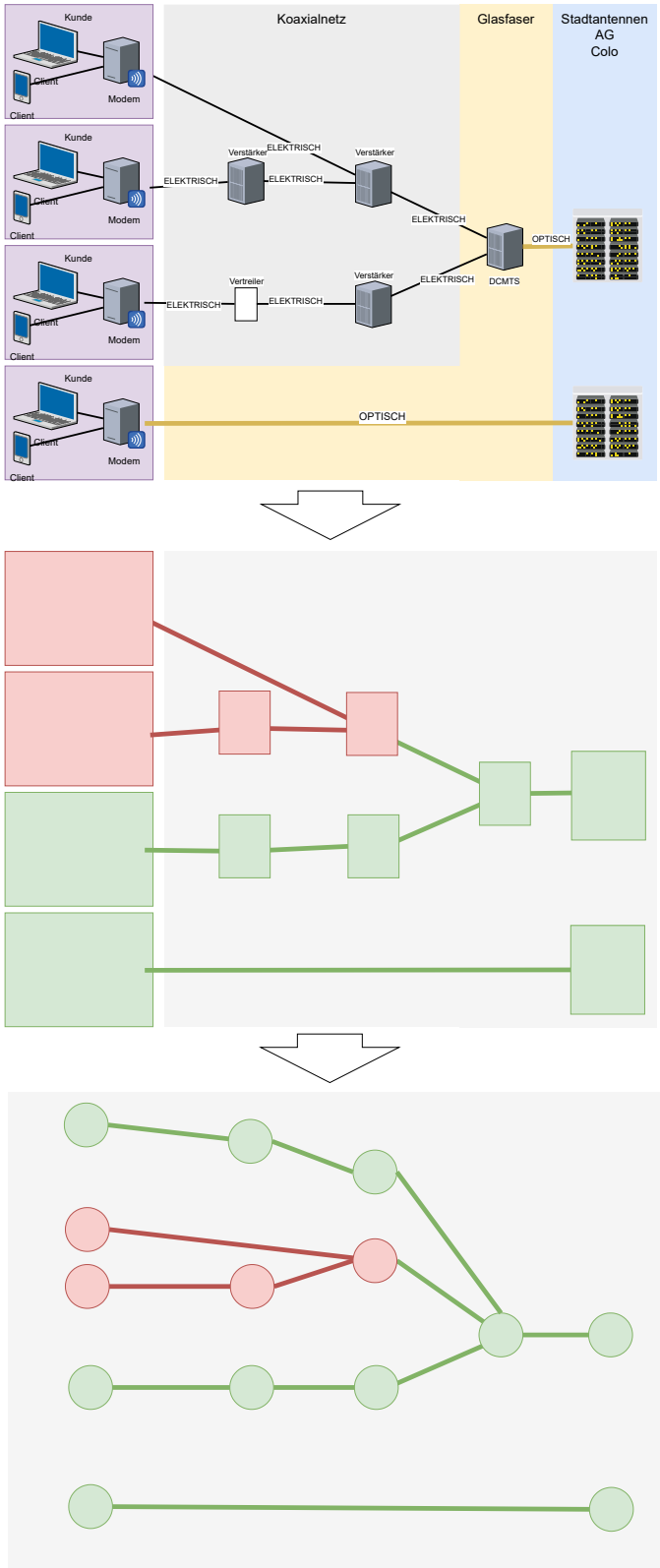


Abbildung 4 Abstraktion der Koaxial-Komponente

Quelle: Eigene Darstellung

Wie in Abbildung 3 deutlich zu erkennen ist, gilt es, eine Komponente zu entwickeln, welche das bestehende Monitoring-Tool Watchdog erweitert.
Es gilt, die Abstraktion der einzelnen Netzwerk-Komponente zu erreichen. Unter Abbildung 4 wird die Abstraktion aufgezeigt, die das HF-Netz durch das Projekt treffen wird. Die einzelnen Komponenten, die zusammenhängen, werden als Knoten, auch Nodes genannt, abgebildet. Falls sich der Zustand dieser Knoten im Wertebereich liegt, wird der Knoten Grün dargestellt. Falls der Zustand sich jedoch ausserhalb dieses Wertebereichs befindet, erscheint der Knoten Rot. Diese Abstraktion wird den Technischen Mitarbeitern dazu dienen, den Fokus auf das Wesentliche zu richten, um dadurch die problematischen Stellen im HF-Netz früher zu ermitteln. Ausserdem würde durch die Zeitersparnis, die durch die Erstellung dieser Visualisierung entsteht, mehr Zeit zur Verfügung stehen, da das manuelle Erfassen aktuell viel Zeit in Anspruch nimmt. Die ganze Kollektion, Aggregation und Interpretation dieser Werte wird später das erstellte Projekt übernehmen und diese anschliessend visuell zur Verfügung stellen.
Unter Abbildung 5 wird beispielhaft eine solche Störungsstelle veranschaulicht.
Durch die Komponente wäre die Störung bereits eingegrenzt, sodass der Fokus auf die roten Knoten gesetzt werden kann. Diese sind in Abbildung 5 mit einer gestrichelten, roten Linie illustrativ umrandet. Diese Linie ist jedoch nicht Teil der Arbeit.

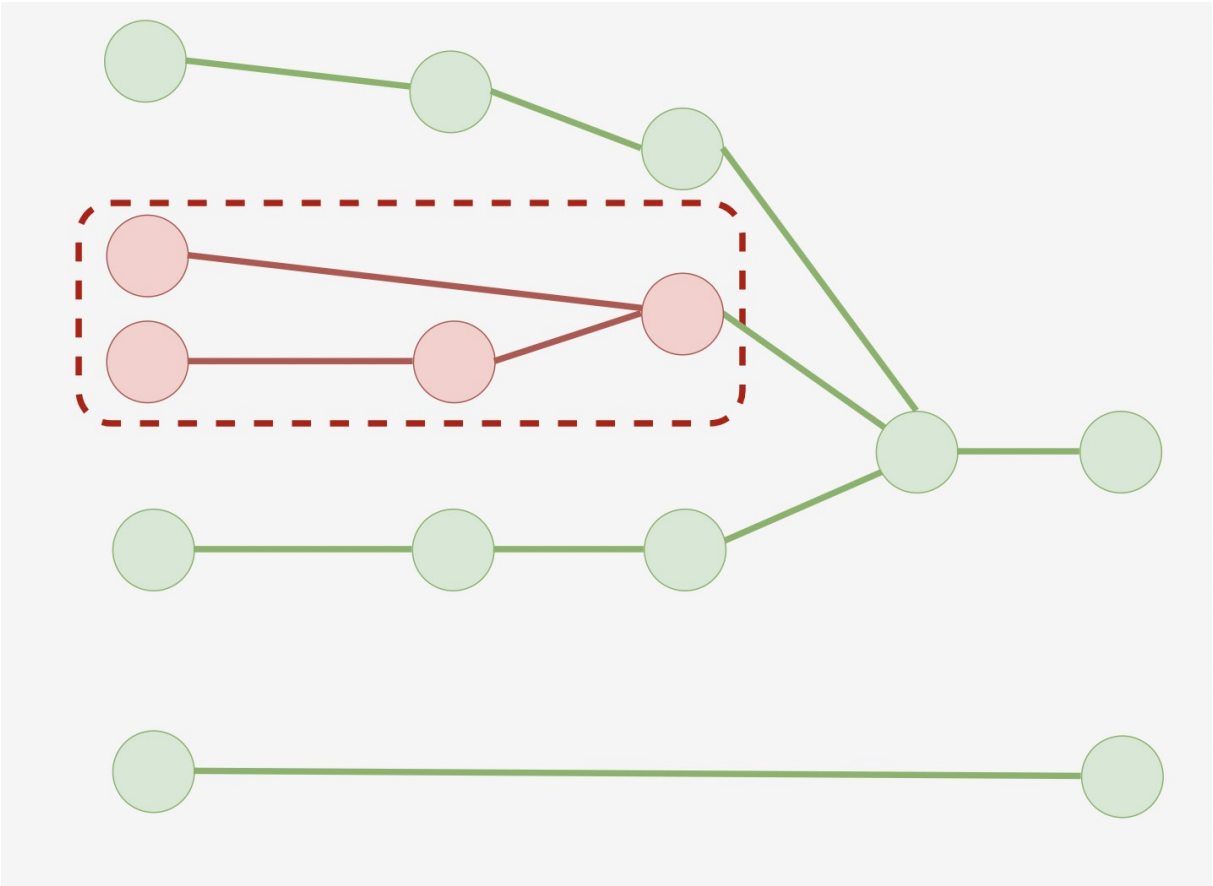


Abbildung 5 Störungsstelle

Quelle: eigene Darstellung

Ein positiver Effekt wäre ausserdem das proaktive Arbeiten, was den Mitarbeitern somit ermöglichen würde, Probleme im Netz zu behandeln, bevor diese zu Ausfällen führen würden.

2 Stand der Technik

Den Stand der Technik wird in diesem Kapitel erläutert.

2.1 Kunden

Hier wird aufgezeigt, wie wichtig die rasche Wiederherstellung der Services ist. Weiter werden Aspekte angesprochen, welche einen loyalen und treuen Kunden ausmachen.

Bei Ausfällen im Koaxialnetz sind die Kunden als erstes betroffen, da der Service dadurch nicht mehr gewährleistet werden kann. Diese Ausfälle sind jedoch laut der Studie von (Buttle & Burton, 2001) weniger schädlich als der Wiederherstellungsprozess selbst.
Ausserdem werden mehrere Faktoren genannt, die einen hohen Einfluss auf die Loyalität des Kunden ausüben. Diese sind aus der Grafik Abbildung 6 ersichtlich.

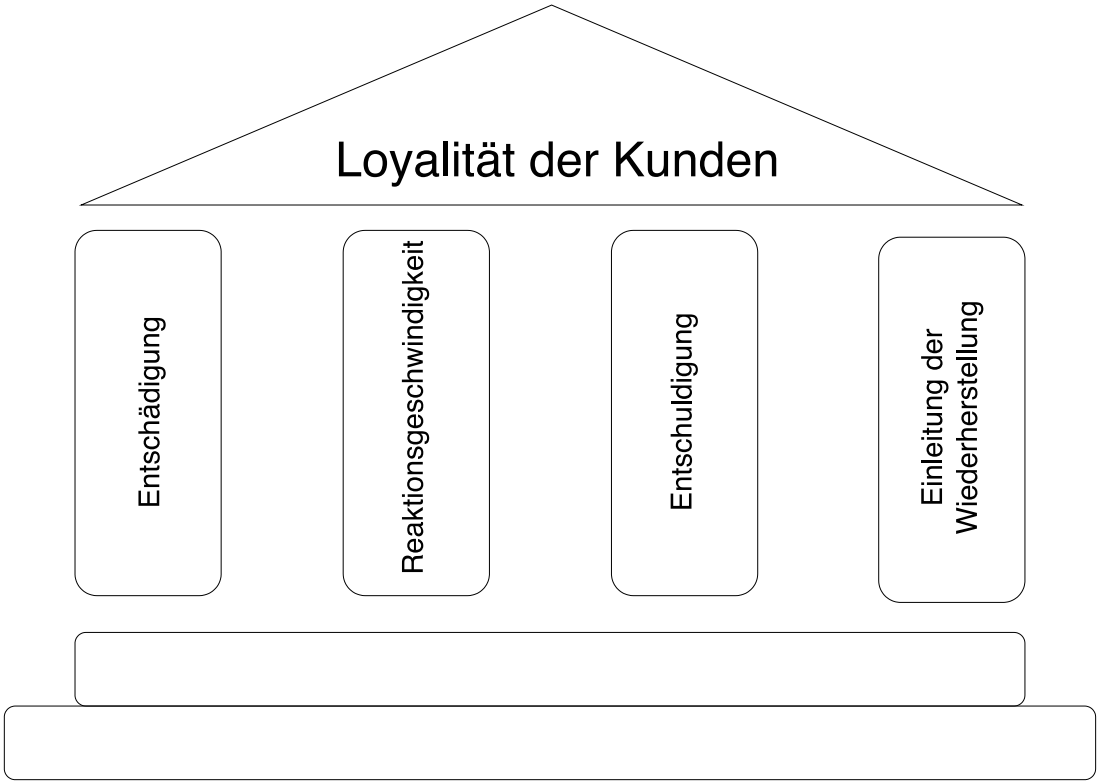


Abbildung 6 Faktoren für eine hohe Kunden Loyalität

Quelle: Eigene Darstellung

Die Arbeit wird zwei dieser vier Wichtigen Punkte betreffen. Reaktionsgeschwindigkeit und die Einleitung der Wiederherstellung werden zukünftig durch das Projekt verbessert. Diese zwei Punkte werden im Kapitel 1.5 ersichtlich, da sich einerseits die Reaktionsgeschwindigkeit des Technikers erhöht und andererseits die Zeit, die er für die Wiederherstellung des Netztes benötigt, sich verkürzt.

Die Angestellten erhalten so eine Erweiterung des Monitoring-Tools welche wesentlichen Auswirkungen auf ihre eigenen Reaktionszeiten ausübt.

2.2 Monitoring Tools

Unter diesem Kapitel wird auf drei der gängigsten und meistverbreiteten Monitoring-Tools (Vogel Communications Group, 2022) eingegangen. Die am besten bewerteten Lösungen, die in einer Umfrage im Jahre 2022 ermittelt wurden, (Gartner, 2022), werden vorgestellt. Sie weisen viele Ähnlichkeiten auf, weswegen hier nur auf besondere Eigenschaften eingegangen wird. Jene Monitoring-Tools, die vor allem für das Überwachen von Server und Applikationen konzipiert sind, wurden aus der Liste entnommen.

2.2.1 PRTG

PRTG bietet verschiedene Lösungen mit unterschiedlichen Optionen an. Je nach Anzahl der zu überwachenden Knoten, fallen unterschiedliche Preise an. Für die Ausgangslage mit ca. 10000 Geräten, wären einmalig 12’999.00 € für die Serverlizenz fällig. PRTG ermöglicht dem Nutzer eigene Dashboards zu erstellen. Somit bietet sich diesem die Möglichkeit, das Monitoring auf die eigenen Bedürfnisse zuzuschneiden. Unter Abbildung 7 ist ein solches Dashboard demonstrativ aufgeführt (PRTG, 2022).

Informationen in Echtzeit

Das Monitoring erfolgt in Echtzeit, heisst, zu jeder Zeit sind die Zustände der Geräte bekannt. Durch eigene Farbcodierungen besteht die Möglichkeit, den Fokus auf die Probleme zu richten, um so diese zu beheben, bevor sie ausbrechen. Durch verschiedene Darstellungen der Messwerte, können verschiedene Perspektiven und Interpretation von Daten sich ergeben.

Bestehende Bibliotheken

PRTG bietet mehr als 300 verschiedene Widgets die per Drag-and-drop in eine Karte hineingezogen werden können. Somit kann das Netz einfach über eine graphische Oberfläche erstellt werden. Diese ermöglichen es Verkehrsdiagramme, Toplisten oder benutzerdefinierte Bilder hinzuzufügen. Es besteht ausserdem die Möglichkeit, eigene Widgets zu erstellen.

Zugänge

Die erstellten Dashboards sind im Besitz von einer eindeutigen URL, die intern oder extern verwendet werden können. Somit könnten Informationen gruppiert und aufbereitet werden, um diese danach der Zuständigen Abteilung zur Verfügung zu stellen.

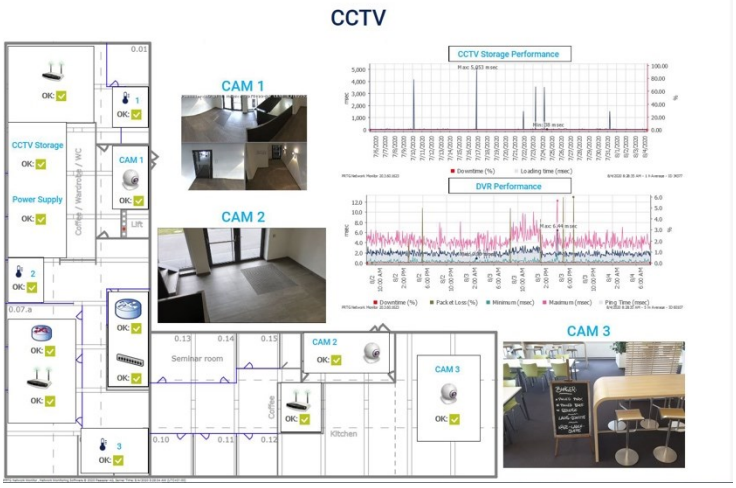


Abbildung 7 Dashboard in PRTG

2.2.2 Zabbix

Bei Zabbix handelt es sich um eine Open-Source Lösung, bei der die Schulungen und der Support kostet. Zabbix bietet auch die Möglichkeit, die Überwachten Geräte auf Karten mit genauen Positionen oder schematisch, wie auf der rechten Seite in Abbildung 8 ersichtlich, darzustellen (Zabbix, 2022).

Vorlagen

In Zabbix stehen viele Vorlagen von zähligen Herstellern zur Verfügung. Diese ermöglichen einen raschen Einsatz von Zabbix, da diese die Schnittstellen zu der Hardware der Hersteller bereits unterstützen.

Datenerfassung

Zabbix bietet praktische Funktionen wie unter anderem das Klonen von Netzwerkgeräten. Somit können Konfigurationen mit wenig Aufwand auf viele Netzwerkgeräte erfolgen. Die erfassten Metriken dieser Netzwerkgeräte können vorverarbeitet werden, heisst, diese Metriken können somit berechnet oder auch aggregiert werden.

Metriken

Metriken lassen sich dank Zabbix bestimmen und einstellen. Somit lassen sich die Wertebereiche, der einzelnen Metriken definieren, um optimal auf Störungen zu reagieren. Es bietet sich die Chance, auf wichtige Störungen zu reagieren und nicht von kleineren Fehlalarmierungen abgelenkt zu werden. Ausserdem werden Vorhersagen bezüglich Ausfallzeiten und Anomalien getätigt, die aufgrund der Netzwerkauslastung deutlich werden.

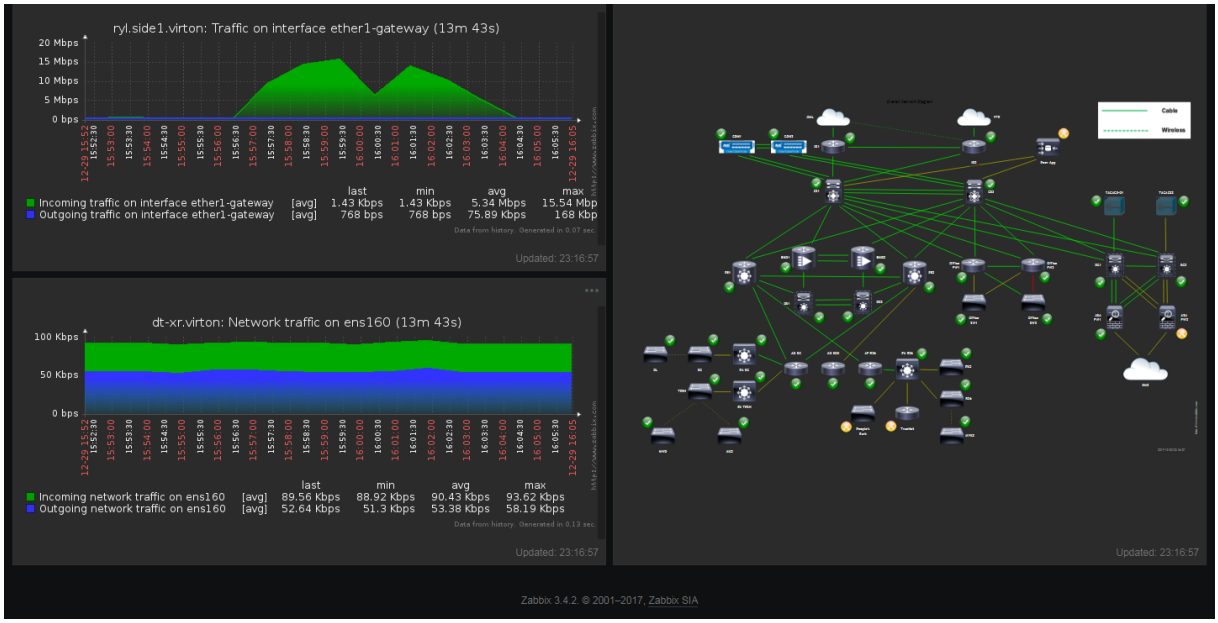


Abbildung 8 Dashboard in Zabbix

Quelle : https://www.eikona-logistics.de/files/bilder/news/zabbix.jpg

2.2.3 Checkmk

Checkmk wurde ursprünglich um den Nagios Kern entwickelt. Aktuell stehen drei unterschiedliche Editionen zur Verfügung. Die frühere Software und die aktuelle Open-Source Variante basiert immer noch auf den Kern von Nagios. Die kommerziellen Editionen Basieren hingegen auf einen von Checkmk entwickelten Kern. Dieser ist performanter als der Nagios Kern und bringt den Vorteil, dass dieser bei Konfigurationsänderungen nicht neu gestartet werden muss(Wikipedia-Autor, 2022c).

Business Logic

In Checkmk gibt es ein Modul (Business-Logic-Modul) welches in der grafischen Oberfläche integriert ist. Dieses ermöglicht dem Nutzer, alle Werte, die einen vom Unternehmen angebotenen Service ausmachen, auf einem Blick zu sehen. Ausserdem lassen sich Ausfälle einzelner Netzwerk Geräte simulieren, aus denen eventuelle Schwachstellen hervorgehen können.

Graphing

Checkmk bietet in den kommerziellen Editionen eigene Systeme zur Darstellung der Metriken. Ansonsten werden, da Checkmk auf Nagios basiert, wie bei früheren Versionen PNP4Nagios genutzt. Je nachdem Edition unterstützt Checkmk auch die Anbindung von Grafana.

Inventar

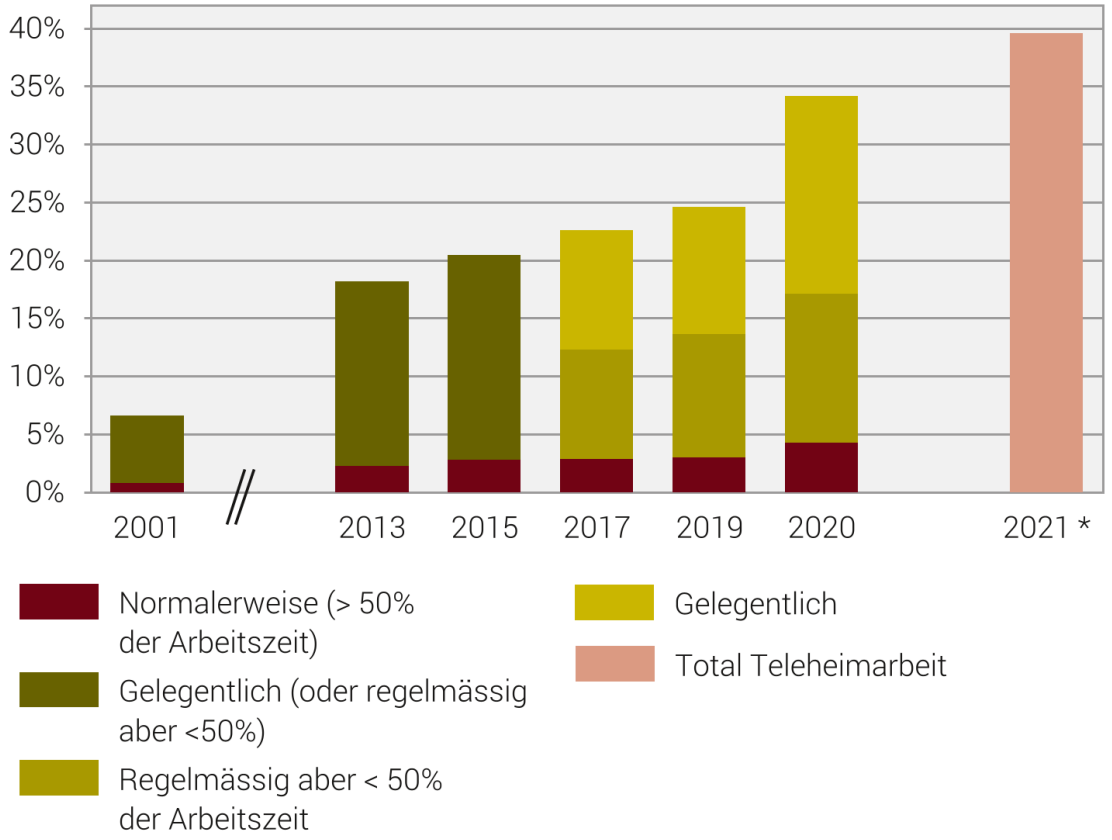
Checkmk ermöglicht es, den gesamten Bestand der Software- und Hardware zu ermitteln.
Automatische Updates können optimiert werden, indem das Update auf eine neue Version erst durchgeführt wird, wenn die Festplatten Auslastung niedrig ist.

2.3 Homeoffice

Laut dem Bundesamt für Statistik hat Corona die Heimarbeit begünstigt. So lag der totale Anteil der Erwerbstätigen, die von der Heimarbeit Gebrauch machten bei rund 25%. Zwei Jahre später, 2021, lag dieser mittlerweile auf knapp 40%. Wie sich das ganze zukünftig entwickelt wird sich noch zeigen. Der Abbildung 9 ist jedoch ein kontinuierlicher Trend zu mehr Heimarbeit zu entnehmen (Bundesamt für Statistik, 2022).
Das bedeutet, dass die Ansprüche an das HF-Netz steigen. Ausserdem wird durch die Abschaltung von Digital TV im Jahr 2022, die Anzahl von den IPTV-Geräten in die Höhe schnellen, was die Anforderungen des HF-Netzes weiter erhöht.

Teleheimarbeit der Erwerbstätigen, Entwicklung

Erwerbstätige (ohne Lehrlinge), in %



* Zeitreihenbruch: ab 2021 Änderung der Erhebungsmethode (mixed-mode) und Neuformulierung der Frage

Quelle: BFS – Schweizerische Arbeitskräfteerhebung (SAKE) © BFS 2022

Abbildung 9 Homeoffice zahlen

2.4 Visualisierung

Die Vorgabe des Arbeitgebers war es, die Visualisierungen mit d3.js, siehe Kapitel 8.1, durchzuführen. Aus diesem Grund werden die unterschiedlichen Visualisierungen, die eine hierarchische Map zulassen und die das Framework bietet, hier vorgestellt.

2.4.1 Tree

Bei der Darstellung als Tree werden die Daten, die Hierarchisch angeordnet sind, auch so abgebildet. Der erste Knoten gilt als Wurzelknoten, die letzten als Blatt Knoten.
Jeder Knoten wird mit einem anderen Knoten, über sogenannte Kanten verbunden (Wikipedia-Autor, 2022a).

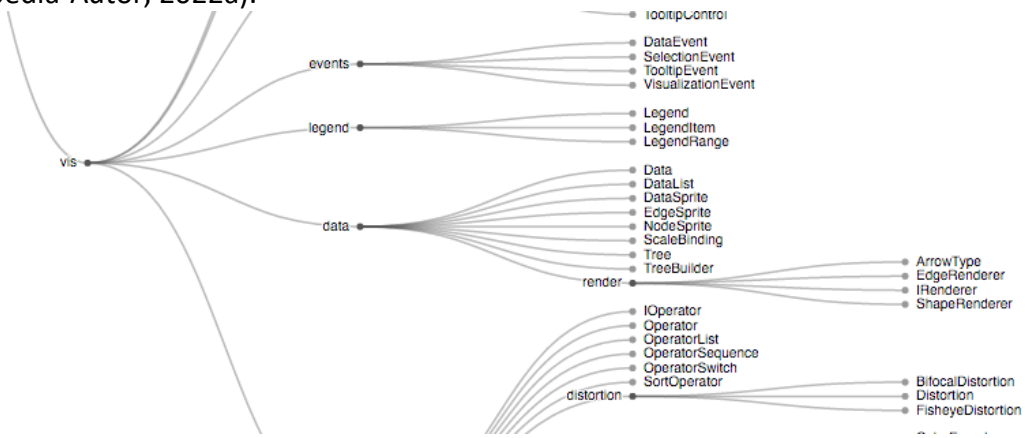


Abbildung 10 Tree

Quelle: https://raw.githubusercontent.com/d3/d3-hierarchy/main/img/tree.png

2.4.2 Treemap

Die wurde durch Ben Schneidermann erfunden, nachdem dieser die grössten Ressourcenfresser seiner Festplatte ermitteln wollte. So wird die Struktur der Daten und deren Grösse mit Kacheln dargestellt. Je grösser die Kachel, desto grösser die Datei. Die Zusammengehörigkeit oder Verschachtelung, kann hingegen mittels Farbwahl geschehen. Zur Illustration sind, siehe Abbildung 11, die Städte von Indien nach Grösse in einer Treemap abgebildet. Die unterschiedlichen Farben wurden in diesem Beispiel verwendet, um Städte in den dazugehörigen Bundesstaaten zu unterteilen (Wikipedia-Autor, 2017).

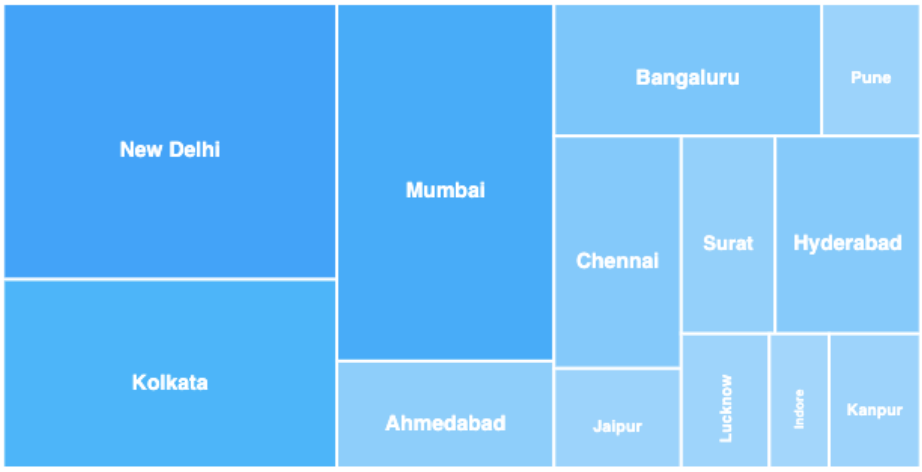


Abbildung 11 Treemap

Quelle: https://apexcharts.com/wp-content/uploads/2020/10/basic-treemap.png

2.4.3 Pack

Das Pack stellt die hierarchische Struktur der Daten dar, indem es jeden weiter verschachtelten Knoten, in den aktuellen Knoten abbildet. Durch dieses Vorgehen wird das Vergrössern einzelner Knoten notwendig. Der Nachteil dieser Darstellung liegt somit darin, dass eine Verzerrung stattfindet und nur die Blatt Knoten wahrheitsgetreu abgebildet werden können (d3, 2022)

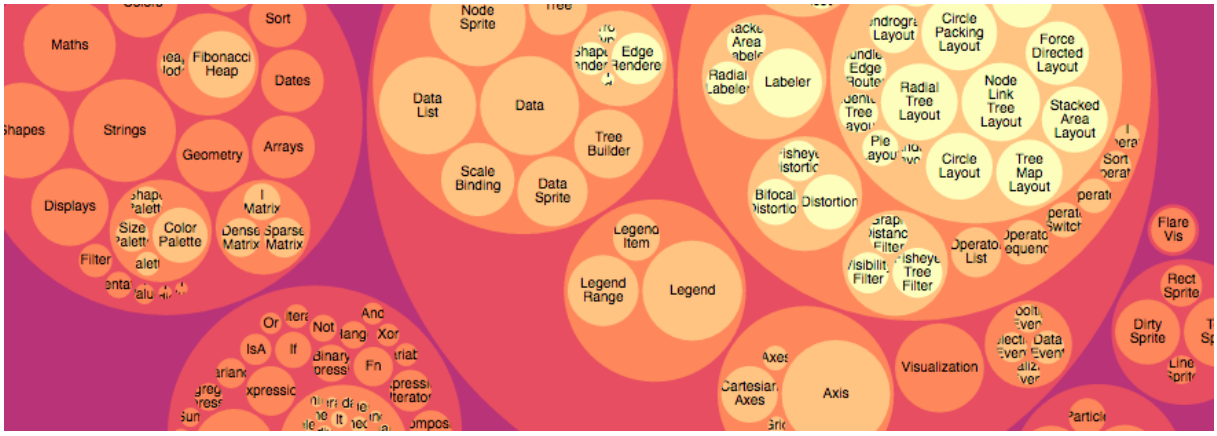


Abbildung 12 Pack

Quelle: https://raw.githubusercontent.com/d3/d3-hierarchy/main/img/pack.png

2.5 Messwerte

Damit die eine Aussage über die Qualität des Netzes getätigt werden kann, müssen vorher Messwerte und deren Wertebereich definiert werden (Wikipedia-Autor, 2021b). Die Messwerte der einzelnen Netzwerkgeräte, können über das SNMP-Protokoll oder auch Netzwerkverwaltungsprotokoll genannt, ausgelesen werden. Dieses Protokoll kümmert sich um die Kommunikation zwischen diesen (Wikipedia-Autor, 2022g).

2.6 Software

Unter diesem Kapitel wird die verwendete Softwarearchitektur beschrieben.

2.6.1 Fassade

Die Fassade, angesiedelt bei der Kategorie der Strukturmuster, dient zur Vereinfachung von Schnittstellen. Unter Schnittstellen sind solche gemeint, welche zu anderen Subsystemen liegen. Die Fassade hilft, die Verwendung dieser Subsysteme zu vereinfachen, indem es die Funktionen dieser Subsysteme (Fowler M. , 2022a). Somit, können Mehrere Subsysteme verwendet werden, ohne dass die genaueren Details dieser Subsysteme dem Nutzer bekannt sein müssen (Wikipedia-Autor, 2022b).

2.6.2 Adapter

Der Adapter gehört ebenfalls zu der Kategorie der Strukturmuster. Dieser wird vor allem dort verwendet, wo Klassen bereits existieren und wiederverwendet werden sollen. Der Vorteil liegt darin, dass die bestehenden Klassen dabei unverändert und der Adapter sich um die Einzelheiten (Wikipedia-Autor, 2021c).

2.6.3 DTO

Das Datentransferobjekt, kurz DTO, gehört zu der Pattern Kategorie der Entwurfsmuster. Es wird verwendet, um eine Anzahl von Daten in ein Objekt zu speichern. Der Vorteil liegt darin, dass die Daten erst nach der Aggregation, zur Verfügung gestellt werden. Somit verringern sich die Aufrufe an ein System auf genau einen. Aufgrund der Reduktion der Aufrufe werden die Ressourcen geschont (Fowler, https://martinfowler.com/, 2022b).

2.7 Tests

In diesem Kapitel werden die angewandten Testarten beschrieben.

2.7.1 Integrationstest

Die Integrationstest werden eingesetzt, um die Anwendung als Ganzes zu Testen. Das bedeutet, dass diese das Zusammenspiel verschiedener Systeme oder Module abdecken können(Pittet, 2022).

2.7.2 Funktionstests

Funktionstests dienen dazu, Endresultate zu verifizieren. Im Gegensatz zu den Integrationstest, welche vor allem das Zusammenspiel der Module prüfen, wird mit den Funktionstests die Richtigkeit der Outputs verifiziert (Pittet, 2022).

2.8 Projektlücke

Die Loyalität der Kunden hängt laut der Studie von (Buttle & Burton, 2001) von den vier Pfeilern, siehe Kapitel 2.1, ab. Dieses Projekt würde der Reaktionsgeschwindigkeit und Einleitung der Wiederherstellung zugutekommen. Die anderen zwei Faktoren würden andere Abteilungen, Kundendienst und Buchhaltung betreffen und sind somit nicht Teil dieser Arbeit.

Die drei Softwarelösungen, die in Kapitel 2.2 beschrieben werden, unterscheiden sich zwar in Ihren Ausführungen voneinander, funktionell ähneln sie sich hingegen. So werden unter anderem mittels SNMP-Anfragen auf die zu untersuchenden Netzwerkgeräte Daten gesammelt, interpretiert und grafisch dargestellt. Die Monitoring-Tools sind vermehrt auf die Architektur im Kern des Netztes ausgelegt. So stehen die Metriken für jegliche Netzwerk-Geräte wie Server, Switches und Router zur Verfügung, um diese Netzwerke abzubilden und zu bewachen. Für die aktiven Netzwerkkomponenten wie sie hier aggregiert werden, besitzen keine Logik oder Schnittstellen, die ansprechbar über diese Tools wären.
Die Informationen über die HF-Architektur ist in Datenbanken abgespeichert und müssen manuell ergänzt werden. Um die HF-Architektur inklusive der ganzen Abhängigkeiten in ein neues Monitoring Tool zu migrieren wäre unter den aktuellen Bedingungen unzumutbar.

Das Interesse an einer Abbildung des gesamten Koaxial-Netztes inklusive der Abhängigkeiten ist gross. Diese würde dem ganzen Unternehmen dienen.

Durch das Aufkommen von Homeoffice hat die Anzahl der Nutzer zugenommen (siehe Kapitel 2.3). Gleichzeitig konzentriert sich die Kommunikation durch die Abschaltung von Digital TV auf wenige Frequenzen im Breitband Netz. Das Projekt wird beitragen, dass Fehler besser erkannt und frühzeitig behoben werden können.

3 Ideen und Konzepte

Zu Beginn des Projektes, war das Konzept und Ziel bereits klar. Das verwendete Monitoring-Tool, eine Eigenentwicklung, ist bereits im Einsatz und soll weiterentwickelt werden. Wie das erfolgen soll, wird konzeptionell in diesem Kapitel beschrieben werden.

3.1 Roter Faden

Am Anfang wurde für das Projekt eine Grobplanung, hier aufgezählt, definiert. Die nachfolgenden Punkte dienten als roter Faden für die Arbeit:

1. Projektdokumentation erstellen
2. Anforderungskatalog erstellen
3. Den aktuellen Stand des Monitoring Tools Watchdog ermitteln
4. Andere Systeme als Variante recherchieren
5. Evaluation der Recherchen
6. Implementation
7. Validation
8. Testen
9. Präsentation

3.2 Zusammenhang Netz

Das HF-Netz ist gross und hängt wie ein Binärer Baum zusammen. Bestehend aus der Wurzel, in dem Fall dem DCMTS, die einzelnen Äste sind als Verstärker anzusehen und die Blätter, als die eigentlichen Kunden. Obwohl diese physisch zusammenhängen, gibt es keine visuelle Abbildung, die diesen Zusammenhang wieder gibt. Dieses Problem tritt auf, da nur auf den ersten Knoten, dem DCMTS und den letzten Knoten, den Modems der Kunden, über das SNMP-Protokoll zugegriffen werden kann. Alle anderen Komponenten besitzen keine Schnittstelle, die das Auslesen von Informationen ermöglichen. Aus diesem Grund ist es Sinnvoll, den Zusammenhang dieser Geräte darzustellen.

3.3 Netzwerk Störungen Koaxial

Koaxial-Störungen können sich auf unterschiedlichste Arten Manifestieren. Es gilt diese Störungen im Koaxial-Netz mittels definierten Metriken zu erkennen. Die Netzwerk-Geräte stellen sehr viele Metriken zur Verfügung, die auslesbar sind. Es gilt jedoch herauszufinden, welche von den Möglichen Metriken, von Bedeutung sind.

3.4 Darstellung Baum

Aktuell sind die Werte über das interne Monitoring-Tool auslesbar. Die Ansichten sind jedoch begrenzt, so lassen sich keine Zusammenhänge erkennen. Die Metriken sind in Listen oder als einzelne Werte Auslesbar. Falls ein Knoten untersucht werden soll, müssen die Werte einzeln manuell zusammengetragen werden. Je näher der zu untersuchende Knoten bei der Wurzel liegt, desto höher ist der Aufwand diesen zu untersuchen, da mehr Netzwerk-Geräte untersucht werden müssen. Diese Problematik wird in Abbildung 13 deutlich. Die Idee scheint sinnvoll, durch dieses Projekt das eingesetzte Monitoring-Tool Watchdog mit einer Baum-Ansicht zu erweitert.

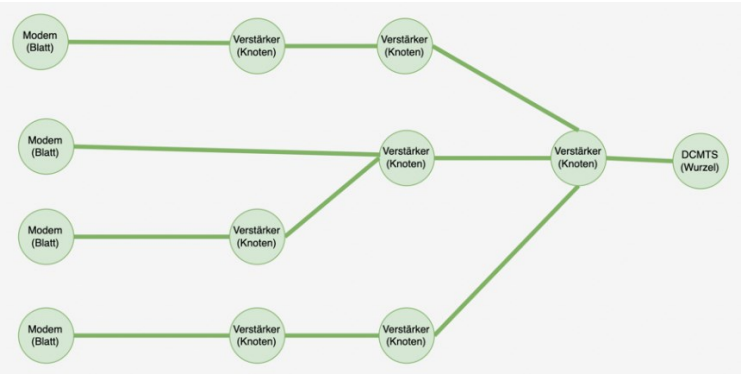


Abbildung 13 Darstellung als Baum

Quelle: Eigene Darstellung

4 Methoden

In diesem Kapitel werden die angewandten Methoden beschrieben, die zur erfolgreichen Umsetzung des Projekts beigetragen haben.

4.1 Projektmanagement

Hier werden die Methoden, die das Projektmanagement betreffen, beschrieben.

4.1.1 Soda

Soda ist ein Vorgehensmodell, was unter Anderem in der Informatik eingesetzt wird. Es leitet sich vom in der Software weit verbreiteten Vorgehensmodell Scrum ab. Der Unterschied zu Scrum liegt darin, dass zusätzlich ein Projektrahmen darum geschnürt wird(HSLU, 2022).

Soda unterteilt diesen Projektrahmen in Projektführung, hier violett in der Abbildung 14 eingefärbt und Projektdurchführung. Während sich die Projektführung mit den Aufgaben planen, steuern und kontrollieren befasst, kommen in der der Projektdurchführung zusätzlich die vier konventionellen Phasen Initialisierung, Konzeption, Realisierung und Einführung dazu (HSLU, 2022).

Soda ist ein iterativ-inkrementelles Vorgehensmodell der Software-Entwicklung, wobei die Bedürfnisse je nach Möglichkeiten der Beteiligten, agil ergänzt werden können (HSLU, 2022).

SoDa bringt zusätzlichen die konventionellen Phasen Initialisierung, Konzeption, Realisierung und Einführung mit. Die Phasen Konzeption und Realisierung ähneln hingegen dem Vorgehensmodell Scrum (HSLU, 2022).

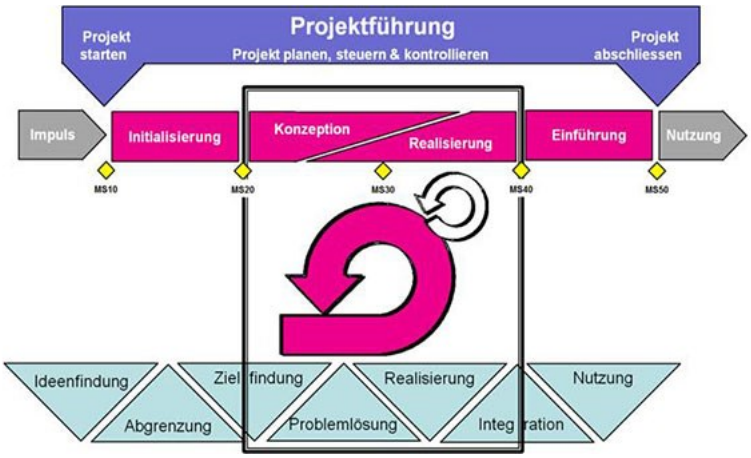


Abbildung 14 SoDa - Software Development Agile

4.1.1.1 Rollen

Die Rollen des Productowners, Scrummasters und des Agile Teams wurden in diesem Projekt einfachheitshalber und um den Aufwand gering zu halten, zusammengelegt.

4.1.1.2 Meilensteine:

Für die Arbeit wurden 5 Meilensteine definiert. Diese wurden nach den konventionellen Phasen von SoDa definiert. Da die Aufwände zur Erreichung der einzelnen Meilensteine sich stark voneinander unterscheiden, wurde der Realisierungsteil in zwei Meilensteine unterteilt.

Meilenstein 1: Initialisierung – Kickoff - 23.09.2022

- Kickoff
- Vorstellungsrunde

Meilenstein 2: Konzeption – Recherchen – 11.10.2022

- Einführung in das Thema.
- Die Vision
- Was gibt es auf dem Markt?
- Notwendigkeit des Projekts.
- Ideen.
- Methoden.
- Theorie der Metriken.

Meilenstein 3: Realisierung – Projektdurchführung – 24.12.2022

- Bestandesaufnahme.
- Projekt aufsetzten.
- Diagramme.
- Metriken.
- DB.
- Backend.
- Frontend.

Meilenstein 4: Realisierung – Validieren – 04.12.2022

- Evaluation
- Metriken überprüfen
- Vergleich der Arbeitsabläufe – Mehrwert?
- Abschluss und Ergänzungen der Arbeit

Meilenstein 5: Einführung – Ende – 03.01.2023

- Die Abgeschlossene Arbeit abgeben.
- Die Präsentation vorbereiten und halten.

4.1.2 Stand der Arbeit

Es werden regelmässig Sitzungen abgehalten, wo der aktuelle Stand der Arbeit besprochen wird. An diesen Sitzungen nehmen der betreuende Dozent Markus Waldmann und der interne Betreuer Evren Yilmaz teil. Es gilt, die Richtung des Projektes zu bewachen und bei Problemen frühzeitig zu reagieren.

4.2 Nutzwertanalyse

Die Nutzwertanalyse ist eine wissenschaftliche Methode, die dazu dient, rationale Entscheidungen zu einer bestimmten Lösung oder Thema zu treffen. Um dieses Vorgehen genauer zu erläutern, wurde die Tabelle 1 erstellt und dient als Beispiel.
In der Nutzwertanalyse werden die wichtigsten Kriterien zeilenweise aufgelistet. Da die Kriterien sich in ihrer Relevanz unterscheiden, werden diese gewichtet. Die Gewichte werden in Prozent angegeben. Die Summe all dieser Gewichte muss total 100 Prozent wieder geben. Je nach Einsatzgebiet, werden maximal 10 oder über 50 Kriterien als notwendig empfunden (Sonntag, 2015). Die verschiedenen Varianten, die miteinander verglichen werden, werden in den Spalten nebeneinander aufgelistet. Pro Variante werden die Spalten in 2 weitere Spalten aufgeteilt, die Bewertung und die daraus resultierenden Punkte für das jeweilige Kriterium.
In dieser Arbeit fällt die Bewertung der Kriterien zwischen eins und sechs aus. Angelehnt am Schweizer Notensystem gilt auch in dieser Arbeit die eins als ungenügend und die sechs als ausgezeichnet. So wird jedes Kriterium, was durch das dazugehörige Gewicht widergespiegelt wird, mit der Bewertung für die jeweilige Varianten multipliziert. Anschliessend werden alle Punkte aufsummiert und ergeben das Total der Variante. In Tabelle 1 ist illustrativ zu sehen, dass die Variante 1 Total 48 und die Variante 2, 148 Punkte für die ersten beiden Kriterien erhalten haben. Würden nur diese zwei Kriterien berücksichtigt werden, würde die Variante 2 als Sieger hervor gehen (Solis, 2022).

Tabelle 1 Nutzwertanalyse

Nr.	Kriterium	Gewicht (%)	Variante 1		Variante 2	
			Bewertung	Punkte	Bewertung	Punkte
1	Kriterium 1	25	2	25 * 2 = 50	6	6 * 25 = 150
2	Kriterium 2	8	6	8 * 6 = 48	6	8 * 6 = 48
…	…	…	…	…	…	…
		100 (Summe aller Gewichte)	Total Variante 1:	48	Total Variante 2:	148

4.3 Bedarfspegel und Einheitspegel

Beim Pegelkonzept eines HFC-Netzes wird, zwischen dem Bedarf- und dem Einheitspegel unterschieden. Beim Bedarfspegel werden die Verstärker alle individuell eingestellt, damit das Signal für alle angehängten Liegenschaften im gewünschten Wertebereich liegt. Beim Einheitspegel hingegen, wird der Verstärker immer gleich eingestellt. Die angeschlossenen Wohneinheiten müssen anhand der passiven Komponenten in den gewünschten Wertebereich geführt werden (SUISSEDIGITAL, 2016).

4.4 Tools

Hier werden die Tools abgebildet, siehe Tabelle 2, die zur Unterstützung des Projektes genutzt wurden.

Tabelle 2 verwendete Tools

Tools	Verwendung
Onedrive - Hochschule Luzern	Ablage für Dokumente
Microsoft office	Dokumentation, Präsentation, Tabellen
Teams	Kommunikationstool
Zoom	Kommunikationstool
GitLab (Databaar)	Projektmanagement
draw.io	Diagramme und Grafiken
PyCharm	Backend
IntelliJ	Frontend
Workbench	Datenbanken

4.5 Risikoanalyse

In der Initialisierungsphase wurde eine Risikoanalyse durchgeführt. Die Risiken wurden in Tabelle 3 dokumentiert und anschliessend in einer Risikomatrix, siehe Abbildung 15, visualisiert. Die Risiken wurden, je nach Eintrittswahrscheinlichkeit (EW) und der Schadenhöhe (SH) kategorisiert. Von grün, was einem geringen Risiko, bis hin zum dunkelroten, sehr hohen Risiko.

Schadenhöhe	Extrem	5			1.4	1.1	
	Bedeutend	4				1.3	
	Moderat	3			1.2 / 1.5		
	Gering	2					
	Nebensächlich	1					
			1	2	3	4	5
			Selten	un-wahrscheinlich	Möglich	Wahrscheinlich	Sehr wahrscheinlich
			Eintrittswahrscheinlichkeit				

Abbildung 15 Risikomatrix (ohne Massnahmen)

Tabelle 3 Risiken

Nr.	Risiko	Beschreibung	EW	SH	Bewertung
1.1	Ausfall Projektmitglied	Durch Krankheit fällt ein Projektmitglied aus	4	5	Sehr hoch
1.2	Projektaufwand zu hoch	Falls der Projektaufwand höher liegt als die eingeplante Zeit	3	3	Hoch
1.3	Unvorhergesehene Anforderungen	Anforderungen, die erst im Laufe der Zeit sich bemerkbar machen und sich negativ auf die erfolgreiche Projektdurchführung sich auswirken.	4	4	Sehr hoch
1.4	Dokumentation	Dokumentation nicht ausreichend geführt	3	5	Hoch
1.5	nicht umsetzbar	Projekt mit den zu Verfügung stehenden Mitteln und Ressourcen nicht umsetzbar.	3	3	Hoch

Nach der Ermittlung der Risiken, wurden die dazugehörigen Massnahmen, siehe Tabelle 4, ausgearbeitet. Diese Massnahmen wurden wieder in einer Risikomatrix, Abbildung 16, dargestellt. Diese wurden mit dem Arbeitgeber besprochen.

Schadenhöhe	Extrem	5					
	Bedeutend	4					
	Moderat	3		1.4			
	Gering	2	1.1 / 1.2	1.3			
	Nebensächlich	1	1.5				
			1	2	3	4	5
			Selten	Un-wahrscheinlich	Möglich	Wahrscheinlich	Sehr wahrscheinlich
			Eintrittswahrscheinlichkeit				

Abbildung 16 Risikomatrix (mit Massnahmen)

Tabelle 4 Massnahmen

Nr.	Massnahmen	EW	SH	Bewertung
1.1	Das Projekt früh beginnen, um im Besitz aller nötigen Informationen zu sein.	1	2	Gering
1.2	Zwischenlösungen evaluieren und beschreiben.	1	2	Gering
1.3	Anforderungsanalyse durchführen.	2	2	Mittel
1.4	Die Dokumentation frühzeitig beginnen und kontinuierlich nachtragen.	1	3	Mittel
1.5	Über die Einzusetzenden Technologien frühzeitig Recherchen durchführen.	1	1	Gering

4.6 Softwareentwicklung

Für die Modellierung und Abstraktion des Projektes werden unterschiedliche Diagramme verwendet. Da das Monitoring-Tool Watchdog sensible Daten zur Verfügung stellt, dürfen die Diagramme bedingt nach aussen kommuniziert werden. Aus diesem Grund werden die Diagramme zum Teil vereinfacht dargestellt.

UML unterteilt die Diagramme grob in zwei Gruppen. Einerseits die Strukturdiagramme und andererseits die Verhaltensdiagramme. Die Grenzen sind nicht so streng definiert und so kann es vorkommen, dass ein Diagramm Eigenschaften aus beiden Gruppen enthält. („Unified Modeling Language“, 2022)

4.6.1 Kontextdiagramm

Um die Schnittstellen des Systems mit der Umwelt besser darzustellen, wird in diesem Projekt auf Kontextdiagramme zugegriffen (Wikipedia-Autor, 2021a).

4.6.2 Komponentendiagramm

Um die Struktur des Systems aufzuzeigen, wird das Komponentendiagramm erstellt. Dabei werden Komponenten und die dazu gehörigen Schnittstellen aufgeführt. Somit werden auch weitere Abhängigkeiten sichtbar. Da das Produkt am Ende eine eigene Komponente sein wird, ist es sinnvoll dieses Diagramm zu wählen (Wikipedia-Autor, 2022e).

4.6.3 Sequenzdiagramm

Dem Sequenzdiagramm ist der Austausch von Nachrichten im System zu entnehmen. So werden die ausgetauschten Nachrichten zwischen den Objekten ersichtlich. Ausserdem ermöglicht das Sequenzdiagramm es, die Nachrichten und deren Reihenfolge genau zu bestimmen. Das ist für das Projekt wichtig da es dem Verständnis des Programmcodes dient (Wikipedia-Autor, 2022j).

4.6.4 Use Case Diagramm

Use Case Diagramme dienen dazu, die Struktur und Verhalten von Systemen mit Ihren Nutzern darzustellen. So werden Beziehungen oder Abhängigkeiten dargestellt, die der Nutzer zu der dargestellten Software hat (Wikipedia-Autor, 2022d) («Anwendungsfalldiagramm», 2022).

4.7 Tests

Die Visualisierungen werden am Ende mittels Integrations- und Funktionstest überprüft.

5 Realisierung

5.1 Gantt Diagramm

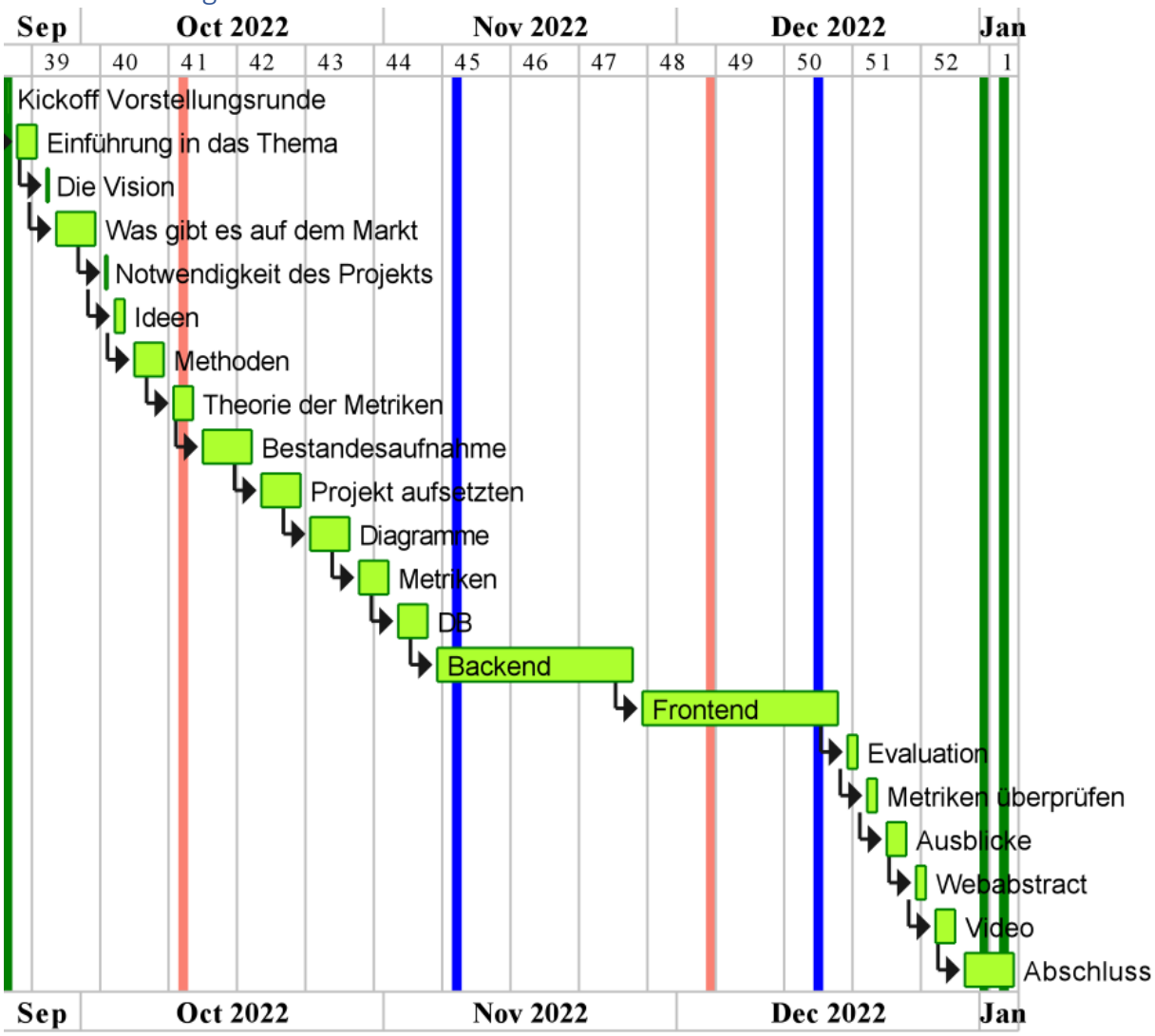


Abbildung 17 Gantt Diagramm

Quelle: Eigene Darstellung

In diesem Gantt Diagramm, Abbildung 17, sind die Meilensteine und Aufgaben visuell dargestellt. Die Meilensteine wurden, wie in 4.1 beschrieben. Die Meilensteine, die eingehalten wurden, sind grün markiert. Die roten Linien weisen auf die Daten der geplanten Meilensteine hin, während die blauen Linien, auf die effektiven Daten des Meilensteins hinweisen. Somit ist zu erkennen, dass der erste Meilenstein um drei Wochen und der zweite Meilenstein, sich um 2 Wochen nach hinten verschoben hat. Das hat unter anderem mit der Bestandesaufnahme, dem Aufsetzten des Projektes und der Einarbeitung zu tun, welche die drei verschiedenen Frameworks mit sich gebracht haben.

5.2 Arbeitsrapport

Unter diesem Kapitel ist der Arbeitsrapport, wie in im Anforderungskatalog in Kapitel 8.1 gefordert, abgebildet. Die Aufgaben, die Probleme bereitet haben und sich negativ auf die Durchführung des Projektes ausgewirkt haben, wurden in Tabelle 5, unter der Spalte «Probleme» vermerkt.

Tabelle 5 Arbeitsrapport

Datum	AWH	Aufgaben	Probleme
22.09.2022	2	Projektvorbereitung	-
23.09.2022	2	Kickoff Meeting (Waldmann, Yilmaz, Krasniqi)	-
24.09.2022	1	Doku starten. PDF erstellt mit Kapiteln & Formatierungen	-
24.09.2022	1	Protokollieren (Rein schreiben)	-
24.09.2022	2	1. Problem, Fragestellung, Vision	-
24.09.2022	1	OneDrive	-
24.09.2022	2	1.1 Problem	-
24.09.2022	3	HF-Netz Grafik	-
24.09.2022	1	Vision	-
30.09.2022	9	Verunreinigte Daten suchen	-
31.09.22	8	Daten bereinigen	vergleich PSuite-DB-Watchdog
05.10.2022	2	Graphen (Problem, Ausgangslage, Fragestellung)	-
05.10.2022	1	Problem	-
05.10.2022	4	Ausgangslage	grosses Projekt mit unzähligen Abhängigkeiten
06.10.2022	1	Fragestellung	-
06.10.2022	1	Systemgrenzen	-
06.10.2022	4	Vision (Text, Grafiken)	-
07.10.2022	3	Zotero	-
08.10.2022	3	Experte (Buchmann)	-
19.10.2022 bis 29.10.2022	18.5	Recherchen	-
01.11.2022	3	Ideen	-
02.11.2022 bis 05.11.2022	12	Methoden	-
08.11.2022 bis 14.11.2022	42	backend	Probleme mit dem Framework, Know-how
15.11.2022	10	Refactoring backend	Troubleshootern

16.11.2022	8	Threads & Graphen	Probleme mit Threading bei zwei Listen
17.11.2022	7	Threads	-
18.11.2022	8	Refactoring & Graphen	Tool für Graphen recherchiert
23.11.2022	6	Graphen & Projektmanagement	-
24.11.2022	4	Statusmeeting & Nacharbeit	-
25.11.2022	4	Zwischenpräsentation inkl.	-
27.11.2022	5	Einarbeitung und aufsetzen Frontend	-
28.11.2022 bis 29.11.2022	9	Recherchen Frontend	-
01.12.2022 bis 06.12.2022	39	Frontend	Neu zeichnen problematisch,
06.12.2022	7	backend	Fehler US SNR, Fehler Verschachtlung, Adapter
15.12.2022	2	Präsentation	-
19.12.2022	4	Test	-
19.12.2022	2	Schulung	-
19.12.2022 bis 03.01.2022	126	Doku	-
			-
TOTAL	**368**		-

5.3 Mögliche Variante

In diesem Kapitel werden, die unter Kapitel 2.2 vorgestellten, Varianten mittels Nutzwertanalyse bewertet, um die geeignete Variante zu ermitteln.

Tabelle 6 Nutzwertanalyse der Varianten

Nr.	Kriterium	Gewicht (%)	Software PRTG		Software Zabbix		Software checkmk		Software Watchdog	
			Bewertung	Punkte	Bewertung	Punkte	Bewertung	Punkte	Bewertung	Punkte
1	Preis der Lizenz	15	3	45	6	90	3	45	4	60
2	Erfahrung des Lieferanten	7	6	42	6	42	6	42	4	28
3	Support – Knowhow	10	5	50	5	50	5	50	5	50
4	Support - Kosten	12	2	24	3	36	3	36	6	72
5	Dokumentation	14	5	70	5	70	6	84	3	42
6	Implementierung	10	4	40	2	20	3	30	6	60
7	Source Code	3	4	12	4	12	4	12	6	18
8	Betriebssystem	5	6	30	6	30	6	30	5	25
9	Anzahl der Knoten	13	4	52	5	65	5	65	6	78
10	Intuitive Benutzeroberflächen	11	6	66	5	55	6	66	6	66
Nutzwert		100	Total PRTG:	431	Total Zabbix:	470	Total checkmk:	460	Total Watchdog:	499

Zur rationalen Ermittlung der geeignetsten Varianten wurden 10 Kernkriterien definiert. Die Varianten PRTG, Zabbix und checkmk in sind externe Lösungen, während es sich beim Watchdog um ein eigens entwickeltes Monitoring Tool handelt.
Jede der möglichen Varianten hat stärken und Schwächen, auf die in diesem Kapitel eingegangen wird. Grundsätzlich zeigt die Nutzwertanalyse auf, dass unter diesen gewählten Kriterien, die Eigenentwicklung im Watchdog aufgrund der höchsten Anzahl an Punkten, sich am besten eignen würde.

Die Kern Kriterien, welche die Softwarelösung erfüllen muss, wurden in Zusammenarbeit mit dem Projektbetreuer bestimmt.

Jede Variante hatte eine Besonderheit, was sie aus der Masse herausstechen ließ. Diese wurde in der Bewertung mit der dazugehörigen Variantenfarbe markiert.

5.3.1 Die einzelnen Varianten

Hier werden die Vorteile und Nachsteile der einzelnen Varianten, die in Kapitel 2.2 vorgestellt wurden, mit der Bewertung von Tabelle 6, weiter erläutert. Anschliessend wird erklärt, welche dieser Varianten sich durchgesetzt hat.

5.3.1.1 PRTG:

PRTG besticht vor allem durch die Intuitive Benutzeroberfläche. Diesem Punkt kommt vor allem zugute, dass im PRTG die Visualisierungen eine URL besitzen. Dies ermöglicht es, diese überall einzubinden. Der grösste Nachteil ist der Preis, der je nach Anzahl der Sensoren und deren Messwerte, in die Höhe schnellt. So würden, aufgrund der zu bewachenden Sensoren, laut der angebotenen Lösung 29'900.00€ pro Jahr, fällig werden. Näheres wird in Kapitel 2.2.1 vorgestellt.

5.3.1.2 Zabbix

Der Vorteil von Zabbix liegt darin, dass es sich um eine Open Source Lösung handelt und dadurch keine Lizenz benötigt wird. Es ist ausserdem weite verbreitet und hat sich als Monitoring Tool etabliert, was sich auch durch die Referenzen seiner Nutzer wieder spiegelt. Dass Zabbix viel bietet, wird der Software gleichzeitig zum Verhängnis. Es ist nämlich so mächtig, im Sinne seiner Möglichkeiten, dass der Einsatz all diese Funktionen vorher gekonnt sein müssen. Dies bezüglich bietet Zabbix Schulungen und Support an, der aufbauend ist und kostenpflichtig ist.

5.3.1.3 checkmk

Was am meisten für checkmk spricht ist die Tatsache, dass checkmk sehr gut und auf Deutsch dokumentiert ist. Es werden ganze Benutzerhandbücher online auf deren Homepage zur Verfügung gestellt, welche über die Suchfunktion durchsucht, werden können. Einziger negativer Punkt sind auch hier die Kosten, die je nach Service und Support anfallen.

5.3.1.4 Watchdog

Beim Monitoring Tool Watchdog handelt es sich um eine Eigenentwicklung. Der Vorteil liegt darin, dass die volle Kontrolle über die Daten, die Funktionalität und die Ansichten beim eigenen Unternehmen liegt. Der Nachteil an dieser Lösung, ist die Wartung dieser Software, die nicht vernachlässigt werden darf. Bei jeder Änderung der Begebenheiten, sei es Software oder Hardware bedingt, müssend diese vom Arbeitgeber selbst im Code eingepflegt werden.

5.3.2 Ergebnis Nutzwertanalyse - Varianten

Die Entscheidung fiel, vor allem wegen dem kleineren Aufwand und der geringeren Kosten, auf die Eigenentwickelte Software Watchdog. Langfristig gesehen, wurden jedoch Interessen Bekundungen getätigt, Zabbix zu testen. Zabbix überzeugt durch das breite Spektrum an Funktionen und der Tatsache, dass es sich dabei um eine Open Source Lösung handelt.

5.3.3 Allgemein über die Monitoring Tools

Die Monitoring-tools, die im Kapitel 2.2 beschrieben wurden, bieten zwar verschiedene auslesbare Parameter an, diese sind jedoch bedingt für die Darstellung des gesamten Netztes angedacht. Das ist unter anderem an der Anzahl der Knoten erkennbar, wodurch sie eine beschränkte Anzahl zur Verfügung stellen. Die untersuchten Monitoring Tools sind für Netzwerk Architekturen gedacht, bei welchen die Netzwerk-Geräte eine Schnittstelle besitzen, über die Daten ausgelesen werden können. Das zu untersuchende Koaxial-Netz ist jedoch, bis auf den Ersten Knoten und die letzten Knoten den Modems, nicht auslesbar.

In dieser Arbeit wurde die Bewertung der einzelnen Varianten aufgrund der öffentlich zugänglichen Informationen getätigt. Die Monitoring Tools wurden somit nicht installiert und getestet, da diese aufgrund der Sicherheitsmassnahmen nur mit hohem Aufwand hätten implementiert werden können und es ausserdem das Zeitbudget nicht zuliess.

Die Sicherheitsvorkehrungen, um im internen Netzt auf die verschiedenen Devices zu gelangen und diese Messungen zu tätigten, waren in der kurzen Zeit nicht möglich.

Der Arbeitgeber war bekundete Interesse an Zabbix und nach diesem Projekt, wird dieses Monitoring Tool intern getestet.

5.4 Mögliche Visualisierung

In diesem Kapitel wird auf die Wahl der Visualisierung eingegangen. Es gibt mehrere Möglichkeiten das Koaxialnetz, was in dieser Arbeit analog einem Baum entspricht, darzustellen. Dieser hängt hierarchisch zusammen und kann auf unterschiedliche Weise abgebildet werden. Im Monitoring Tool Watchdog wird für die Visualisierung der Daten die JavaScript Library d3.js verwendet und wurde auch in diesem Projekt, siehe Kapitel 8.1, vorgegeben. Folglich werden drei Arten bewertet, die trotz mehrerer Verschachtlungen, der Übersicht gerecht werden. Unter dem Kapitel 2.4 wurden diese Visualisierungen vorgestellt und werden hier ausgewertet.

Tabelle 7 Nutzwertanalyse Visualisierung

Nr.	Kriterium	Gewicht (%)	Tree		Treemap		Pack	
			Bewertung	Punkte	Bewertung	Punkte	Bewertung	Punkte
1	Modems abbilden	5	6	30	3	15	4	20
2	Häuser abbilden	5	6	30	3	15	4	20
3	Verstärker abbilden	5	6	30	3	15	4	20
4	DCMTS (Wurzel) abbilden	5	6	30	3	15	4	20
5	Verschachtelung ersichtlich	20	6	120	2	40	4	80
6	verschiedene Dimensionen	10	5	50	5	50	5	50
7	Metriken - abbildbar	10	5	50	5	50	5	50
8	Metriken - wählbar	15	6	90	6	90	6	90
9	Intuitiv	10	6	60	2	20	4	40
10	Analogie zum Koaxialnetz	15	6	90	2	30	4	60
Nutzwert		100	Total Tree:	580	Total Treemap:	340	Total Pack:	450

Die Visualisierung muss die einzelnen HF-Komponenten abbilden können. Das sind Modems, Häuser, Verstärker und die sogenannten DCMTS. Das Netz ist verschachtelt, daher müssen die Visualisierungen diese abbilden können. Unter dem Punkt verschiedene Dimensionen sind erweiterte Informationen, die in derselben Ansicht dargestellt werden können, ohne diese zu beeinträchtigen. Das wichtigste Kriterien waren «Verschachtelung», «Metriken» und «Analogie zum Koaxialnetz», die durch die Visualisierung stattfinden sollte. Dem entsprechend gaben diese am meisten Punkte.

5.4.1 Ergebnis Nutzwertanalyse – Visualisierung

Die Visualisierung Tree wurde diesen Kriterien am besten gerecht. Die anderen wären auch im Stande das Koaxialnetz abzubilden, bei einer kleinen Anzahl von Knoten. Falls diese Anzahl an Knoten jedoch nach oben schnellt, verbüssen die Varianten Treemap und Pack am meisten an Übersicht. Aus diesen Gründen fiel die Wahl auf den Tree.

5.5 Analyse IST-Zustand

Für die Analyse des IST-Zustandes, wurde der Watchdog, mittels der vorgestellten Methoden in Kapitel 4.6, analysiert. Der Watchdog befindet sich auf intern auf einem Server und ist nur vom Intranet zugänglich. Die Software wird in Front- und Backend aufgeteilt. Das Backend dient der Datenbeschaffung auf den Datenbanken, während das Frontend sich für deren Darstellung kümmert. Für das Frontend wird das Framework von Angular genutzt, wo mit der Programmiersprache TypeScript gearbeitet wird. Im Backend wird das Framework Django eingesetzt, wo in der Programmiersprache Python entwickelt werden kann. Bei den Datenbanken sind unterschiedlich im Einsatz. Für Daten, die eine Historie besitzen, sprich solche welche wiederkehrend mit einem Zeitstempel gespeichert werden, kommt RRDTool zum Einsatz. Bei allen anderen kommen MySQL Datenbanken zum Einsatz. Die Vorgaben bezüglich dieser Technologien ist in Kapitel 8.1 vermerkt und wurde vom Arbeitgeber vorgegeben.

5.6 Systemkontextdiagramm

Mittels dem Systemkontextdiagramm, siehe 4.6.1, werden die Grenzen des Systems, hier der Watchdog, aufgezeigt. Das System nutzt weitere externe Komponenten, welche aus Gründen der Übersichtlichkeit weggelassen wurden. So werden nur die sichtbar, die für das Projekt relevant sind, aufgeführt. Des Weiteren sind die Hauptnutzer, der Techniker und sein direkter Vorgesetzter, ersichtlich.

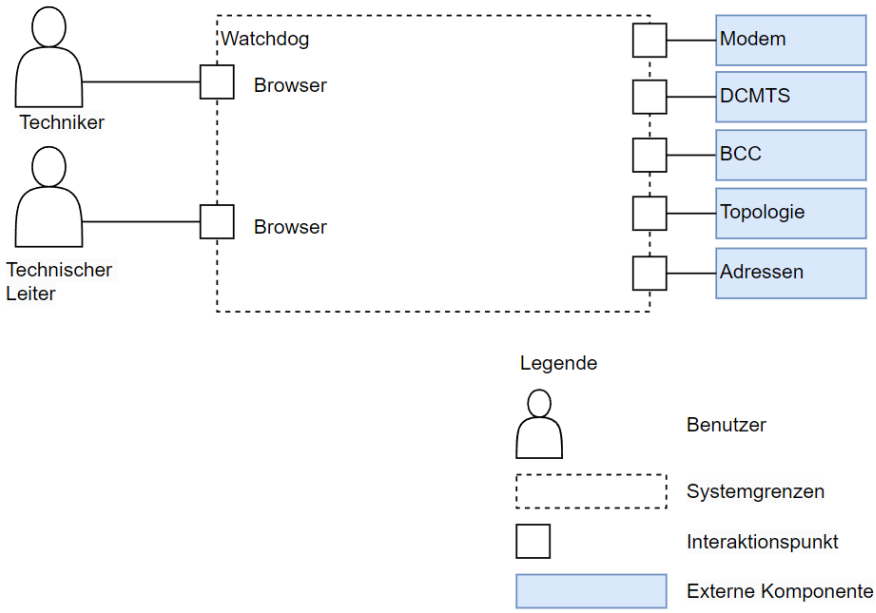


Abbildung 18 Systemgrenzen

Quelle: In Anlehnung an: https://image.jimcdn.com/app/cms/image/transf/dimension=605x1024:format=jpg/path/s7637a3fcec5efe31/image/i9bf2e4362c45993c/version/1526649822/image.jpg

5.7 Prozesse und Szenarien

Hier werden Prozesse anhand Use Case Diagrammen aufgezeigt und erläutert, welche in 4.6.4 beschrieben wurde, eingesetzt.

5.7.1 Use Case - IST

Aktuell kann der Technische Mitarbeiter über das System Watchdog das Netzt bewachen. Der Techniker kann jedoch nur einzelne Bereiche einsehen und hat keine Möglichkeit, eine komplette Übersicht über das HF-Netz zu erlangen. Ausser, so wie das bis anhin gemacht wird, die Informationen manuell auf Papier zeichnen. Beide Seiten, der Angestellte und sein Vorgesetzter kommen unter diesen Bedingungen erschwert an Informationen.

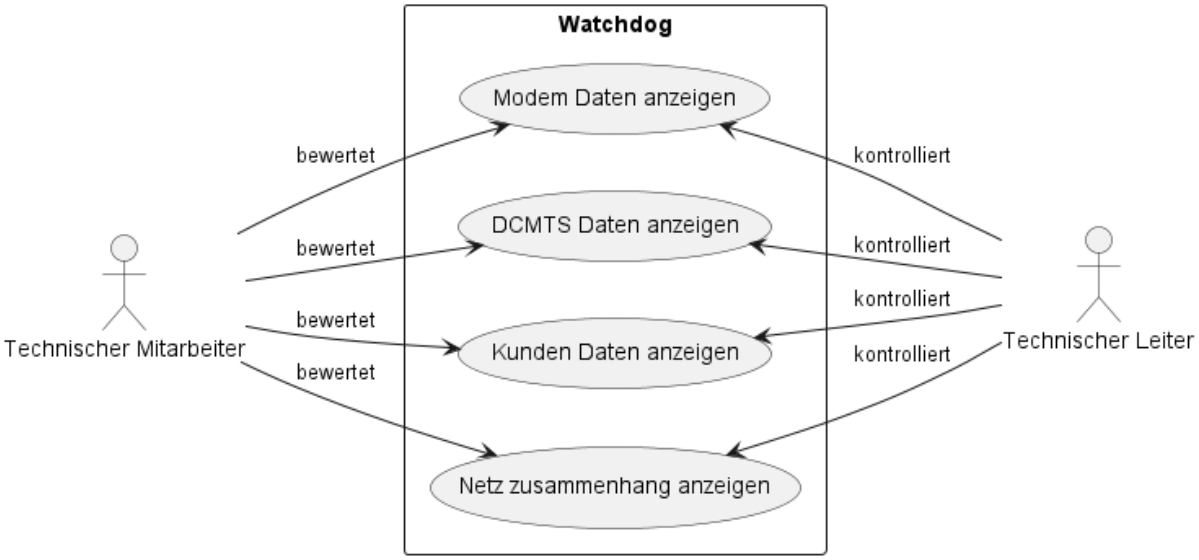


Abbildung 19 Use Case - IST

Quelle: Eigene Darstellung

5.7.2 Use Case - SOLL

Nach diesem Projekt wird der Ablauf so sein, dass die Aggregation aller notwendigen Informationen über das System selbst stattfinden wird.

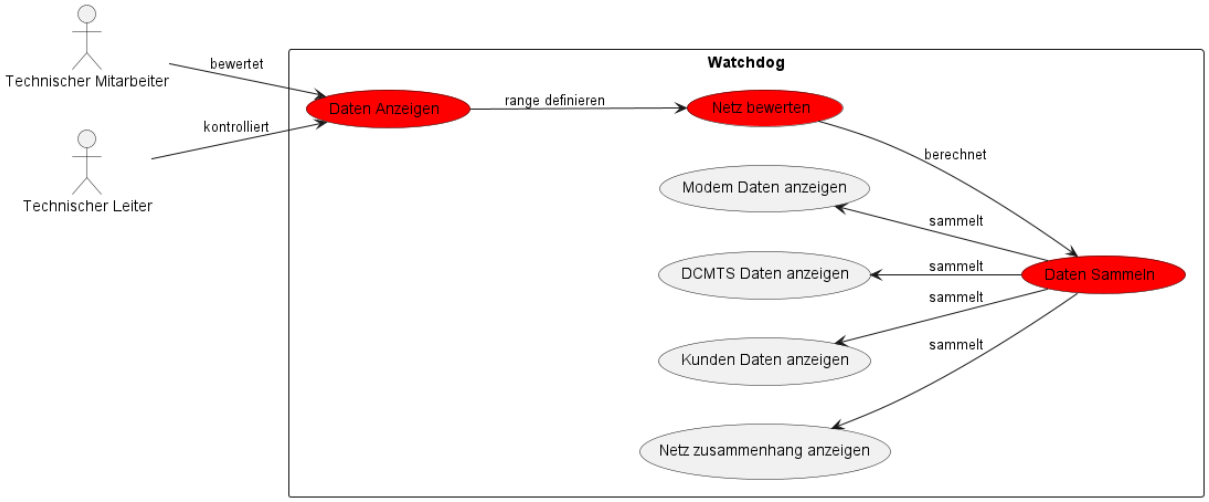


Abbildung 20 Use Case - SOLL

Quelle: Eigene Abbildung

5.8 Messwerte ermitteln

Da die Messwerte jederzeit erweiterbar sind, galt für das Projekt, die Visualisierungen mit bereits verwendeten Messwerten zu gestalten. Die Schwierigkeit lag darin, diese zu sammeln und zu aggregieren. Die Messwerte werden an verschiedenen Stellen gesammelt. Sie werden ausserdem in vier Kategorien unterteilt:

- Signalstärke – Hinweg
- Signalstärke – Rückweg
- Signalqualität – Hinweg
- Signalqualität – Rückweg

Unter Hinweg wird die Richtung des Signals, von der Firma aus hin zum Kunden beschrieben. Beim Rückweg handelt es sich um die entgegengesetzte Richtung, also alles, was vom Kunden aus zur Firma gesendet wird.

Falls Messwerte ausserhalb des definierten Wertebereichs liegen, sind diese ein Indikator dafür, dass an der Verbindung etwas nicht stimmt. Das können unterschiedliche Ursachen zur Folge haben.

5.9 Pegelkonzept

Der Auftraggeber verwendet das Pegelkonzept Einheitspegel, was unter dem Kapitel 4.3 beschrieben wird. Somit sind alle Verstärker, mit einzelnen Ausnahmen, gleich eingestellt. Falls sich die Signale zwischen Sender und Empfänger, in diesem Fall zwischen Verstärker und Modem nicht in einem vordefinierten Werteberiech liegen, ist dies ein Indiz dafür, dass zwischen diesen beiden Komponenten etwas nicht stimmt. Allein dieses vom Unternehmen eingesetzte Pegelkonzept ermöglicht es, Rückschlüsse auf einzelne Verstärker zu ziehen, da die Einstellungen der Signalstärke im Vorhinein für alle Netzwerk-Komponenten gleich sind.

Die Messwerte und deren Eigenschaften wurden in dieser Arbeit vereinfacht und konzeptionell dargestellt, da die Theorie dazu weit in die Elektrotechnik reicht und diese nicht Teil dieser Arbeit sind.

5.10 Komponentendiagramm backend

Unter diesem Kapitel werden die wichtigsten Komponenten, unter Berücksichtigung von Kapitel 4.6.1, beschrieben und deren Designentscheid begründet.

5.10.1 Komponentendiagramm IST

Unter Abbildung 21 sind die Komponenten abgebildet, die für das Projekt relevant sind und dieses Betreffen. Die einzelnen Komponenten sind in sogenannten Domänen aufgeteilt. Jede dieser Domänen deckt einen in sich geschlossenen Bereich ab. Das ermöglicht eine lose Kopplung und hohe Kohäsion der einzelnen Komponenten zu erreichen. Der Nachteil dabei, die Informationen müssen vom Techniker selbst aggregiert werden, da zusammenhängende Informationen erschwert einsehbar sind.

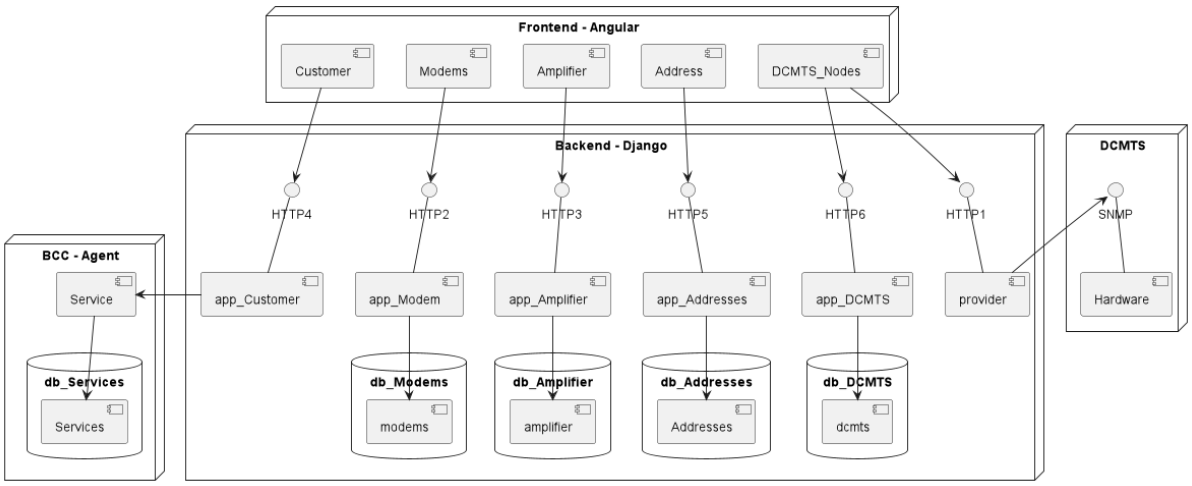


Abbildung 21 Komponentendiagramm

Quelle: Eigene Darstellung

5.10.2 Komponentendiagramm SOLL

Für die neue Ansicht, TreeMap, wird die Architektur so erweitert, dass auf bestehende Komponenten zugegriffen werden kann. Dabei kommen drei Patterns zum Einsatz. Diese werden in diesem Kapitel beschrieben. Die Abbildung 22 zeigt auf, wie die Komponenten neu über das Fassaden Pattern genutzt werden.

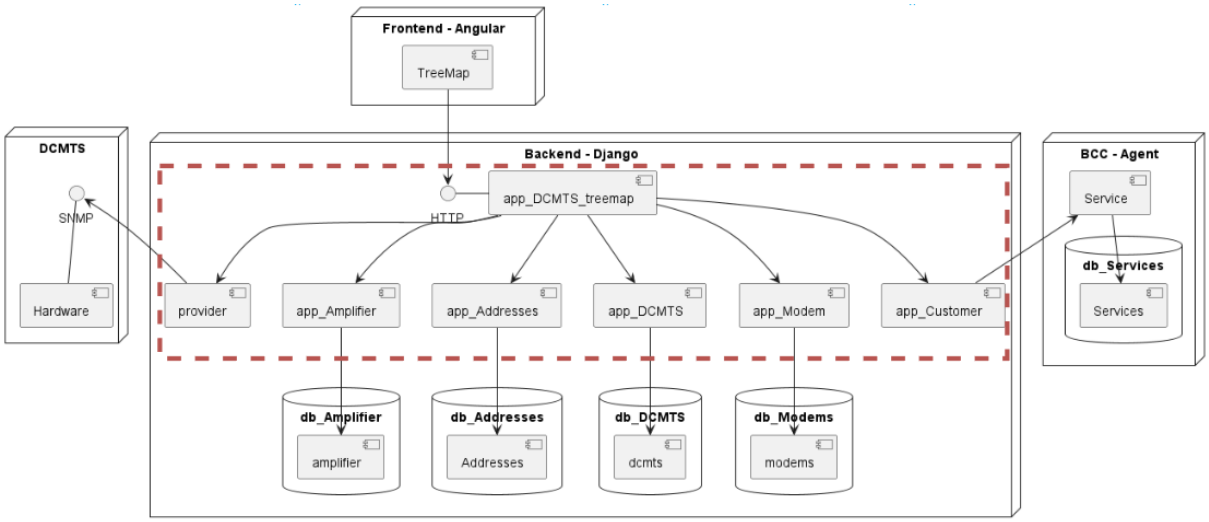


Abbildung 22 Komponentendiagramm Fassade

Quelle: Eigene Darstellung

5.10.2.1 Fassade

Unter Abbildung 22 ist die Fassade, gestrichelt und rot markiert, die im Projekt zum Einsatz kommt. Durch das Einsetzten dieses Patterns, wurden die Abhängigkeiten reduziert, die vorher zwischen dem Frontend und dem Backend herrschten. Durch die Fassade wurde ausserdem die Kommunikation auf eine einzige Abfrage reduziert. Die Nutzer dieser View, brauchen somit nur einen Aufruf, um an die Daten zu gelangen. Der Vorteil dabei, falls die Nutzer über Remote auf die Ansicht zugreifen, sind sie weniger von einer permanenten Internetverbindung abhängig. Diese Verbesserung kommt nur der neuen Visualisierung TreeMap zugute, welche neu im Rahmen dieses Projektes erstellt wird.

5.10.2.2 DTO

Die Daten werden erst an das Frontend gesendet, wenn dies bereit sind. Diese werden, bevor sie zugestellt werden, als Datentransferobjekt aggregiert. Das Datentransferobjekt wurde so gewählt, dass sie das HF-Netz so detailgetreu wie möglich abbildet. Das ganze Datentransferobjekt ist unter Abbildung 23 ersichtlich. Die Attribute mit einem Fragezeichen dahinter deuten darauf hin, dass diese Informationen eventuell nicht existieren, da dieser Knoten in der Realität nicht existiert. Die Interfaces, aus denen das DTO zusammengesetzt wurde, werden hier noch erläutert.

DCMTSNetworkTreeMap:
Unter diesem Interface werden die Informationen des DCMTS definiert. Dazu gehört unter anderem die Adresse, die Koordinaten, die angehängten Liegenschaften und weiteren Knoten. Die beiden letzteren sind als Listen gespeichert.

AmplifierNetworkTreeMap
Diese Schnittstelle repräsentiert die einzelnen Verstärker. Dieses Interface ist ähnlich zum DCMTS. Auch hier werden die weiteren Verstärker oder Liegenschaften in Listen gespeichert.

AddressNetworkTreeMap
Das hier genannte Interface enthält die Informationen der Liegenschaften. Eine Besonderheit, was dieses Interface ausmacht, ist es, dass nur hier die Liste mit den Modems angehängt werden kann.

ModemOnTreeMap
Dieses Interface enthält die Informationen des Modems. Anhand dieser Informationen können die einzelnen Positionen gefärbt werden.

```
export interface DCMTSNetworkTreeMap {
  id: number;
  number: string;
  name: string;
  latitude: string;
  longitude: string;
  address_dcmts?: AddressNetworkTreeMap[];
  amplifiers?: AmplifierNetworkTreeMap[];
}

export interface AmplifierNetworkTreeMap {
  id: number;
  number: string;
  name: string;
  latitude: string;
  longitude: string;
  amplifiers?: AmplifierNetworkTreeMap[];
  address_amplifier?: AddressNetworkTreeMap[];
}

export interface AddressNetworkTreeMap {
  id: number;
  name: string;
  number: string;
  latitude: string;
  longitude: string;
  zip: number;
  modems?: ModemOnTreeMap[];
}

export interface  ModemOnTreeMap {
  id: string;
  address: string;
  address_number: number;
  address_add: string;
  ip: string;
  mac: string;
  status: string;
  rx_power: number;
  us01: number;
  ds_uncorrected: number;
  ds01: number;
  us_snr: number;
}
```

Abbildung 23 DTO

Quelle: Eigene Darstellung

5.10.2.3 Adapter

Um die Daten im Frontend zu nutzen, wurde das Datentransferobjekt vorher mit einem Adapter, siehe 2.6.2, ergänzt. Das vereinfacht einerseits die Struktur der Daten und andererseits, koppelt der Adapter diese Struktur an die verwendete Visualisierung. Der Vorteil, falls die Entscheidung zukünftig auf eine andere Visualisierung fällt, kann diese durch einen weiteren Adapter realisiert werden. Somit bleiben die Beiden Klassen, die des DTO und die der Visualisierung unverändert. Unter Abbildung 24 wird der Unterschied zum eigentlichen Datentransferobjekt sichtbar. Die Daten vereinfacht und nur diese, die für die Visualisierung benötigt werden wurden verarbeitet. Alle anderen wurden weggelassen.


```
{
    "name": "01 Sonnegg",
    "children": [
        {
            "name": "Alte Kappelerstrasse 49",
            "children": []
        },
        {
            "name": "Alte Kappelerstrasse 61",
            "children": [
                {
                    "mac": "B42A0E90E589",
                    "name": "Alte Kappelerstrasse 61",
                    "ds01": 61.4,
                    "ip": "10.101.7.183",
                    "ds_uncorrected": 115758,
                    "us_snr": 352,
                    "rx_power": 0,
                    "us01": 104.2
                }
            ]
        },
    }
]
```


Abbildung 24 Adapter

Quelle: Eigene Darstellung

5.11 Sequenzdiagramme backend

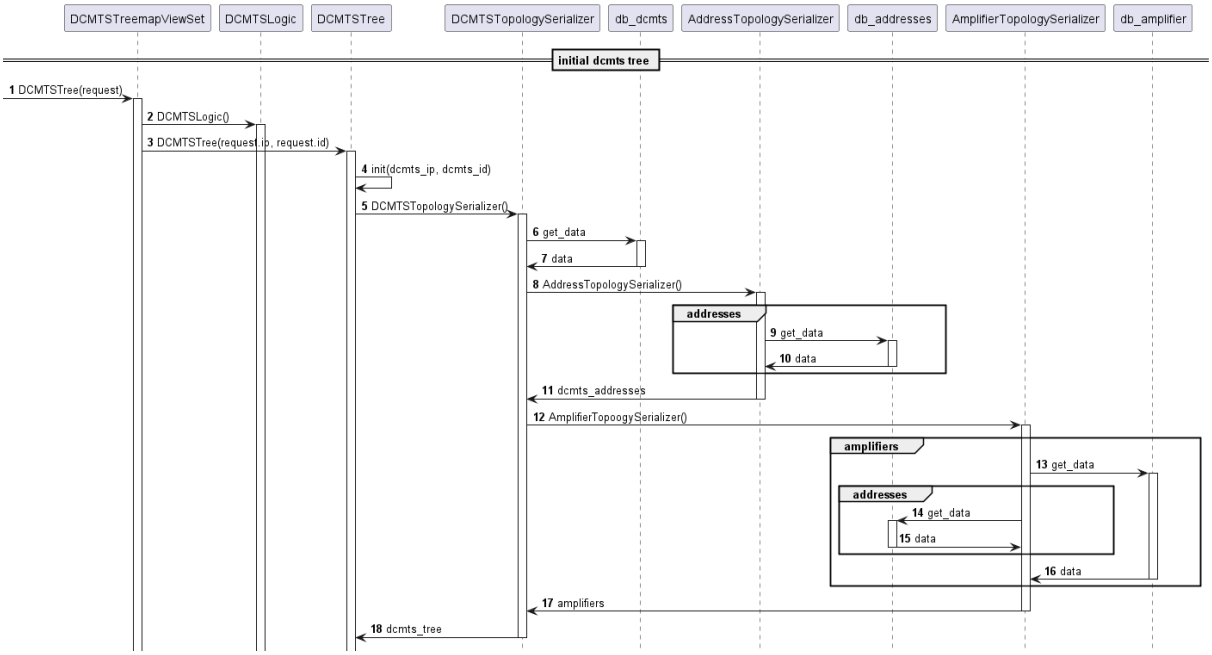


Abbildung 25 Sequenzdiagramm - DTO initialisieren

Quellen: Eigene Darstellung

In diesem Kapitel wird, unter Berücksichtigung vom Kapitel 4.6.3, auf den Ablauf im Backend eingegangen. In Abbildung 25 wird aufgezeigt, wie des Datentransferobjekt initialisiert wird. Die Abbildung 26 zeigt hingegen auf, wie dieses anschliessend mit den Daten aus den verschiedenen Bereichen aggregiert wird. Die Daten stammen, je nach Schnittstelle, aus Datenbanken, anderen Systemen oder werden aus den Geräten selbst per SNMP-Protokoll ausgelesen. Die einzelnen Aufrufe und deren Bedeutung werden in diesem Kapitel erklärt.

5.11.1 Initialisierung des DTO

1- Vom Frontend kommt eine GET-Anfrage, das Datentransferobjekt zu erstellen. Die ID und IP des gewünschten DCMTS müssen in der Anfrage mit gesendet werden.
2- Die Fassade, die die Logik enthält, DCMTSLogic genannt, wird initialisiert.
3- Zugriff auf die DCMTS Datenbank. Es wird mit der IP und der ID des DCMST versucht, auf das gewünschte DCMTS zuzugreifen.
4- Das Datentransferobjekt wird mit den Informationen IP und ID initialisiert.
5- Um die richtige Reihenfolge der Daten zu gewährleisten, werden diese serialisiert. Dafür wird der DCMTSTopologySerializer initialisiert.
6- Anfrage an die DCMTS Datenbank, dem Wurzelknoten.
7- Die Daten von der Angefragten DCMTS Datenbank die zurückgesendet werden.
8- Es wird ein Objekt, AddressTopologySerializer benötigt, was die ankommenden Daten serialisiert. Es handelt sich dabei um die Liegenschaften, die in einer grossen Anzahl folgen können.
9- Die db_addresses Datenbank wird angefragt, alle Liegenschaften, die am Wurzelknoten DCMTS hängen, zurückzusenden.
10- Alle Liegenschaften, welche am Wurzelknoten hängen, werden von der db_addresses zurückgesendet.
11- Die Daten, die die Liegenschaften betreffen, werden serialisiert.

12- Als nächstes wird ein Objekt generiert, der AmplifierTopologySerializer, der für die Serialisierung der Verstärker zuständig ist. Die Verstärker repräsentieren die weiteren Knoten.
13- Alle Verstärker, die an diesem DCMTS hängen, werden einzeln aus der Verstärkerdatenbank db_amplifier gelesen.
14- Pro Verstärker werden die Liegenschaften aus der Datenbank db_addresses gelesen.
15- Die Liegenschaften die aus der db_adresses Datenbank gefunden wurden werden von der Datenbank zurückgesendet.
16- Die Verstärker, inklusive angehängten Liegenschaften, werden zurückgesendet.
17- Die Verstärker und Liegenschaften werden serialisiert.
18- Die komplette Topologie aus Wurzel, Knoten und Liegenschaften wird serialisiert und dem Datentransferobjekt übergeben.

5.11.2 Modems an DTO anhängen

19- Der Wurzelknoten, das DCMTS, wird hier über das SNMP angefragt. Es werden alle Modems angefragt, die an diesem Gerät angehängt sind.
20- Die Modems, die am DCMTS hängen, werden zurückgesendet.

5.11.3 Modems mit Informationen erweitern

21- Die Modems werden an die Fassade, DCMTSLogic, übergeben.
22- Es werden Threads erstellt, die pro Modem aufgeteilt werden.
23- Die bestehenden Daten der Modems mit den erweiterten Messwerten vereinen.
24- Hier werden erweiterten Daten über SNMP von den Modems ausgelesen.
25- Die Antwort der Modems, hier die Daten mit den Metriken, werden von den einzelnen Geräten geliefert.
26- Das Datentransferobjekt wird mit den erweiterten Informationen ergänzt.

5.11.4 Modems mit Adressen erweitern

27- Das Datenobjekt sendet die Modems an die Fassade.
28- Es werden Threads erstellt, die pro Modem aufgeteilt werden.
29- Die bestehenden Daten der Modems mit den erweiterten Liegenschaften vereinen.
30- Hier werden die kombinierten Daten anschliessend an ein Objekt gesendet, dem ServiceSerializer, der diese serialisiert.
31- Die Modems, die serialisiert wurden, werden zurückgesendet.
32- Das Datentransferobjekt wird mit den neuen Informationen Modems aktualisiert.

5.11.5 Modem am DCMTS

33- Das Datentransferobjekt sendet die zu prüfenden Liegenschaften, welche am Wurzelknoten hängen, an die Fassade DCMTSLogic.
34- Die Fassade prüft für alle Liegenschaften am Wurzelknoten, ob sich Modems an diesen befinden. Falls ja, werden diese Modems an diesen Liegenschaften angehängt.
35- Die erweiterten Liegenschaften werden dem Datentransferobjekt übergeben.

5.11.6 Modem am Verstärker

36- Das Datentransferobjekt sendet die zu prüfenden Verstärker, welche am DCMTS hängen, an die Fassade DCMTSLogic.
37- Die Fassade prüft für jede Liegenschaft am Verstärker, ob sich Modems da befinden. Falls ja, werden diese Modems an diesen Liegenschaften angehängt.
38- Die Liegenschaften mit den erweiterten Informationen der Modems werden dem Datentransferobjekt übergeben.
39- Datentransferobjekt dem Frontend zustellen.

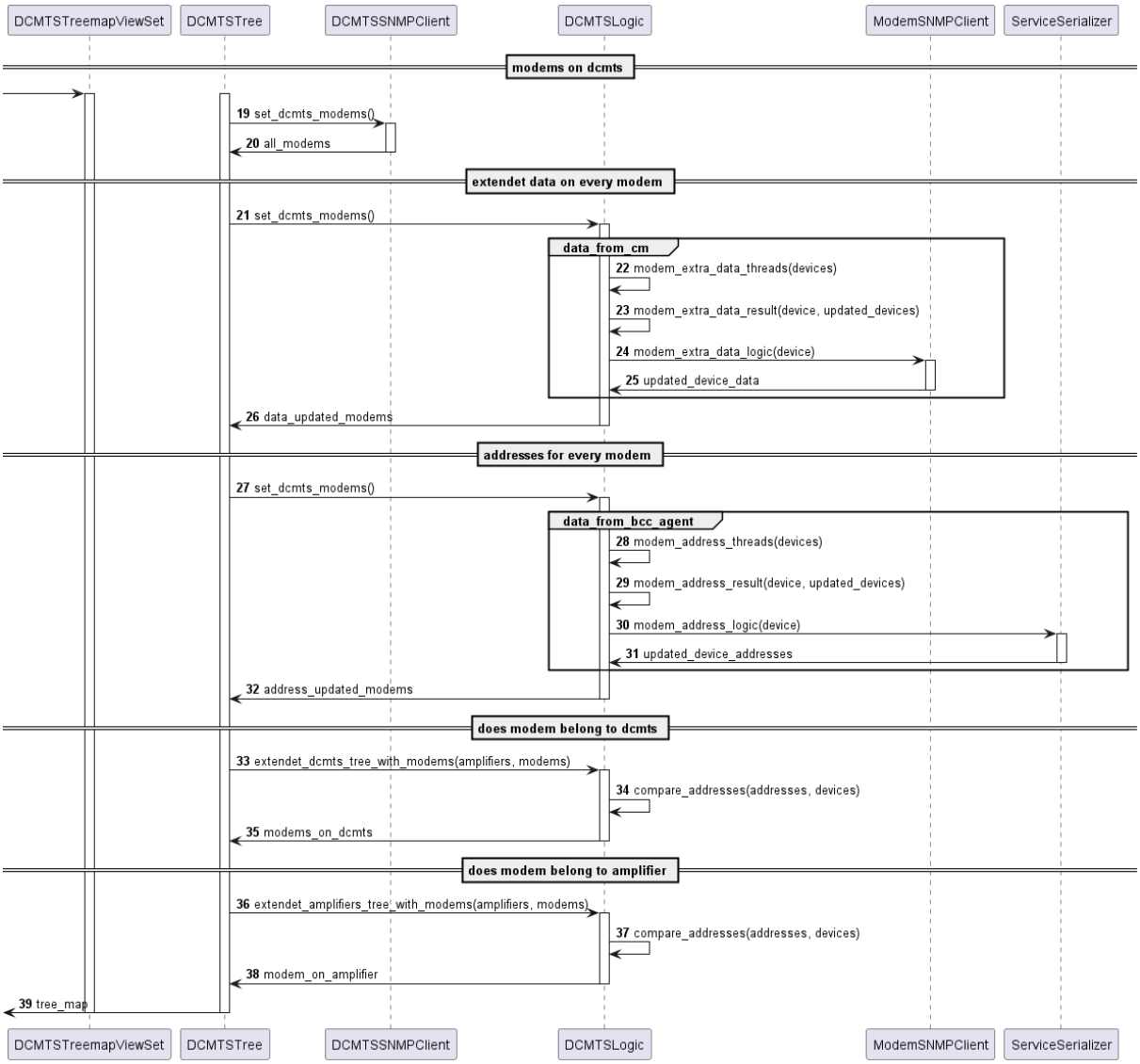


Abbildung 26 Sequenzdiagramm - DTO erweitern

Quelle: Eigene Darstellung

5.12 Parallelisierung backend

Um das Backend performanter zu gestalten, wurden das Backend Parallelisiert. In Abbildung 26 ist diese Parallelisierung ersichtlich. Der sich in den den Schleifen befindete Teil data_from_cm und und data_from_bcc_agent wurde paralellisiert. Für einen mittelgrossen Baum mit 16Verstärkern und 78 Modmes wurde die Performenz so drastisch gesteigert. Ohne parallelisierung dauerte die Generierung des Datentransferobjektes, gerundet, 17 Sekunden. Mit der Parallelisierung, war das Datentransferobjekt nach knappen 9 Sekunden erstellt.

5.12.1 Probleme der Parallelisierung

Die Parallelisierung ist für die beiden Abschnitte, unter den Schritten 33 und 36 in Abbildung 26 auch vorgesehen. Das Problem, zwei Listen parallel miteinander zu vergleichen, konnte leider nicht gelöst werden und wird zu einem späteren Zeitpunkt in Angriff genommen. Die Probleme traten bedingt auf und riefen komische Anomalien hervor. Das Ganze Datentransferobjekt stimmte nicht mehr überein, was der ausschlaggebende Grund ist, diese beiden offenen Punkte vorübergehend fallen zu lassen.

5.13 Einführung Frontend

In diesem Kapitel wird auf das Frontend näher eingegangen. Das Frontend ist mit dem Framework Angular und der Programmiersprache TypScript realisiert. Es wird im Rahmen dieses Projektes erweitert. Das Frontend wird, wie in Tabelle 15 vorgegeben, realisiert.

5.14 Bestandesaufnahme Frontend

Das Projekt besteht aus verschiedenen Komponenten. Diese sind nach Domänen unterteilt. Unter Abbildung 27 sind die verschiedenen Komponenten aufgeführt. Im Projekt bestehen die Komponenten, Angular typisch, aus drei Dateien. Die HTML-Datei, wo die Struktur und die Elemente festgelegt sind. Der CSS-Datei, wo das Design festgehalten wird und der TS-Datei, wo das Verhalten beschrieben wird. Das ist ein Merkmal von Angular, was für die Erweiterung berücksichtigt und auch übernommen wurde. Die Treemap-Komponente des Frontendes greift über eine GET-Schnittstelle auf das Backend zu. Dieses liefert danach das Datentransferobjekt, wie im Kapitel 5.11 ausführlich beschrieben. Ausserdem ist erkennbar, dass die Umstellung auf das Datentransferobjekt, der Weiterentwicklung des Frontendes zugutekommt. Es ist hat sich erübrigt, mehrere Schnittstellen oder die Reihenfolge dieser zu berücksichtigen, da die Anfrage auf eine reduziert wurde.

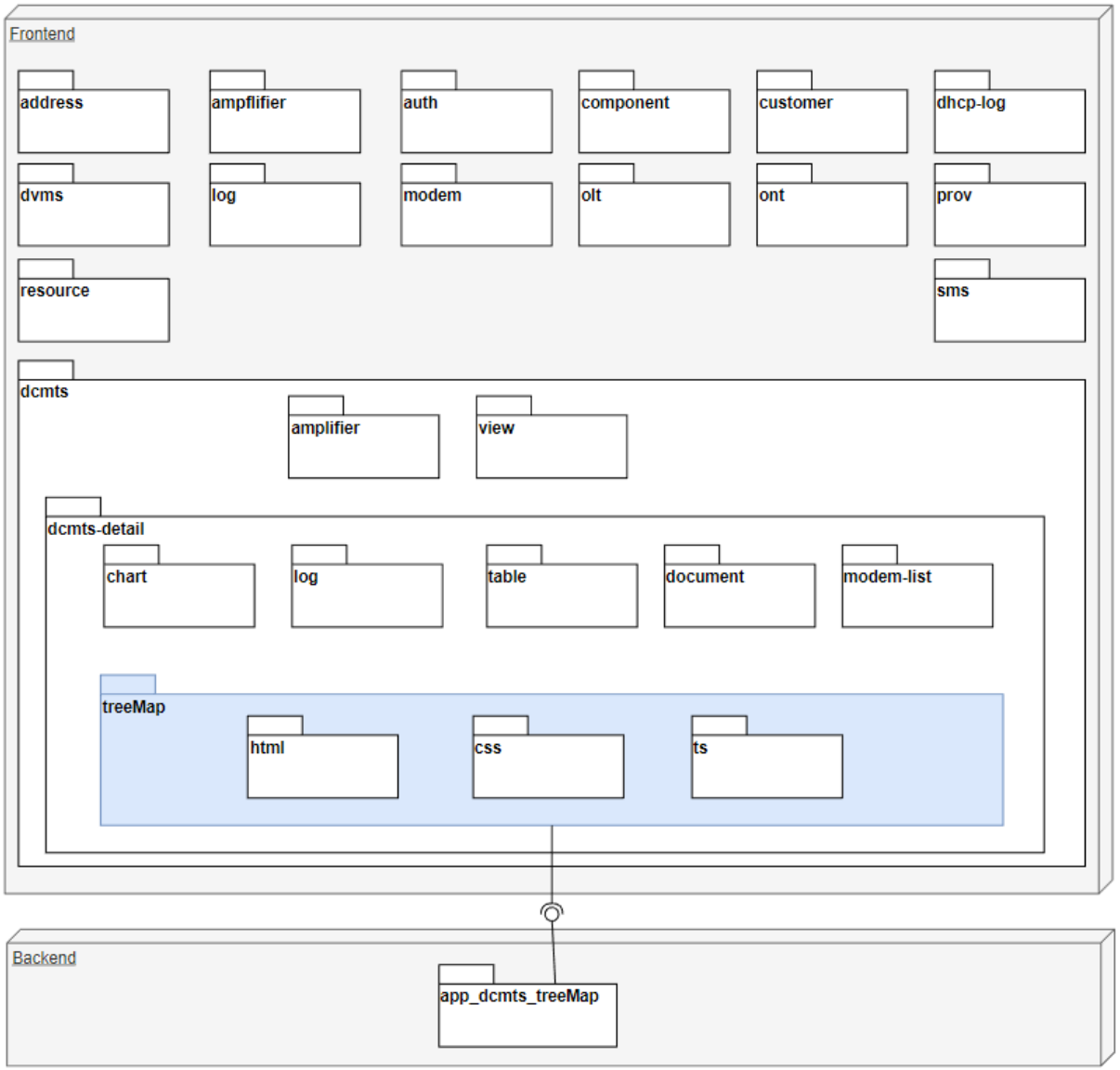


Abbildung 27 Komponentendiagramm – Frontend

Quelle: Eigene Darstellung

5.15 Implementation Frontend

Für das Frontend wurde, aufgrund der Nutzwertanalyse aus Kapitel 5.4.1, die Visualisierung als Tree gewählt. Diese wurde, aus Zeitgründen, vereinfacht und aus den Beispielen vom Framework d3 aufgebaut. Anschliessend wurden diese so angepasst, dass sie den Anforderungen des Projektes entsprachen. Der Vorteil, der sich daraus ergibt, liegt darin, dass andere Visualisierungen der Hierarchischen Varianten einfach zukünftig implementiert werden, da diese die gleiche Datenstruktur besitzen. Die Funktionsweise des Frontendes wird unter Abbildung 28 mit Blöcken abstrakt beschrieben.

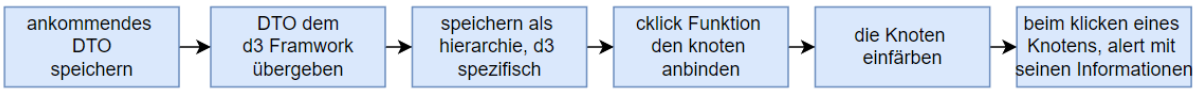


Abbildung 28 Funktionsweise Frontend

Quelle: Eigene Darstellung

5.16 Integrationstest – GUI

Für den Integrationstest, Test 01 im Kapitel 8.2 und die Beschreibung im Kapitel 2.7.1, des GUI wird das DCMTS mit der Nummer 10 ausgewählt. Diese Zone wird für die Performance Messung von einem Technikeschen Angestellten, siehe Abbildung 30, Manuel erfasst. In diesem Kapitel wird die Richtigkeit des neu geschaffenen Systems, mit derjenigen des Technischen Angestellten verglichen. Der Test zeigt auf, dass die beiden Graphen, von Abbildung 29 und Abbildung 30, identisch sind und übereinstimmen. Die Knoten, Liegenschaften und Modems werden korrekt abgebildet. Einzig die Modems, die währenddessen abgestellt worden sind und nicht mehr erreichbar waren. Die Überlegung, Modems die Offline sind, ebenfalls einzublenden wird nach Projektende untersucht werden.

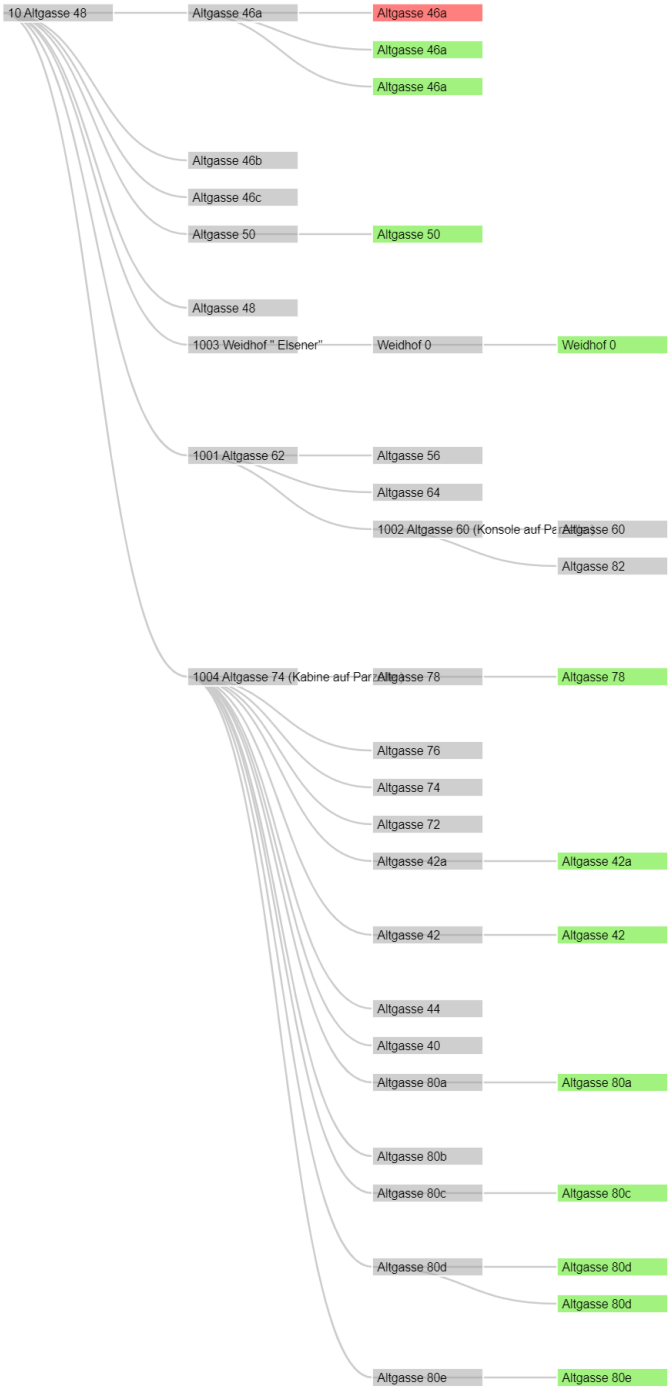


Abbildung 29 Node 10 automatisch vom System

Quelle: Eigene Darstellung

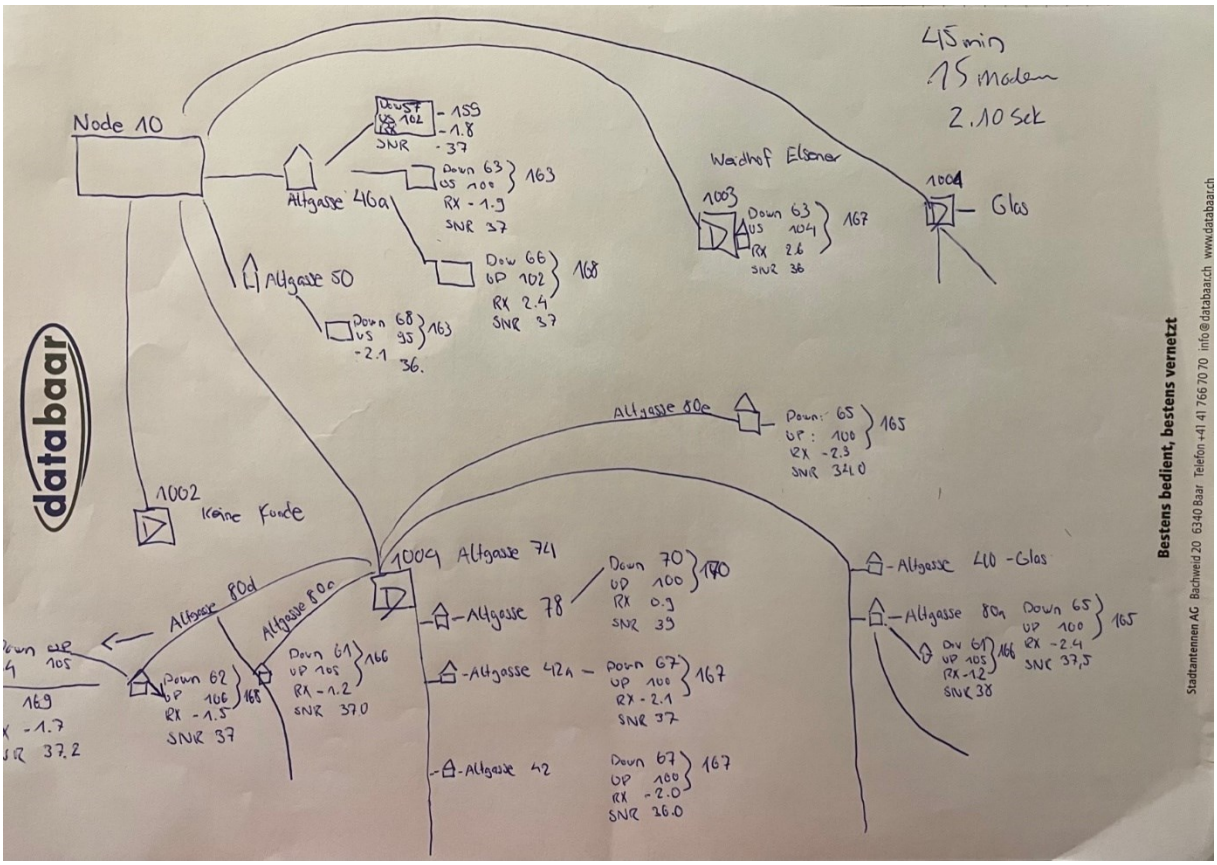


Abbildung 30 Node 10 manuell erfasst

Quelle: Eigene Darstellung

5.17 Funktionstest – GUI

In diesem Kapitel werden die einzelnen Funktionstests, siehe 2.7.2, beschrieben, die durchgeführt wurden.

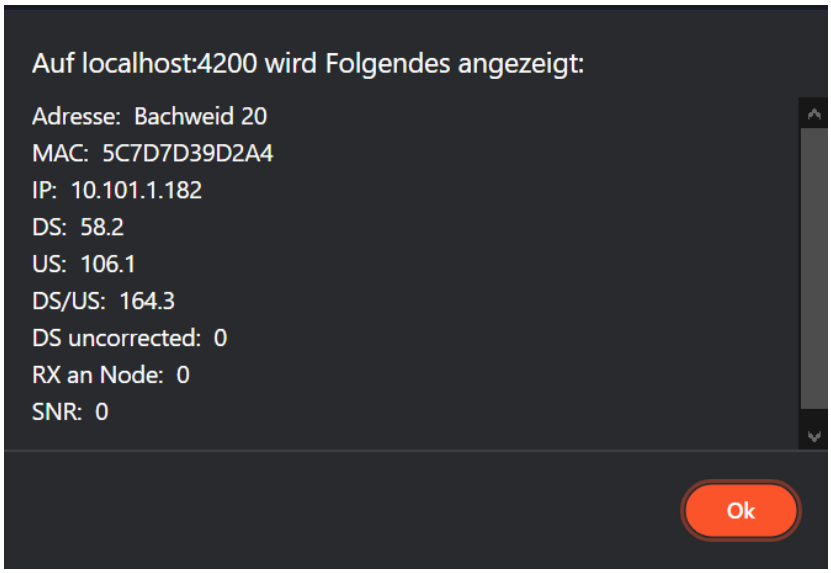


Abbildung 31 Referenzmodem

Eigene Darstellung

5.17.1 Test des GUI

In diesem Kapitel werden die Tests beschrieben, die die Darstellung der Visualisierung betreffen. Es wurden die Metriken des Referenzmodem, siehe Abbildung 31, getestet. Ein Test jeweils im Wertebereich und ein Test, jeweils ausserhalb dieses Wertebereichs. Falls die Metrik im GUI richtig dargestellt wird, wird die Metrik in Tabelle 8 grün, andernfalls rot eingefärbt. Die Auswertung sind in Tabelle 8 abgebildet. Der Tabelle ist zu entnehmen, dass die Tests erfolgreich waren. Für sonstige Ergebnisse wurde die Farbe Gelb eingesetzt. Bei diesen beiden Metriken, «RX and der Node» und «SNR» fiel auf, dass die Dezimalstelle falsch angegeben war. Dieser Fehler wird nach der Abgabe der Arbeit korrigiert.

Tabelle 8 Testresultate Funktionstest GUI

Metrik	innerhalb Wertebereich	ausserhalb Wertebereich
DS	Test 03	Test 04
US	Test 05	Test 06
DS/US	Test 07	Test 08
DS uncorrected	Test 09	Test 10
RX an Node	Test 11	Test 12
SNR	Test 13	Test 14

5.17.2 Test der Abfragen

In diesem Kapitel wird getestet, ob sich die Metriken des Modems ändern, wenn diese physisch vor Ort verändert werden. Diese Änderung fand durch einen sechs dB Dämpfer statt, der vor dem zu Messenden Modem angehängt wurde. Die Änderungen, die durch den Dämpfer sich hätten ereignen müssen, haben sich auch ereignet. Der Test verlief zufriedenstellend. Die in Gelb eingefärbten Test, deuten wieder auf die Fehler, die bereits in Kapitel 5.17.1 aufgetaucht sind.

Metrik	Änderung richtig
DS	Test 15
US	Test 16
DS/US	Test 17
DS uncorrected	Test 18
RX an Node	Test 19
SNR	Test 20

Tabelle 9 Testresultate Funktionstest Dämpfer am Modem

Quelle: Eigene Darstellung

Das Modem, was während des Tests verwendet wurde, ist weiter unten abgebildet. Die Werte, die sich mit dem Einsatz des sechs dB Dämpfers verändern, sind ersichtlich. Die Daten in Abbildung 32 entsprechen dem Modem ohne Dämpfer. Die Daten in Abbildung 33 entsprechen hingegen dem Modem, welches den sechs dB Dämpfer verbaut bekommen hat. Durch den Einsatz des Dämpfers, unterscheiden sich die die Werte der betroffenen Metriken um circa sechs dB, was genau der Höhe des Dämpfers entspricht.

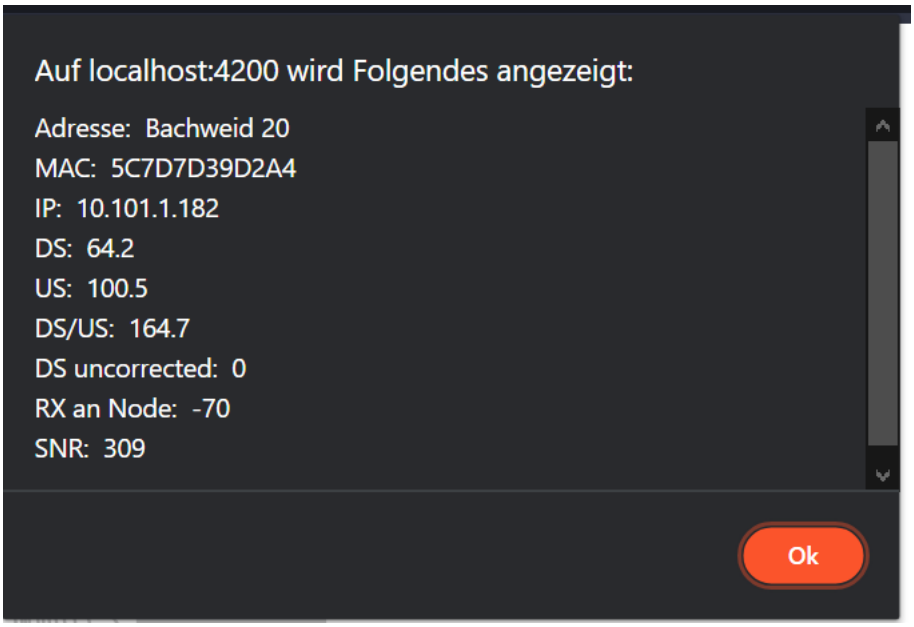


Abbildung 32 Modem ohne Dämpfer

Quelle: Eigene Darstellung

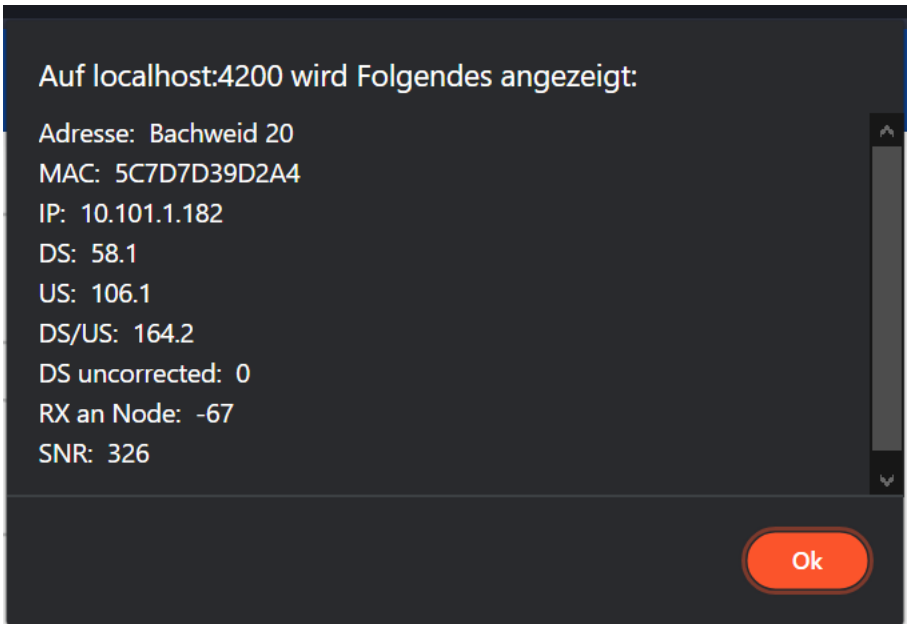


Abbildung 33 Modem mit Dämpfer

Quelle: Eigene Darstellung

5.18 Resultate

Unter diesem Kapitel werden die Resultate besprochen. Es wird auf das Frontend eingegangen und den Vorteil, den diese mitbringen.

5.18.1 Bestandesaufnahme Monitoring

Es wurden drei Technische Angestellte aufgeboten, jeweils eine komplette Zone auf deren Zustand zu prüfen. Aus Zeitgründen wurde die Aufgabe gekürzt auf die Bestandesaufnahme. Das Auswerten der verschiedenen Metriken wurde aus oben genanntem Grund weggelassen. Die verschiedenen Zonen sind im Anhang, in Kapitel 8.3, aufzufinden.

Der Ablauf war wie folgt:

1. Die Topologie des Koaxialnetzes aufzeichnen
2. Die Liegenschaften, die angehängt sind, aufzeichnen
3. Die Modems, die an diesen Liegenschaften hängen, aufzeichnen
4. Die Metriken dieser Modems notieren

Die Auswertung dieser Aufnahmen ist in Tabelle 10 ersichtlich. Laut dieser Tabelle, geht hervor, dass im Schnitt mindestens drei Minuten pro Modem nötig sind, um einen kompletten eine Zone manuell aufzunehmen. Es gilt zu berücksichtigen, dass es sich in dem Experiment um kleinere Zonen mit wenigen Modems handelte. Im Schnitt hängen zwischen 80 und 120 Modems an einer Zone. Das hätte laut der unteren Schätzung von drei Minuten pro Modem, zwischen 240 und 360 Minuten in Anspruch genommen. Diese Zeit steht den Technischen Angestellten nicht zur Verfügung.

Tabelle 10 Auswertung Performance der Techniker

Techniker	Node	Anzahl Modems	Zeit (min)	Zeit/Modem
Jurian	48 Grundstrasse	17	43	2.529411765
Leo	03 Erlenhof	15	60	4
Jasin	10 Altgasse	15	45	3
		Total (min)pro Modem im Schnitt:		3.176470588

Die Gleichen Zonen wurden mit der neu geschaffenen Komponente aufgenommen. Die neu benötigten Zeiten sind unter Tabelle 11 ersichtlich. So hat sich die durchschnittlich benötigte Zeit pro Modem von 3.16 Minuten auf 0.149 Sekunden verkürzt. Eine Verbesserung der Performance um 99.92%.

Tabelle 11 Auswertung Performance der Komponente

Techniker	Node	Anzahl Modems	Zeit (sek)	Zeit/Modem
Jurian	48 Grundstrasse	17	2.59	0.152352941
Leo	03 Erlenhof	15	2.34	0.156
Jasin	10 Altgasse	15	2.1	0.14
		Total (sek) pro Modem im Schnitt:		0.14945098

5.18.2 Das GUI

In diesem Kapitel werden die Elemente des GUI, mit Hilfe der Abbildung 34, beschrieben.

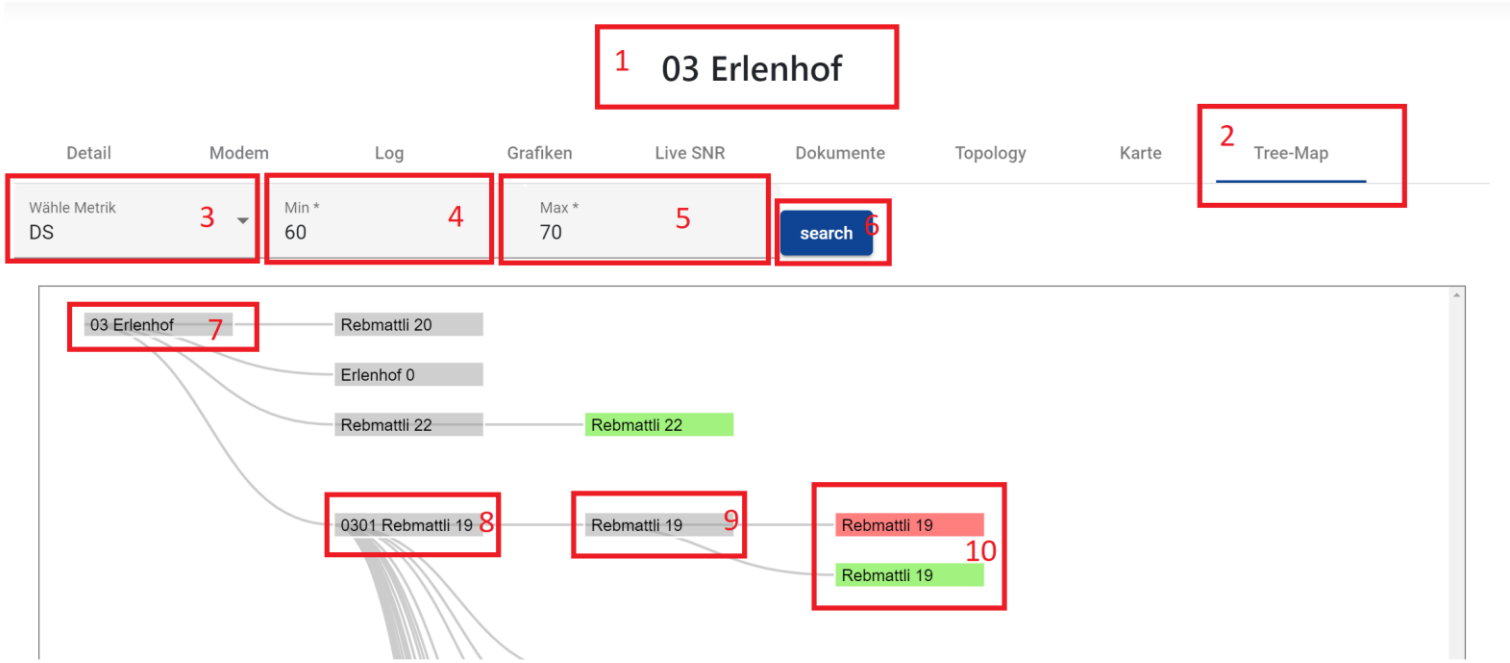


Abbildung 34 GUI

Quelle Eigene Darstellung

Unter Punkt 1 ist der Name des DCMTS oder auch Wurzelknotens eingeblendet. Durch Anwählen des Tree-Map Reiter, welches unter Punkt zwei markiert wurde, gelangt der Benutzer zu der eigentlichen Visualisierung. Als nächstes, mit Punkt drei gekennzeichnet, wird die Metrik gewählt. Danach kann, unter Punkt vier der untere und unter Punkt fünf der obere, Wertebereich definiert werden. Der unter Punkt sechs markierte Knopf initiiert anschliessend die eigentliche Darstellung der Grafik. Der Punkt sieben repräsentiert den Wurzelknoten, in diesem Fall Wurzelknoten 03 Erlenhof. Anschliessend wird der Baum gezeichnet. Unter Punkt acht ist ein weiterer Knoten, ein Verstärker, grau abgebildet. Unter Punkt neun ist eine Liegenschaft, ebenfalls grau, dargestellt. Der Unterschied liegt darin, dass Verstärker eine Nummerierung besitzen, die vor dem Namen angehängt wird. Rot oder Grün, je nachdem ob der gewählte Wert im gewählten Wertebereich liegt, sind die Modems, die diese Voraussetzungen erfüllen oder eben nicht erfüllen. Durch einen Mausklick auf dem Modem, öffnet sich eine Alert Meldung mit den einsehbaren Metriken. Eine solche Alert Meldung ist unter Abbildung 32 ersichtlich.

6 Evaluation und Validation

Dieser Abschnitt dient der Reflexion, des durchgeführten Projektes.

6.1 Erreichen und Erfüllen der Anforderungen

In diesem Kapitel werden die Anforderungen auf ihre Erfüllung überprüft. Falls die Kriterien erreicht wurden, wurde der Typ grün eingefärbt. Falls die Anforderung nicht erreicht wurde, wurde der Typ gelb eingefärbt. Der Buchstabe M in der Spalte «Typ» weist auf eine Anforderung hin, die erfüllt werden muss. Der Buchstabe K hingegen auf ein Kriterium, was erfüllt werden kann. Unter der Spalte «Beschreibung» wird der Status der Anforderung kommentiert. Im Anhang befindet sich unter Kapitel 8.1, der Anforderungskatalog.

6.1.1 Technischen Anforderungen

Tabelle 12 Erfüllte Technische Anforderung

Punkt	Kriterien	Typ	Beschreibung
1	**Technische Anforderungen**		
1.1	**Allgemeine Funktionalität**		
1.1.a	Mehrbenutzerfähig	M	Das bestehende Projekt wurde erweitert
1.2	**Betriebssystem**		
1.2.a	Linux	M	bestehende Umgebung verwendet
1.3	**Datenbank**		
1.3.a	MySQL (MariaDB)	M	Daten wurden im Backend aggregiert, keine Veränderung des Backend
1.4	**Schnittstellen**		
1.4.a	Backend Watchdog	M	Die Schnittstelle wird vom Backend zur Verfügung gestellt
1.5	**Framework**		
1.5.a	Angular (Frontend)	M	Das Frontend wurde mit Angular umgesetzt.
1.5.b	Django (Backend)	M	Das Backend wurde mit Django umgesetzt.
1.5.c	D3 (Visualisierung)	K	Die Visualisierung konnte mit dem Framework von d3 umgesetzt werden.

6.1.2 Funktionale Anforderungen

Tabelle 13 Erfüllte Funktionale Anforderungen

Punkt	Kriterien	Typ	Beschreibung
2	**Funktionale Anforderungen**		
2.1	**Techniker**		
2.1.a	Netzzustand - aktuellen eines DCMTS darstellen	M	Der Zustand des Koaxialnetztes kann durch die Visualisierung ermittelt werden
2.1.b	Netzzustand - justieren	K	Einstellungen können noch nicht darüber getätigt werden. Wird zu einem späteren Zeitpunkt umgesetzt.
2.1.c	Farbliche Darstellung	M	Farblich wird das ganze unterschieden.
2.1.d	Metriken wählen	M	Es können verschiedene Metriken gewählt werden, die angezeigt werden.
2.1.e	Genauen Werte eines Modems auslesbar	M	Die genauen Metriken können durch ein Alert Fenster angezeigt werden.
2.2	**Teamleiter Technik**		
2.2.a	Überblick	M	Ein Überblick über die ganze Zone ist möglich.
2.2.b	Kontrolle	M	Es ist noch unübersichtlich. Default Werte wären wünschenswert. Somit müssen die Wertebereiche der einzelnen Metriken nicht jedes Mal eingestellt werden.
2.2.c	Planung	M	Diese kann mittels der Visualisierung erfolgen.
2.3	**Technischer Leiter**		
2.3.a	Kontrolle Netzzustand	K	Braucht Default Visualisierungen. Werden in nach der Abgabe erstellt.

6.1.3 Allgemeine Anforderungen

Tabelle 14 Erfüllung Allgemeine Anforderungen

Punkt	Kriterien	Typ	Beschreibung
3	**Allgemeine Anforderungen**		
3.1	**Unternehmen**		
3.1.a	Schulung	M	Wurde durchgeführt und den Technikern erklärt.
3.1.b	Visualisierung im Watchdog implementieren	M	Die Visualisierung ist im Watchdog implementiert. Jedoch erst auf der Testumgebung.
3.1.c	Vorgehensmethode Wasserfall & Agile	M	Nicht immer konsequent eingehalten. Gründe Knowhow der Frameworks, WM Qatar, keine Doku
3.1.d	Eingesetzten Technologien sind besprochen	M	Die Technologien wurden wie besprochen eingesetzt. Es wurde jedoch nicht separat ein Projekt erstellt, da viele Abhängigkeiten vorhanden waren.

3.2	**Hochschule**		
3.2.a	Unterschiedliche Lösungen evaluiert werden	M	Wurde durchgeführt.
3.2.b	Abgabetermin (03.01.2023)	M	Wurde eingehalten
3.2.c	Zwischenpräsentation	M	Doku war nicht so weit. Verschiedene Faktoren.
3.2.d	Endpräsentation	M	Wird eingehalten
3.2.e	Arbeitsaufwand (360h)	M	Wurde eingehalten.
3.2.f	Regelmässige Review mit Arbeitgeber	M	Viele Supportfälle aufgrund der WM und externen Systemen (Amino, IPTV) und Services. Schulung neuer Angestellten.
3.2.g	Testkonzept	M	Wurde erstellt.
3.2.h	Webabstract	M	Wurde erstellt.
3.2.i	Pitch Video	M	Wurde erstellt.
3.2.j	Arbeitsrapport	M	Wurde erstellt

6.2 Reflexion

Ich bin zufrieden mit dem Endprodukt. Die Anforderungen des Arbeitgebers konnten erfolgreich umgesetzt werden. Die Visualisierung des Koaxial Netzes funktioniert und weist bereits auf fehlerhafte Liegenschaften oder Positionen hin. Die Visualisierung ist intuitiv und es braucht nicht viel, diese den Nutzern näher zu bringen. Es ist auch eine grosse Hilfe, um neuen Angestellten oder einer Branchenfremden Person, das HF-Netz näher zu bringen.

Das Projekt stiess von Anfang an auf grosse Hindernisse. So wurden aufgrund der Fussball WM die Services des Unternehmens besonders in Anspruch genommen. Dies hatte zur Folge, dass die Supportabteilung mit Aufträgen zugeschüttet wurde und wenig Zeit übrigblieb, um an diesem Projekt zu wirken. Ein weiterer Grund, der sich auch negativ bemerkbar machte, war es, dass keine Dokumentation des Monitoring Systems vorhanden war. Das hatte zur Folge, dass viel Reverse Engineering betrieben wurde. In den Verwendeten Frameworks und Programmiersprachen wies ich persönlich keine nennenswerte Erfahrung und Kenntnisse auf, was auch vermehrt gegen eine Erfolgreiche Durchführung des Projektes sprach. Des Weiteren fiel Herrn Yilmaz, krankheitshalber, länger aus. Als während dieser Zeit noch zwei neue Angestellte ins Team kamen, die eingeführt werden mussten, kamen Zweifel auf.

So kam es, dass ich unter diesen Voraussetzungen, leider Abstriche machen musste. Ich hinkte mit der Doku hinterher, was sich am Ende mit einem hohen Aufwand mehr als nur rächte. Die Projektmanagement Methoden wurden zwar geplant, bei Durchführung wiegten die im oberen Abschnitt erwähnten Probleme jedoch zum Teil höher.

Die angewendeten Methoden, vor allem die Nutzwertanalyse hat mir sehr gefallen. Es half mir rational die Varianten der Grafiken oder die der Software Tools zu betrachten. Diese Methode werde ich bestimmt weiter verwenden bei solchen Entscheidungen.

Alles in allem war das Projekt ein Erfolg. Es war schön zu sehen, wie die Arbeitskollegen bei der Präsentation des Projektes gefallen dran fanden, da sie so, zukünftig viel speditiver ihre Arbeit verrichten konnten und nicht lange draussen herumirren werden bei der Fehlersuche.

7 Ausblick

Die neu geschaffenen Visualisierungen dienen mehreren Angestellten und können in verschiedenen Bereichen nützlich sein. Einerseits können die Techniker diese verwenden, um die Störungen im Netz früher ausfindig zu machen, um diese so speditiver zu lösen. Zur Illustration, in Abbildung 35 fällt früh auf, dass die Kunden an der Adresse Deinikonerstrasse 16, kein sauberes Signal beziehen. In diesem Fall könnte die Liegenschaft, anstelle einer aufwendigen Reparatur, auf die neue Technologie Glasfaser umgebaut werden. Aus Sicht des Teamleiters helfen die Visualisierungen, Lösungskonzepte für die Zonen zu erstellen. Das kann bis hin zu der Änderung der Architektur des HF-Netzes gehen. Aus Sicht des Technischen Leiters, können Stichproben vom aktuellen HF-Netz Zustand entnommen werden. Ausserdem sind die Visualisierungen, bei Themen, welche das Koaxialnetz betreffen, als Diskussionsgrundlage gut geeignet.

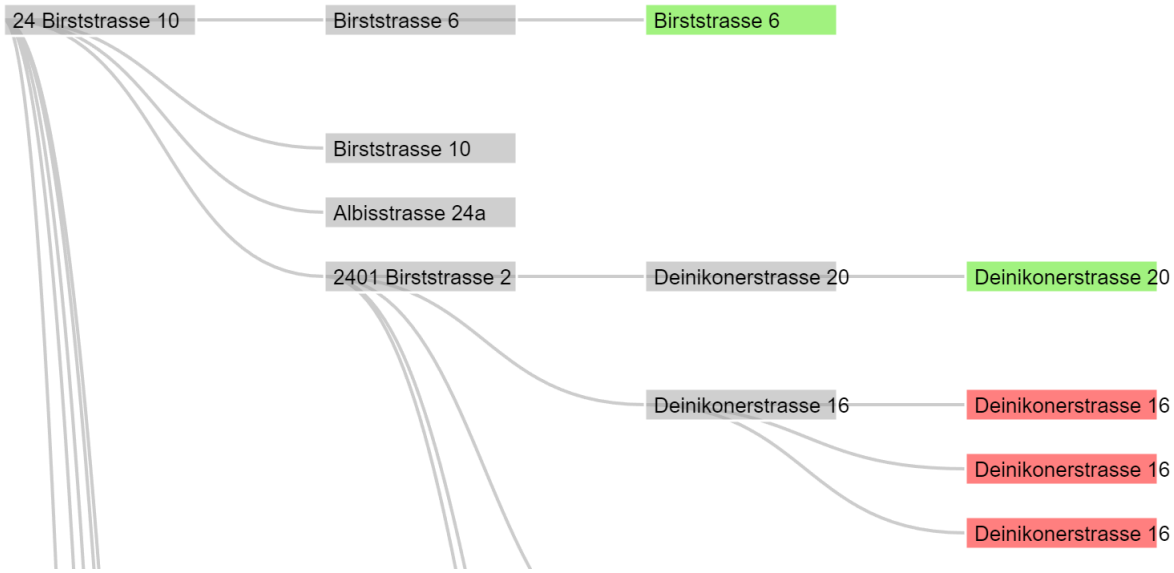


Abbildung 35 Ausblick

Quelle: Eigene Darstellung

8 Anhänge

Unter diesem Kapitel sind die Anhänge des Projektes aufzufinden.

8.1 Anforderungskatalog

In diesem Kapitel sind die Anforderungen dokumentiert. Diese wurden in drei Hauptgruppen unterteilt. Die erste Kategorie, Tabelle 15 Technische Anforderungen beschreibt die Technologien, welche der Arbeitgeber voraussetzt und welche zum Einsatz kommen. Unter Funktionale Anforderungen, sieh
Tabelle 16 Funktionale Anforderungen, wird auf die Anforderungen des Verhaltens des Systems eingegangen. Im dritten Teil, siehe Tabelle 17 Allgemeine Anforderungen, werden die Anforderungen der Hochschule und sonstige Anforderungen vom Arbeitgeber beschrieben.

Tabelle 15 Technische Anforderungen

Punkt	Kriterien	Typ	Beschreibung
1	**Technische Anforderungen**		
1.1	**Allgemeine Funktionalität**		
1.1.a	Mehrbenutzerfähig	M	
1.2	**Betriebssystem**		
1.2.a	Linux	M	
1.3	**Datenbank**		
1.3.a	MySQL (MariaDB)	M	
1.4	**Schnittstellen**		
1.4.a	Backend Watchdog	M	
1.5	**Framework**		
1.5.a	Angular (Frontend)	M	
1.5.b	Django (Backend)	M	
1.5.c	D3 (Visualisierung)	K	

Tabelle 16 Funktionale Anforderungen

Punkt	Kriterien	Typ	Beschreibung
2	**Funktionale Anforderungen**		
2.1	**Techniker**		
2.1.a	Netzzustand - aktuellen eines DCMTS darstellen	M	
2.1.b	Netzzustand - justieren	K	
2.1.c	Farbliche Darstellung	M	
2.1.d	Metriken wählen	M	
2.1.e	Genauen Werte eines Modems auslesbar	M	
2.2	**Teamleiter Technik**		
2.2.a	Überblick	M	
2.2.b	Kontrolle	M	
2.2.c	Planung	M	
2.3	**Technischer Leiter**		
2.3.a	Kontrolle Netzzustand	K	

Tabelle 17 Allgemeine Anforderungen

Punkt	Kriterien	Typ	Beschreibung
3	**Allgemeine Anforderungen**		
3.1	**Unternehmen**		
3.1.a	Schulung	M	
3.1.b	Visualisierung im Watchdog implementieren	M	
3.1.c	Vorgehensmethode Wasserfall & Agile	M	
3.1.d	Eingesetzten Technologien sind besprochen	M	
3.2	**Hochschule**		
3.2.a	Unterschiedliche Lösungen evaluiert werden	M	
3.2.b	Abgabetermin (03.01.2023)	M	
3.2.c	Zwischenpräsentation	M	
3.2.d	Endpräsentation	M	
3.2.e	Arbeitsaufwand (360h)	M	
3.2.f	Regelmässige Review mit Arbeitgeber	M	
3.2.g	Testkonzept	M	
3.2.h	Webabstract	M	
3.2.i	Pitch Video	M	
3.2.j	Arbeitsrapport	M	

8.2 Testplan

In diesem Kapitel ist der Testplan aufgelistet.

ID	01
Name	Auflisten des DTO
Testfall-Beschreibung	Das Datentransferobjekt (DTO) muss vom Backend zurückgesendet werden.
Testprozeduren	1. Backend starten (Entwicklungsumgebung) 2. Postman öffnen 3. GET-Anfrage an die URL mit den Informationen des DCMTS, ID und IP, senden
Erwartete Resultate	Der Ganze Baum wird aufgelistet. Der Baum muss Manuel kontrolliert werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden.

ID	02
Name	Alert Meldung mit Informationen über ein einzelnes Modem
Testfall-Beschreibung	Es sollen die Werte eines einzelnen Modems ausgelesen werden und in der Alert Meldung angezeigt werden.
Testprozeduren	1. Watchdog öffnen 2. DCMTS Ansicht öffnen 3. DCMTS Details anzeigen, an dem das Modem angehängt ist. 4. TreeMap Ansicht öffnen 5. Auf das Modem klicken
Erwartete Resultate	Die Alert Meldung geht auf und die einzelnen Metriken sind abgebildet. Aktuell sind es 9 Informationen.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das zu testende Modem muss online sein.

ID	03
Name	DS im Frontend liegt im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss grün angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	04
Name	DS im Frontend liegt NICHT im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich NICHT im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss rot angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	05
Name	US im Frontend liegt im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss grün angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	06
Name	US im Frontend liegt NICHT im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich NICHT im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss rot angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	07
Name	DS/US im Frontend liegt im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss grün angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	08
Name	DS/US im Frontend liegt NICHT im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich NICHT im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss rot angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	09
Name	DS uncorrected im Frontend liegt im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss grün angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	10
Name	DS uncorrected im Frontend liegt NICHT im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich NICHT im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss rot angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	11
Name	RX an Node im Frontend liegt im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss grün angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	12
Name	RX an Node im Frontend liegt NICHT im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich NICHT im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss rot angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	13
Name	SNR im Frontend liegt im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss grün angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	14
Name	SNR im Frontend liegt NICHT im Wertebereich
Testfall-Beschreibung	Es soll geprüft werden, ob die Farbe des Modems sich ändert im Frontend, wenn die Metrik sich NICHT im Wertebereich befindet.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen
Erwartete Resultate	Das Modem muss rot angezeigt werden.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein.

ID	15
Name	DS im Frontend, in der Alert Meldung verändert sich
Testfall-Beschreibung	Es soll geprüft werden, ob sich der Werte des Modems, welcher in der Alert Meldung angezeigt wird, sich verändert. Dieser wird mit einem Dämpfer verändert. Die Höhe der Dämpfung soll sich in der Signalstärke wieder spiegeln.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen 5. Modem öffnen (Alert Meldung) 6. Screenshot von den Metriken entnehmen 7. Dämpfer am Modem anhängen. 8. Punkt 1 bis 6 wiederholen. 9. Die beiden werte vergleichen
Erwartete Resultate	Die Metrik zwischen den beiden Zuständen muss um den Betrag sinken, den der Dämpfer trägt.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein. Der Dämpfer darf erst nach der ersten Messung angehängt sein. Die Visualisierung muss neu geladen werden.

ID	16
Name	US im Frontend, in der Alert Meldung verändert sich
Testfall-Beschreibung	Es soll geprüft werden, ob sich der Werte des Modems, welcher in der Alert Meldung angezeigt wird, sich verändert. Dieser wird mit einem Dämpfer verändert. Die Höhe der Dämpfung soll sich in der Signalstärke wieder spiegeln.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen 5. Modem öffnen (Alert Meldung) 6. Screenshot von den Metriken entnehmen 7. Dämpfer am Modem anhängen. 8. Punkt 1 bis 6 wiederholen. 9. Die beiden werte vergleichen
Erwartete Resultate	Die Metrik zwischen den beiden Zuständen muss sich um den Betrag erhöhen, den der Dämpfer trägt.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein. Der Dämpfer darf erst nach der ersten Messung angehängt sein. Die Visualisierung muss neu geladen werden.

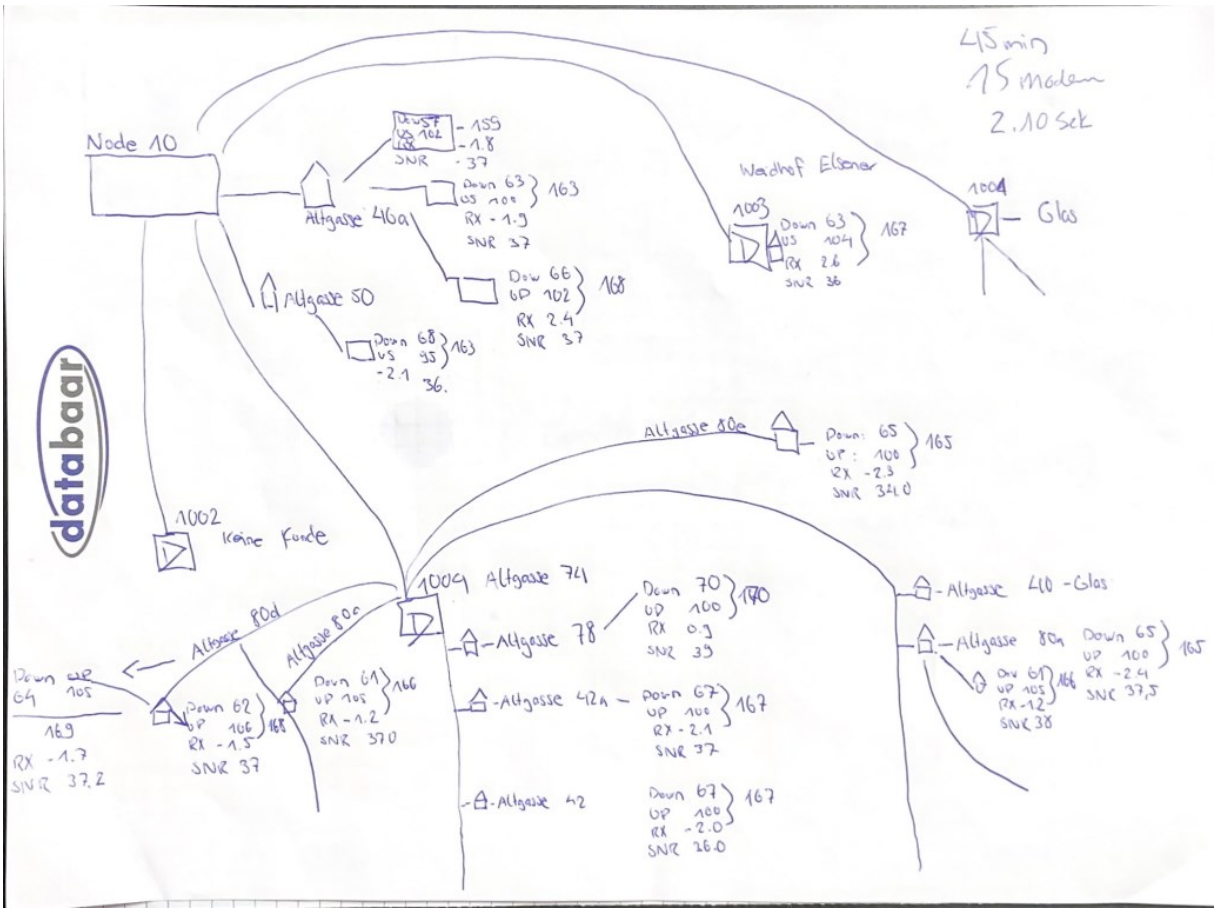
ID	17
Name	DS/US im Frontend, in der Alert Meldung verändert sich NICHT
Testfall-Beschreibung	Es soll geprüft werden, ob sich der Werte des Modems, welcher in der Alert Meldung angezeigt wird, sich verändert. Dieser wird mit einem Dämpfer verändert. Die Höhe der Dämpfung soll sich in der Signalstärke wieder spiegeln.
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen 5. Modem öffnen (Alert Meldung) 6. Screenshot von den Metriken entnehmen 7. Dämpfer am Modem anhängen. 8. Punkt 1 bis 6 wiederholen. 9. Die beiden werte vergleichen
Erwartete Resultate	Die Metrik zwischen den beiden Zuständen darf sich nicht verändern.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein. Der Dämpfer darf erst nach der ersten Messung angehängt sein. Die Visualisierung muss neu geladen werden.

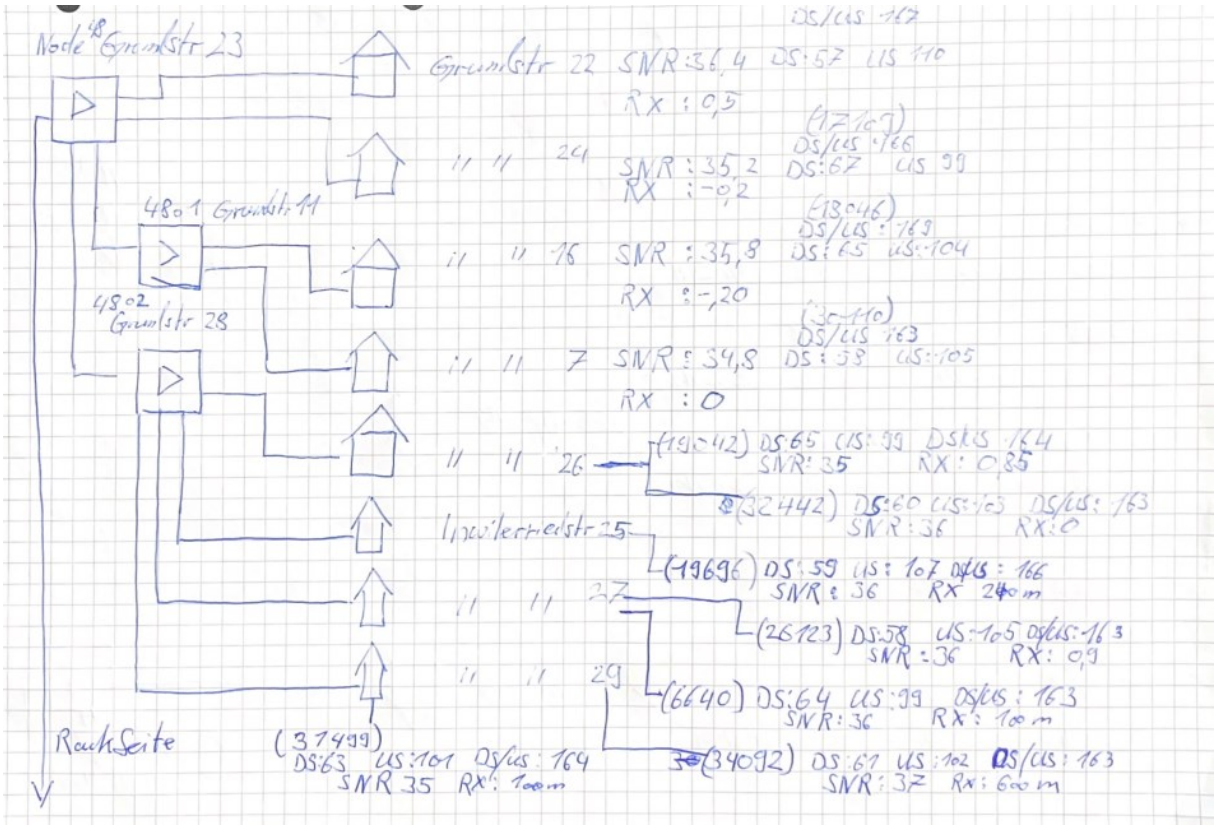
ID	18
Name	DS uncorrected im Frontend, in der Alert Meldung verändert sich NICHT
Testfall-Beschreibung	Es soll geprüft werden, ob sich der Werte des Modems, welcher in der Alert Meldung angezeigt wird, sich verändert. Einzelne Metriken werden verändert, die hier getestete Metrik darf sich nicht ändern!
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen 5. Modem öffnen (Alert Meldung) 6. Screenshot von den Metriken entnehmen 7. Dämpfer am Modem anhängen. 8. Punkt 1 bis 6 wiederholen. 9. Die beiden werte vergleichen
Erwartete Resultate	Die Metrik zwischen den beiden Zuständen darf sich nicht verändern.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein. Der Dämpfer darf erst nach der ersten Messung angehängt sein. Die Visualisierung muss neu geladen werden.

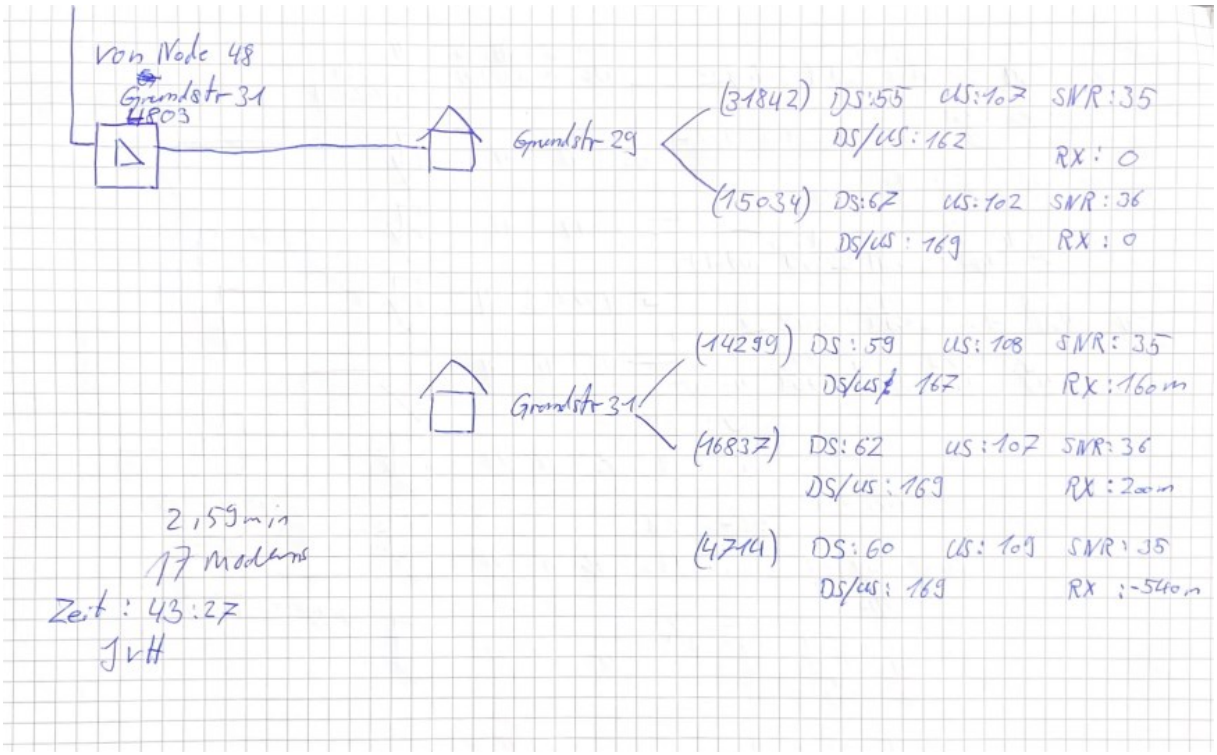
ID	19
Name	RX an Node im Frontend, in der Alert Meldung verändert sich NICHT
Testfall-Beschreibung	Es soll geprüft werden, ob sich der Werte des Modems, welcher in der Alert Meldung angezeigt wird, sich verändert. Einzelne Metriken werden verändert, die hier getestete Metrik darf sich nicht ändern!
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen 5. Modem öffnen (Alert Meldung) 6. Screenshot von den Metriken entnehmen 7. Dämpfer am Modem anhängen. 8. Punkt 1 bis 6 wiederholen. 9. Die beiden werte vergleichen
Erwartete Resultate	Die Metrik zwischen den beiden Zuständen darf sich nicht verändern.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein. Der Dämpfer darf erst nach der ersten Messung angehängt sein. Die Visualisierung muss neu geladen werden.

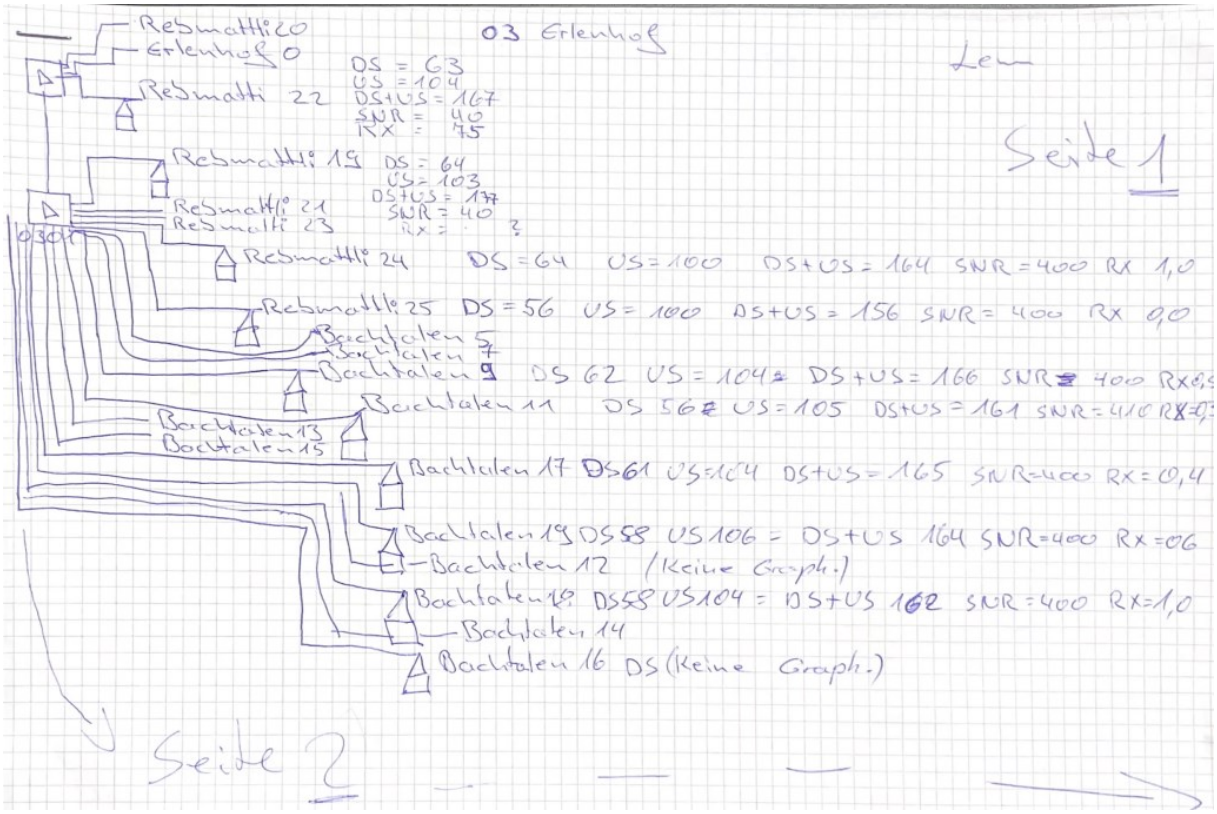
ID	20
Name	DS/US im Frontend, in der Alert Meldung verändert sich NICHT
Testfall-Beschreibung	Es soll geprüft werden, ob sich der Werte des Modems, welcher in der Alert Meldung angezeigt wird, sich verändert. Einzelne Metriken werden verändert, die hier getestete Metrik darf sich nicht ändern!
Testprozeduren	1. Kunden GUI öffnen 2. DCMTS öffnen, an dem das Modem angehängt ist. 3. TreeMap Ansicht öffnen 4. Das Modem an der Adresse suchen 5. Modem öffnen (Alert Meldung) 6. Screenshot von den Metriken entnehmen 7. Dämpfer am Modem anhängen. 8. Punkt 1 bis 6 wiederholen. 9. Die beiden werte vergleichen
Erwartete Resultate	Die Metrik zwischen den beiden Zuständen darf sich nicht verändern.
Vorbedingungen	Die Testumgebung des Backend muss laufen. Der Token muss angepasst werden. Das richtige DCMTS muss gewählt sein. Das zu testende Modem muss online sein. Der Dämpfer darf erst nach der ersten Messung angehängt sein. Die Visualisierung muss neu geladen werden.

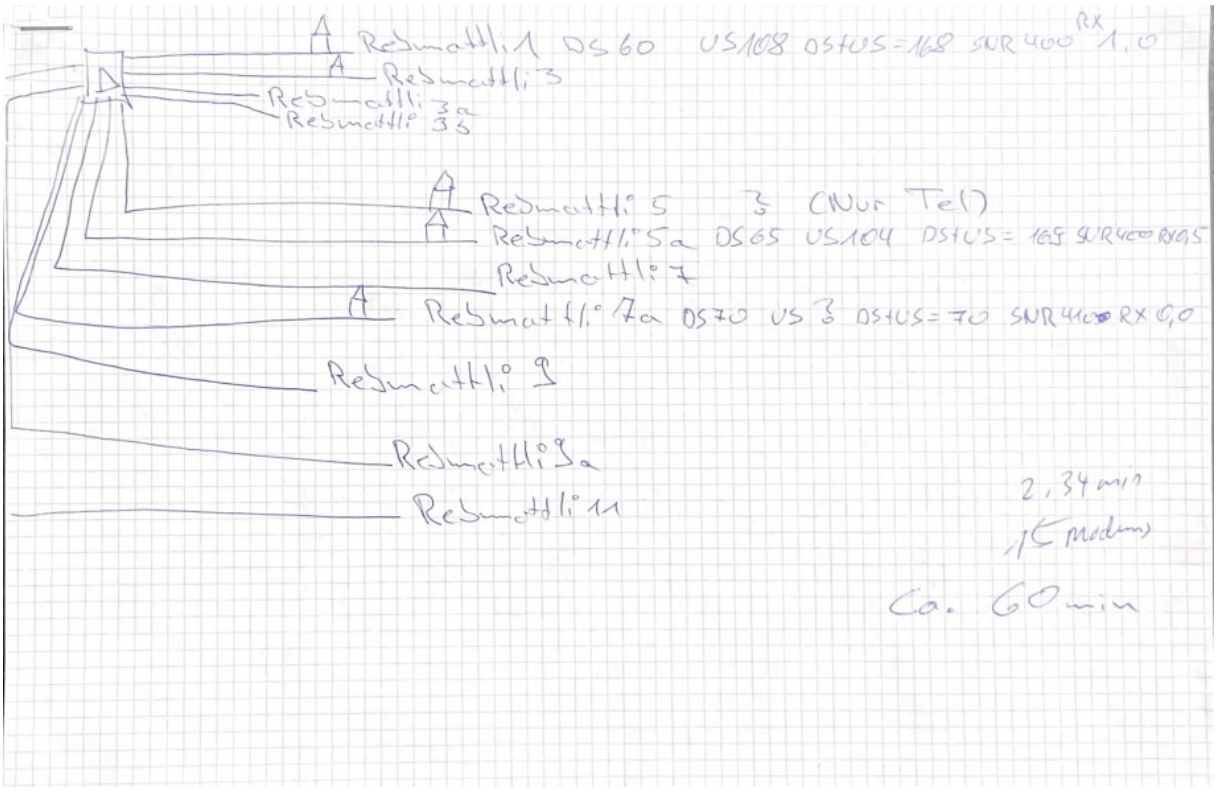
8.3 Bestandesaufnahme der Techniker






von Node 48
Grundstr 31
4803
Grundstr 29
(31842) DS:55 US:107 SNR:35
DS/US: 162
RX: 0
(15034) DS:67 US:102 SNR:36
DS/US: 169 RX: 0
Grundstr 31
(14299) DS: 59 US: 108 SNR: 35
DS/US: 167 RX: 160m
(16837) DS: 62 US: 107 SNR: 36
DS/US: 169 RX: 200m
(4714) DS: 60 US: 109 SNR: 35
DS/US: 169 RX: -540m
2,59 min
17 Modems
Zeit: 43:27
JvH


Rebmatti 20
Erlenhof 0
03 Erlenhof
Lem
Seite 1
Rebmatti 22 DS = 63 US = 104 DS+US = 167 SNR = 40 RX = 75
Rebmatti 19 DS = 64 US = 103 DS+US = 177 SNR = 40 RX = ?
Rebmatti 21
Rebmatti 23
Rebmatti 24 DS = 64 US = 100 DS+US = 164 SNR = 400 RX 1,0
Rebmatti 25 DS = 56 US = 100 DS+US = 156 SNR = 400 RX 0,0
Bachtalen 5
Bachtalen 7
Bachtalen 9 DS 62 US = 104 DS+US = 166 SNR = 400
Bachtalen 11 DS 56 US = 105 DS+US = 161 SNR = 410
Bachtalen 13
Bachtalen 15
Bachtalen 17 DS 61 US 104 DS+US = 165 SNR = 400 RX = 0,4
Bachtalen 19 DS 58 US 106 = DS+US 164 SNR = 400 RX = 0,6
Bachtalen 12 (Keine Graph.)
Bachtalen 18 DS 58 US 104 = DS+US 162 SNR = 400 RX = 1,0
Bachtalen 14
Bachtalen 16 DS (Keine Graph.)
Seite 2


Rebmattli 1 DS 60 US 108 DS+US = 168 SNR 400 RX 1,0
Rebmattli 3
Rebmattli 3a
Rebmattli 3b
Rebmattli 5 3 (Nur Tel)
Rebmattli 5a DS 65 US 104 DS+US = 169 SNR 400 RX 0,5
Rebmattli 7
Rebmattli 7a DS 70 US 3 DS+US = 70 SNR 400 RX 0,0
Rebmattli 9
Rebmattli 9a
Rebmattli 11
2,34 mio
15 modems
Ca. 60 min

9 Abkürzungs-, Abbildungs-, Tabellen-, Quellen-Verzeichnis

9.1 Abkürzungsverzeichnis

CTM ..... *Coaxial Tree Map*
dB ..... dezibel
DB ..... *Datenbank*
DCMTS ..... *Distributed-cable Modem Termination System*
DS ..... *Downstream*
DTO ..... *Datentransferobjekt*
HF ..... *Hochfrequenz*
RX ..... *Receive*
SNMP ..... *Simple Network Management Protocol*
SNR ..... *Signal-to-noise ratio*
US ..... *Upstream*

9.2 Abbildungsverzeichnis

Abbildung 1 HF-Netz 7
Abbildung 2 Systemkomponente Ausgangslage 8
Abbildung 3 Systemkomponente CTM vereinfacht 9
Abbildung 4 Abstraktion der Koaxial-Komponente 11
Abbildung 5 Störungsstelle 12
Abbildung 6 Faktoren für eine hohe Kunden Loyalität 13
Abbildung 7 Dashboard in PRTG 14
Abbildung 8 Dashboard in Zabbix 15
Abbildung 9 Homeoffice zahlen 17
Abbildung 10 Tree 18
Abbildung 11 Treemap 18
Abbildung 12 Pack 19
Abbildung 13 Darstellung als Baum 22
Abbildung 14 SoDa - Software Development Agile 23
Abbildung 15 Risikomatrix (ohne Massnahmen) 26
Abbildung 16 Risikomatrix (mit Massnahmen) 27
Abbildung 17 Gantt Diagramm 29
Abbildung 18 Systemgrenzen 36
Abbildung 19 Use Case - IST 37
Abbildung 20 Use Case - SOLL 37
Abbildung 21 Komponentendiagramm 39
Abbildung 22 Komponentendiagramm Fassade 39
Abbildung 23 DTO 41
Abbildung 24 Adapter 42
Abbildung 25 Sequenzdiagramm - DTO initialisieren 43
Abbildung 26 Sequenzdiagramm - DTO erweitern 45
Abbildung 27 Komponentendiagramm – Frontend 47
Abbildung 28 Funktionsweise Frontend 47
Abbildung 29 Node 10 automatisch vom System 48
Abbildung 30 Node 10 manuell erfasst 49
Abbildung 31 Referenzmodem 49

Abbildung 32 Modem ohne Dämpfer 51
Abbildung 33 Modem mit Dämpfer 51
Abbildung 34 GUI 53
Abbildung 35 Ausblick 57

9.3 Tabellenverzeichnis

Tabelle 1 Nutzwertanalyse 25
Tabelle 2 verwendete Tools 26
Tabelle 3 Risiken 27
Tabelle 4 Massnahmen 27
Tabelle 5 Arbeitsrapport 30
Tabelle 6 Nutzwertanalyse der Varianten 32
Tabelle 7 Nutzwertanalyse Visualisierung 35
Tabelle 8 Testresultate Funktionstest GUI 50
Tabelle 9 Testresultate Funktionstest Dämpfer am Modem 50
Tabelle 10 Auswertung Performance der Techniker 52
Tabelle 11 Auswertung Performance der Komponente 52
Tabelle 13 Erfüllte Technische Anforderung 54
Tabelle 14 Erfüllte Funktionale Anforderungen 55
Tabelle 15 Erfüllung Allgemeine Anforderungen 55
Tabelle 16 Technische Anforderungen 58
Tabelle 17 Funktionale Anforderungen 58
Tabelle 18 Allgemeine Anforderungen 59

9.4 Literaturverzeichnis

Brenner, W., Kruse, J., Zarnekov, R., & Sidler, A. (2008). *Qualität im Internet.* Springer Verlag. doi:10.1007/s00502-008-0550-z

Bundesamt für Statistik. (30. 10 2022). *https://www.bfs.admin.ch/bfs/*. Von https://www.bfs.admin.ch/bfs/de/home/statistiken/kultur-medien-informationsgesellschaft-sport/informationsgesellschaft/gesamtindikatoren/volkswirtschaft/teleheimarbeit.html abgerufen

Buttle, F., & Burton, J. (2001). *Does service failure influence customer loyalty?* Journal of Consumer Behaviour. doi:10.1002/cb.67

d3. (18. 12 2022). *https://github.com/*. Von https://github.com/d3/d3-hierarchy/blob/7bea49efdc1093b28a8d60d2f8717d8a17803564/README.md abgerufen

Fowler, M. (22. 12 2022). *https://martinfowler.com/*. Von https://martinfowler.com/eaaCatalog/dataTransferObject.html abgerufen

Fowler, M. (22. 12 2022). *https://martinfowler.com/*. Von https://martinfowler.com/eaaCatalog/remoteFacade.html abgerufen

Gartner. (23. 10 2022). *https://www.gartner.com/*. Von https://www.gartner.com/reviews/market/it-infrastructure-monitoring-tools abgerufen

HSLU. (02. 11 2022). *https://www.hslu.ch/*. Von https://www.hslu.ch/de-ch/informatik/studium/soda/ abgerufen

Künter, V. (2011). *Branding und Innovation: Chancen für die Telekommuni kationsanbieter.* Springer Verlag. doi:10.1365/s35764-011-0048-z

Manage Engine. (23. 10 2022). *https://www.manageengine.de/*. Von https://www.manageengine.de/produkte-loesungen/netzwerk-performance/opmanager/preise.html abgerufen

Paessler. (23. 10 2022). *https://www.paessler.com/*. Von https://www.paessler.com/de/prtg/prtg-network-monitor abgerufen

Pittet, S. (27. 12 2022). *https://www.atlassian.com/*. Von https://www.atlassian.com/de/continuous-delivery/software-testing/types-of-software-testing abgerufen

PRTG. (28. 10 2022). *https://www.paessler.com/*. Von https://www.paessler.com/prtg/dashboards abgerufen

Röttiger, U. (2019). *Vertrauen und Glaubwürdigkeit in der Unternehmenskommunikation.* Wiesbaden: Springer Fachmedien Wiesbaden. doi:10.1007/978-3-658-03894-6_15-1

Schwaber, K., & Sutherland, J. (01. November 2020). *https://scrumguides.org/.* Von https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-German.pdf abgerufen

Solis, T. (10. 01 2022). *https://www.scribbr.de/*. Von https://www.scribbr.de/methodik/nutzwertanalyse/ abgerufen

Sonntag, A. (2015). *Instrument: Nutzwertanalyse.* Eschborn: RKW Kompetenzzentrum. Von https://www.inf.uni-hamburg.de/de/inst/ab/itmc/research/completed/promidis/instrumente/nutzwertanalyse abgerufen

SUISSEDIGITAL. (2016). *HVA-Richtlinien.* Bern: SUISSEDIGITAL, Verband für Kommunikationsnetze.

Vogel Communications Group. (07. 10 2022). *https://www.ip-insider.de/*. Von https://www.ip-insider.de/top-35-monitoring-tools-gal-8763/ abgerufen

Wikipedia. (20. 03 2017). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Tree_Map&oldid=163778917 abgerufen

Wikipedia. (02. 09 2019). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Kommunikationsdiagramm_(UML)&oldid=191917844 abgerufen

Wikipedia. (01. 10 2021). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Adapter_(Entwurfsmuster)&oldid=216040884 abgerufen

Wikipedia. (07. 09 2021). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Kontextdiagramm&oldid=215391571 abgerufen

Wikipedia. (29. 08 2021). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Messwert&oldid=215152325 abgerufen

Wikipedia. (22. 08 2022). *https://de.wikipedia.org*. Von https://de.wikipedia.org/w/index.php?title=Baum_(Datenstruktur)&oldid=225536417 abgerufen

Wikipedia. (18. 11 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Anwendungsfalldiagramm&oldid=228074638 abgerufen

Wikipedia. (28. 09 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Checkmk&oldid=226583465 abgerufen

Wikipedia. (07. 02 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Fassade_(Entwurfsmuster)&oldid=219983810 abgerufen

Wikipedia. (01. 04 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Klassendiagramm&oldid=221692820 abgerufen

Wikipedia. (18. 11 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/wiki/Komponentendiagramm abgerufen

Wikipedia. (09. 09 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Sequenzdiagramm&oldid=226008218 abgerufen

Wikipedia. (18. 11 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Simple_Network_Management_Protocol&oldid=228081521 abgerufen

Wikipedia. (18. 02 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Transferobjekt&oldid=220341665 abgerufen

Wikipedia. (26. 10 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Unified_Modeling_Language&oldid=227369457 abgerufen

Wikipedia. (06. 07 2022). *https://de.wikipedia.org/*. Von https://de.wikipedia.org/w/index.php?title=Zustandsdiagramm_(UML)&oldid=224268786 abgerufen

Zabbix. (23. 10 2022). Von https://www.zabbix.com/de/network_monitoring abgerufen