

Anmerkung

Nach Abgabe der Dokumentation wurden diverse Fehler entdeckt. Diese können das Verstehen der Dokumentation beeinträchtigen, daher werden deren Korrekturen im Folgenden aufgelistet.

- **Falsche Querverweise**

Wo? Seite 1, Kapitel 1.2 Was ist ein L1-Verschalter?, Zeile 10

Seite 2, Kapitel 1.2 Was ist ein L1-Verschalter?, Zeile 2

Korrektur: $W|EWE|1 \rightarrow P|L1|d$

- **Falscher beziehungsweise überflüssiger Text**

Wo? Seite 3, Kapitel 2.1 Was ist das Problem?, Zeile 12

Korrektur: $1:1 \rightarrow 1:n$ (oder kein Text)

- **Falsche Darstellung von Zahlenwerten**

Wo? Seite ii, A.3 Berechnung der Personalkosten, Tabelle 3 in der letzten Zeile
Durch eine falsche Formatierung der Zellen wurden die Werte auf 18 und 23 gerundet.

Korrektur: $18 \rightarrow 17,5$ $23 \rightarrow 22,5$

Folgefehler: fehlerhafte Rechnungen

Wo? Seite ii, A.3 Berechnung der Personalkosten, Seite iv Rechnungen (7), (8), (9), (10), (11)

Korrektur:

$$16,44 \frac{\text{€}}{\text{Stunde}} * 17,5 \text{ Stunden} = 287,70 \text{ € Teilpers. –K. Azubi} \quad (7)$$

$$43,08 \frac{\text{€}}{\text{Stunde}} * 22,5 \text{ Stunden} = 969,30 \text{ € Teilpers. –K. Intern} \quad (8)$$

$$969,30 \text{ €} + 287,70 \text{ €} = 1.257 \text{ € Pers. –K. Intern} \quad (9)$$

$$50,00 \frac{\text{€}}{\text{Stunde}} * 22,5 \text{ Stunden} = 1.125 \text{ € Teilpers. –K. Extern} \quad (10)$$

$$1.125 \text{ €} + 287,70 \text{ €} = 1.412,70 \text{ € Pers. –K. Extern} \quad (11)$$

Folgefehler: fehlerhafte Zahlenwerte

Wo? Seite 6, Kapitel 2.8 Kostenvergleich mit Personalkostenberechnung, Zeile 7

Korrektur: 18 → 17,5 23 → 22,5

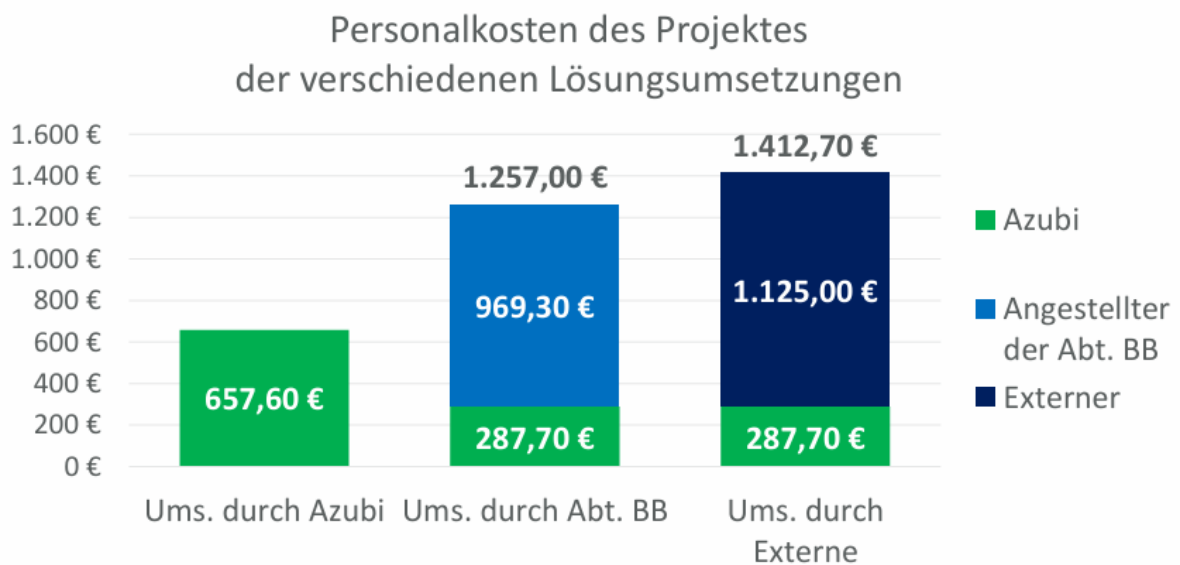
Wo? Seite 7, Kapitel 2.8 Kostenvergleich mit Personalkostenberechnung, Z. 3, 4

Korrektur: 1286,76 → 1257 1445,92 → 1412,70

Folgefehler: fehlerhafte Abbildung

Wo? Seite 7, Kapitel 2.8 Kostenvergleich mit Personalkostenberechnung, Bild

Korrektur:



▪ **Fehlender Elternordner in Pfad**

Wo? Seite xvii, VSCodium: PHP-Code Anpassung, Abbildung 43

Die dunkle Version der Abbildung wurde durch eine helle Version ersetzt, wobei bei beiden Pfadangaben der Elternordner *labor* verloren gegangen ist.

Korrekturbeispiel: `echo "<script src='layer_1.js'></script>";`

↓

`echo "<script src='labor/layer_1.js'></script>";`

Erklärung

Bestätigung über die durchgeführte betriebliche Aufgabe¹

(Diese Bestätigung ist als Deckblatt online einzureichen, gemeinsam mit dem Report/der Dokumentation.)

Prüfling (vollständige Anschrift und Telefonnummer)

Vorname, Name

Straße, Hausnr.

PLZ, Ort

Tel.Nr.:

Ausbildungsbetrieb (vollständige Anschrift)

Firma

Straße, Hausnr.

PLZ, Ort

Tel.Nr.:

Hinweis vorab: Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

Ausbildungsberuf

Bezeichnung der betrieblichen Aufgabe

Erklärung des Prüflings

Hiermit versichere ich, dass ich die betriebliche Aufgabe unter der Betreuung von

Verantwortlicher im Unternehmen

selbstständig durchgeführt und die Unterlagen selbstständig zusammengestellt habe.

Dokumente und Textpassagen, die ich nicht selbstständig erstellt habe, sind von mir gekennzeichnet.

Ort, Datum

Unterschrift des Prüflings

Bestätigung des Ausbildungsbetriebes

Wir bestätigen, dass die Angaben des Prüflings richtig sind.

Ort, Datum

Unterschrift des Verantwortlichen, der die Aufgabe betreut hat.

Ort, Datum

Unterschrift des Ausbilders

¹Zur Vereinfachung wird einheitlich der Begriff „betriebliche Aufgabe“ verwendet. Gemeint sind die Fachaufgabe/die Projektarbeit/der betrieblicher Auftrag. Die unterschiedlichen Bezeichnungen entstehen durch die verschiedenen Berufe, die eine Aufgabe online einstellen.



Abschlussprüfung Sommer 2025
Fachinformatiker für Systemintegration
Dokumentation zur betrieblichen Projektarbeit

L1GUI

Eine grafische Oberflächenlösung zur Nutzung eines Arista L1-Verschalters

Abgabetermin: Oldenburg, den 14.05.2025

Prüfungsbewerber:

Xenia Dusold



EWEnetz

Ausbildungsbetrieb:

EWE NETZ GmbH
Cloppener Straße 302
26133 Oldenburg

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeisung und Verarbeitung in elektronischen Systemen.

Inhaltsverzeichnis

Tabellenverzeichnis	II
Abbildungsverzeichnis	III
1 Einleitung	1
1.1 Wie sieht das Umfeld aus?	1
1.2 Was ist ein L1-Verschalter?	1
1.3 Was sind die Backbone-Tools?	2
2 Vorbereitung und Planung	2
2.1 Was ist das Problem?	3
2.2 Was soll durch das Projekt erzielt werden?	3
2.3 Erfassung des Ist-Zustandes	3
2.4 Definition des Soll-Zustandes	3
2.5 Erstellung der Lösungsvorschläge	4
2.5.1 Weiterentwicklung von Arista Switch Web UI	4
2.5.2 Weiterentwicklung von Arista_WEB_GUI	4
2.5.3 Entwicklung einer Eigenlösung	4
2.6 Anwendung einer Präferenzmatrix	5
2.7 Anwendung einer Nutzwertanalyse auf die Lösungen	6
2.8 Kostenvergleich mit Personalkostenberechnung	6
3 Verwirklichung der Lösung	7
3.1 Konfiguration des L1-Verschalters	7
3.2 Entwicklung der Lösung	8
3.2.1 API-Kommunikation	8
3.2.2 Datenverarbeitung und -darstellung	9
3.2.3 PHP-Formulare	9
3.2.4 Element Header	9
3.2.5 Sektion Legende	9
3.2.6 Sektion Ethernet Ports	10
3.2.7 Sektion Management Ports	10
3.2.8 Sektion Descriptions	10
3.2.9 Sektion Verschaltungen	10
3.2.10 Sektion Komponenten	10
3.2.11 Sektion System	10
3.2.12 Sektion Temperaturen	10
3.2.13 Sektion Alarme	10
3.2.14 Element Tooltip	11
3.3 Integration der Lösung in die bestehende IT-Landschaft	11
3.4 Sind die Anforderungen erfüllt?	12
4 Testen und Fehlerprüfung	13
4.1 Überprüfung der Funktionsfähigkeit der Lösung	13
5 Dokumentation	13
5.1 Anfertigung der Betriebsdokumentation	13
5.2 Anfertigung der Anwenderdokumentation	13
5.3 Einweisen der Administratoren und Anwender in die Lösung	13
6 Abschluss	14
6.1 Wirtschaftlichkeitsanalyse	14
6.2 Soll-Ist-Vergleich	15
6.3 Fazit und Ausblick	15

Quellenverzeichnis IV

A Anhang.....	i
A.1 Zielschema.....	i
A.2 Anforderungsliste.....	ii
A.3 Berechnung der Personalkosten	ii
A.4 API-Konfiguration.....	v
A.5 Entwicklung der Verwaltungsoberfläche.....	vi
A.6 Integration mit VSCodium und PuTTY	xvii
A.7 Integrierte Verwaltungsoberfläche	xix
A.8 Soll-Ist-Zeitvergleich	xx
A.9 Ermittlung des Break-Even-Point	xx

Tabellenverzeichnis

Tabelle 1: Präferenzmatrix	5
Tabelle 2: Nutzwertanalyse	6
Tabelle 3: Zeitliche Aufgabenaufteilung zwischen Azubi und Intern/Extern	ii
Tabelle 4: Durchschnittliche Arbeits- und Krankheitstage pro Jahr von Beschäftigten in Niedersachsen.....	iii
Tabelle 5: Berechnung der Jahreskosten durch einen Azubi.....	iii
Tabelle 6: Berechnung der Jahreskosten durch einen Angestellten der Abteilung BB	iv
Tabelle 7: Soll-Ist-Zeitvergleich	xx

Abbildungsverzeichnis

Abbildung 1: Grobe Zeitplanung.....	2
Abbildung 2: Vergleich der Personalkosten	7
Abbildung 4: Zeit-Euro-Diagramm: Ermittlung des Break-Even-Point	14
Abbildung 5: Zielschema	i
Abbildung 6: Zugriff auf den L1S mit PuTTY	v
Abbildung 7: Aktivierung und Konfiguration der eAPI auf dem L1S.....	v
Abbildung 8: Anzeige und Überprüfung des eAPI-Status auf dem L1S.....	v
Abbildung 9: Start des PHP-Servers über das Programm XAMPP	vi
Abbildung 10: Lokal angelegte Quelldateien.....	vi
Abbildung 11: Webseite der eAPI: Anfrage: show running-config	vii
Abbildung 12: Webseite der eAPI: Anfrage mit JSON-Parametern: show running-config.....	vii
Abbildung 13: PHP: Funktion für API-Zugriff.....	viii
Abbildung 14: PHP: API-Daten mit CLI-Kommandos über Funktion holen und aufteilen (Auszug)	viii
Abbildung 15: PHP: Daten als JavaScript-Variable speichern: Beispiel Interface Descriptions	viii
Abbildung 16: JS: Anhängen der verarbeiteten Daten an Dokumentelement für Sektion Descriptions	ix
Abbildung 17: CSS: Stile der Klassen, die für die farbliche Darstellung des Portstatus nötig sind	ix
Abbildung 18: JS: Portstatus zuweisen über gleichnamige Klassen	ix
Abbildung 19: Popup-Formular für das Anpassen der Porteigenschaften	x
Abbildung 20: Popup-Formular für das Erstellen von Verschaltungen	x
Abbildung 21: Popup-Formular für das Löschen von Verschaltungen.....	x
Abbildung 22: PHP: HTML-Code für die Formulare	xi
Abbildung 23: JS: Hinzufügen der Auswahlmöglichkeiten bei Verschaltungsformular	xi
Abbildung 24: JS: Anzeigen der Daten aus Sektion Descriptions für Portanpassungsformular	xi
Abbildung 25: PHP: Logik und Funktionen für die Formulare	xii
Abbildung 26: Header mit Zeit.....	xii
Abbildung 27: PHP: HTML-Code für Header	xii
Abbildung 28: Sektion Legende: Legende für Portstatus	xii
Abbildung 29: Sektion Management Ports	xiii
Abbildung 30: Sektion Ethernet Ports	xiii
Abbildung 31: PHP: Funktion für Port-Layout	xiii
Abbildung 32: Popup: Port- und SFP-Daten	xiv
Abbildung 33: Sektion Descriptions: Liste der Ports mit Eigenschaften (Auszug)	xiv
Abbildung 34: Sektion Verschaltungen: Liste der Verschaltungen (Auszug)	xv
Abbildung 35: Sektion Komponenten: Nur Komponente Fans aufgeklappt.....	xv
Abbildung 36: Sektion System (Auszug)	xv
Abbildung 37: Sektion Temperaturen (Auszug)	xvi
Abbildung 38: Sektion: Alarme (Auszug).....	xvi
Abbildung 39: Tooltip: Tooltip zu Portdetails anzeigen und ausblenden	xvi
Abbildung 40: Installierte SSH-Erweiterung für VSCodium.....	xvii
Abbildung 41: Anlegen des SSH-Zuganges zum Server	xvii
Abbildung 42: Über VSCodium angelegte und übertragene Quelldateien.....	xvii
Abbildung 43: PHP-Code Anpassung für Integration: Hinzufügen der Content-Nummer	xviii
Abbildung 44: PHP-Code Anpassung für Integration: Einbindung der verlinkten Dateien anpassen	xviii
Abbildung 45: Zugriff auf Server mit PuTTY.....	xviii
Abbildung 46: Dateirechte der über VSCodium angelegten Dateien	xviii
Abbildung 47: Anpassung der Dateirechte der angelegten Dateien	xix
Abbildung 48: Integrierte Verwaltungsoberfläche für den L1S.....	xix

1 Einleitung

Mit dieser Dokumentation wird die Durchführung des IHK-Projektes *L1GUI - Eine grafische Oberflächenlösung zur Nutzung eines Arista L1-Verschalters* festgehalten. Dieses Projekt fand im Zeitraum vom 20.02.2025 bis zum 13.05.2025 als Teil der Abschlussprüfung des Fachinformatikers für Systemintegration statt.

Das Projekt wurde nicht im Ausbildungsbetrieb EWE NETZ GmbH (EWE NETZ), sondern bei der EWE TEL GmbH (EWE TEL) bzw. bei der Abteilung TK-TK-T-IF Backbone (BB) der EWE TEL durchgeführt.

Im Weiteren ist anzumerken, dass der Kostenvergleich in der Phase *Vorbereitung und Planung* sich nicht wie im Projektantrag angenommen auf alle Lösungsvorschläge bezieht, sondern nur genutzt wird, um zu ermitteln, von wem der ausgewählte Vorschlag verwirklicht werden soll. Diese Abweichung führt dazu, dass der Kostenvergleich abweichend vom Projektplan erst nach der Nutzwertanalyse stattfindet.

1.1 Wie sieht das Umfeld aus?

Die EWE TEL ist eine Tochtergesellschaft der EWE AG und ein Telekommunikationsunternehmen, das rund 1.300 Mitarbeiter beschäftigt. Ihr Firmensitz befindet sich in Oldenburg.

Derzeit betreut die EWE TEL rund 700.000 Kunden im privaten als auch geschäftlichen Bereich, wobei sich Produkte für Privatkunden auf Niedersachsen, Bremen und Teile Brandenburgs sowie Teile Nordrhein-Westfalens beschränken, während Produkte für Geschäftskunden deutschlandweit vermarktet werden. Mögliche Produkte gehören dabei zu Sparten wie Internet, Mobilfunk und Festnetztelefonie. [[W|EWE|1](#), [W|EWE|2](#), [P|EWE|1](#) S. 8]

Als Telekommunikationsunternehmen übernimmt die EWE TEL für die EWE NETZ die primäre Ausbildung von Fachinformatikern der Fachrichtung Systemintegration und anderen IT-Ausbildungen. Eine der Abteilungen, die Ausbildungstätigkeiten übernehmen, ist die Abteilung BB.

Die Abteilung BB ist für den Betrieb und die Änderungen des Backbone-Netzes der EWE TEL zuständig. Hierfür werden die Netzelemente des Backbone-Netzes unter anderem regelmäßig durch neue und oder bessere Geräte ausgetauscht, auf den neusten Softwarestand aktualisiert und teilweise neu konfiguriert. Zudem wird das Netz stetig erweitert, um der wachsenden Nachfrage gerecht zu werden.

1.2 Was ist ein L1-Verschalter?

Der Layer-1-Verschalter (L1S) ist eines der Geräte, das von der Abteilung BB verwaltet wird. Dieser befindet sich im Labor der EWE TEL. Dort werden verschiedene Netzwerktopologien simuliert und Protokoll- und Durchsatzanalysen durchgeführt.

Der L1S ist ein Layer 1-Switch und kann als Crosspoint-Switch und Medienkonverter agieren. Als Crosspoint-Switch kann der Switch als remote konfigurierbares Patchfeld verwendet werden, dies ermöglicht beim Verschalten von angeschlossenen Patchen (Glasfaser, Kupferkabel) den Wegfall von händischem und womöglich fehlerhaftem Rangieren. Durch die Funktion als Medienkonverter sind Verbindungen zwischen Rangierungen mit Glasfaser und mit RJ45-Kabeln, sowie Rangierungen mit unterschiedlicher Datenrate von 100 Mbit/s bis zu 10 Gbit/s möglich, solange die Übertragungsrate übereinstimmt. [[W|EWE|1](#), [P|L1|g](#)]

Beim Verschalten können bidirektionale und unidirektionale Verbindungen erstellt werden. Dabei besitzt die bidirektionale eine 1:1 Beziehung und die unidirektionale eine 1:n Beziehung.

Die 1:n Verbindung wird für das Broadcast von einem Eingangsport auf mehrere Ausgangsports genutzt. [[W|EWE|1](#)]

1.3 Was sind die Backbone-Tools?

Die Backbone-Tools sind ein Cluster aus mehreren PHP-Webseiten, die die Verwaltung des Backbone-Netzes unterstützen. Dieser Cluster wird, wie der Server, auf dem dieser sich befindet, von der Abteilung BB betrieben.

Der Webserver mit seinen Seiten vereinfacht unter anderen die Konfiguration von Routern, hilft bei der Zuweisung von IP-Netzen zu einer CPE (Customer Premises Equipment), die diese an die Kunden verteilt, und stellt das Monitoring von bestimmten Netzelementen besser dar.

Der Zugang und Zugriff auf die Backbone-Tools beziehungsweise die Webseiten ist nur über das Admin-Netz möglich, wobei ein etabliertes Rollen- und Berechtigungsmanagement regelt wer auf was zugreifen darf.

2 Vorbereitung und Planung

Um das IHK-Projekt durchführen zu können, muss das Projekt definiert und geplant werden. Dabei sind Punkte wie das Projektproblem, das Projektziel, der Ist-Zustand, der Soll-Zustand, mögliche Lösungsvorschläge und deren Bewertung sowie eine voraussichtliche Zeitplanung zu betrachten.

Für dieses Projekt stehen laut IHK 40 Stunden zur Verfügung. Diese Stunden sind vorläufig auf vier grobe Hauptphasen aufgeteilt und können in der nächsten Abbildung betrachtet werden. Dabei handelt es sich um eine grobe Schätzung. Eine ausführlichere SOLL-Zeitplanung findet sich in der *Tabelle 7: Soll-Ist-Zeitvergleich* im Anhang auf Seite xx.

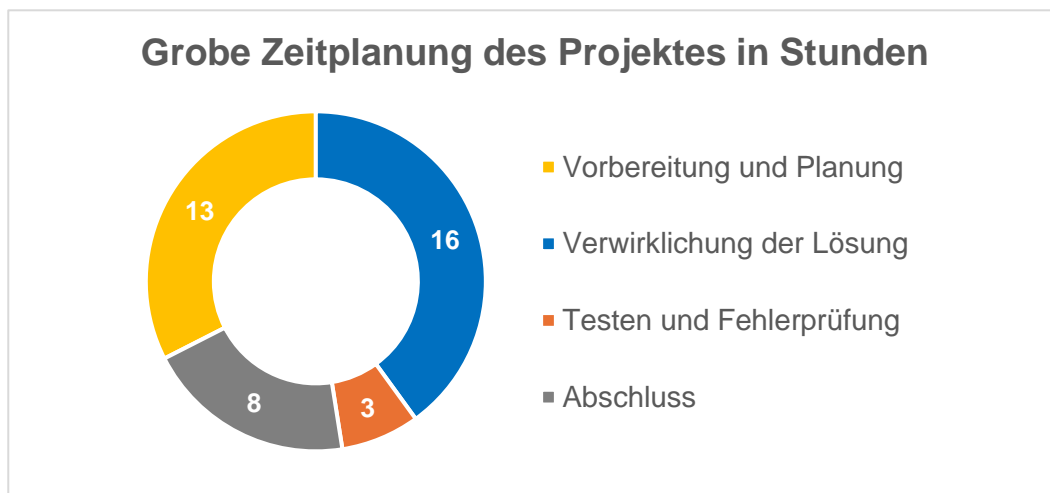


Abbildung 1: Grobe Zeitplanung

Im Weiteren ist die Abteilung BB bei diesem Projekt als Kunde zu betrachten.

2.1 Was ist das Problem?

Der L1S kann von Personen der Abteilung BB und anderen Abteilungen, die die entsprechenden Berechtigungen haben, bedient werden. Hierzu muss die bedienende Person die L1S-Anmeldedaten kennen und L1S-Konsolenkenntnisse besitzen.

Personen, die die Konfiguration des L1S anpassen müssen, aber keine Anmeldedaten und oder Konsolenkenntnisse haben, müssen sich an die Abteilung BB wenden, sodass die gewünschten Anpassungen erfolgen.

Jedoch soll es Personen, die keine besondere Kenntnisse zur Nutzung des L1S besitzen, auch möglich sein, ohne Hilfe der Abteilung BB, den L1S zu bedienen bzw. bestimmte Änderungen an Ports und Verschaltungen zu tätigen. So soll die Abteilung BB entsprechend entlastet werden.

Des Weiteren kann es vorkommen, dass beim Erstellen, Löschen und Bearbeiten von bidirektionalen 1:1-Portverschaltungen ungewollte unidirektionale 1:1-Verbindungen entstehen. Solche einseitigen Verbindungen sollen vermieden werden.

2.2 Was soll durch das Projekt erzielt werden?

Um dieses Problem bzw. Probleme zu beheben, soll eine graphische Oberfläche implementiert werden, welche die Verwaltung des L1S erleichtert, wichtige Daten darstellt und veranschaulicht, sowie mögliche fehlerhafte Konfigurationen verhindert.

Dabei sollen wirtschaftliche sowie benutzerfreundliche Aspekte beachtet werden.

2.3 Erfassung des Ist-Zustandes

Der L1S kann innerhalb des Admin-Netzes mit dem Protokoll SSH verwaltet werden. Eine Anpassung der Konfiguration kann auch über das Konsolenkabel durchgeführt werden. In beiden Fällen wird das textbasierte Terminalinterface des L1S für das Konfigurieren verwendet.

Um als Admin auf dem L1S Änderungen vornehmen zu können, muss man sich als lokaler Admin mit einem Passwort anmelden. Eine integrierte graphische Möglichkeit, die Einstellungen des L1S anzupassen, gibt es nicht.

Auf das Admin-Netz kann nur über ein VPN und mit den entsprechenden Berechtigungen zugegriffen werden. (Siehe oberer Bereich in *Abbildung 4: Zielschema* im Anhang auf Seite i)

2.4 Definition des Soll-Zustandes

Für die weitere Projektplanung wird die erwähnte Konfiguration über das Konsolenkabel außeracht gelassen, da diese nur für Angestellte mit Labor-Zutrittsberechtigung relevant ist.

Im unteren Bereich der *Abbildung 4: Zielschema* im Anhang auf Seite i ist ein grober Projektaufbau dargestellt. In diesem greift der Anwender über den VPN-Tunnel mit einem noch undefinierten Protokoll auf die graphische Oberfläche des L1S zu, um diesen zu verwalten.

Um dieses Projekt umzusetzen, müssen die gewünschten Anforderungen von der Abteilung BB definiert werden. Diese Anforderungen an die graphische Oberfläche für den L1S sind als Checkliste in der *Anforderungsliste* im Anhang auf Seite ii zu finden beziehungsweise in den nächsten drei Paragraphen zu lesen.

Die Lösung soll nur bestimmte Funktionen des L1S graphisch ermöglichen und nicht auf der CLI basieren. Personen ohne Kenntnisse über den L1S sollen die GUI bedienen können. Des Weiteren soll die Realisierung in die vorhandene IT-Landschaft integrierbar sein.

Es sollen die Portzustände des L1S graphisch dargestellt werden und Informationen zu Ports auf Wunsch des jeweiligen Anwenders sichtbar sein. Zudem sollen weitere Informationen wie aktive Verschaltungen und Betriebsparameter angezeigt werden. Die Lösung soll es ermöglichen, dass der Anwender Portzustand und -beschreibung aktiv ändern kann. Dieser soll auch Port-Verschaltungen anpassen können. Bei der Erstellung von Port-Verschaltungen sollen nur bidirektionale Verbindungen möglich sein.

Beim Zugriff soll sich der Anwender nicht mit den CLI-Anmeldedaten anmelden müssen. Die Anmeldung soll über das bestehende Rollen- und Berechtigungsmanagement geschehen.

Für die GUI gibt es auch Einschränkungen. So soll die GUI nicht vom Internet aus erreichbar sein. Die graphische Lösung soll sich im Admin-Netz befinden und nur berechtigten Nutzern Zugriff gewähren.

2.5 Erstellung der Lösungsvorschläge

Auf der Suche nach Lösungsmöglichkeiten für Arista-Switches im Internet konnten nur Verwaltungsweboberflächen für normale Switches aber keine Komplettlösung für einen L1-Verschalter gefunden werden. Diesen Weboberflächen fehlt es an der Möglichkeit Layer 1-Eigenschaften anzupassen. Das müsste, wenn eine dieser GUIs ausgewählt wird, implementiert werden.

2.5.1 Weiterentwicklung von Arista Switch Web UI

Die erste Weboberfläche, die gefunden wurde, ist die GUI *Arista Switch Web UI* von Github-User Vertigo28. Deren Quellcode besteht größtenteils aus Vue und zu geringeren Teilen aus JavaScript, CSS und HTML.

Laut Beschreibung kann dieses Repository Interface-Einstellungen, wie Portstatus, Portbeschreibung, zugewiesene VLANs und Trunk/Access-Modus, verändern. Zudem können mehrere Ports gleichzeitig konfiguriert werden.

Um dieses Repository auszuführen, muss zusätzlich die Laufzeitumgebung *Node.js* und der Paketmanager *npm* installiert werden. [[W|GUI|1](#)]

2.5.2 Weiterentwicklung von Arista_WEB_GUI

Die andere Weboberfläche, die gefunden wurde, ist die GUI *Arista_WEB_GUI* von Github-User YonatanPorat. Deren Quellcode besteht größtenteils aus JavaScript und aus kleineren Teilen von HTML, Python, CSS und Shell.

In der Beschreibung wird recht knapp festgehalten, dass das Repository eine einfache Weboberfläche für Arista Switches sei. Das macht es schwierig einzuschätzen, was für Funktionen vorhanden sind. Es ist jedoch zu erkennen, dass typische Interface-Einstellungen, wie Portbeschreibung und Portstatus, angepasst werden können. [[W|GUI|2](#)]

2.5.3 Entwicklung einer Eigenlösung

Der letzte Vorschlag ist eine Eigenlösung einer Verwaltungs-GUI für den L1S zu entwickeln. Dabei muss die Form der GUI und die Serversprache sowie die Codesprache für das Frontend festgelegt werden.

Als Form wird die Web-GUI bestimmt, da diese keine zusätzliche Installation wie Applikations-GUIs benötigt und flexibel über die verschiedensten Webbrowser erreichbar ist.

Bei der Wahl der Serversprache gibt es unter anderen die Möglichkeiten PHP, JavaScript und Python zu nutzen. Wobei bei allen Optionen für das Frontend Websprachen wie HTML,

JavaScript und CSS verwendet werden. Zudem sind bei jeder Möglichkeit zusätzliche Software oder Module nötig, so ist für PHP ein PHP-Server, für JavaScript die Bibliothek *React*, die Laufzeitumgebung *Node.js* und der Paketmanager *npm*, und für Python ein für den API-Zugriff nötiges Modul, wie *pyeapi* oder *JSONRPClib*, erforderlich. [[W|PY|1](#), [W|PY|2](#)]

Dadurch, dass bereits ein PHP-Server für die Backbone-Tools vorhanden ist und dieser für die Lösung mitgenutzt werden kann, wird PHP als Serversprache ausgewählt. Das hat den Vorteil, dass kein zusätzlicher Server installiert und konfiguriert werden muss. Und ermöglicht eine einfache Integration in die bestehende IT-Landschaft, genauer in die Backbone-Tools. Obendrein muss die gewollte Anmeldung über das bestehendes Rollen- und Berechtigungsmanagement nicht in der GUI implementiert werden, da diese Anmeldung beim Zugriff auf die Backbone-Tools bereits erfolgt.

2.6 Anwendung einer Präferenzmatrix

Um einen Vorschlag aus den oben genannten Möglichkeiten auszuwählen, muss zuerst entschieden werden, wie diese in einer Nutzwertanalyse bewertet werden sollen. Zu diesem Zweck wurden im Gespräch mit der Abteilung BB Bewertungskriterien festgelegt und mithilfe einer Präferenzmatrix die jeweilige Gewichtung bestimmt.

Gewichtung	Rang	Anzahl	Kriterium		(erstellt in Excel)					
13,33%	3	2	a	Zeitaufwand	b					
26,67%	2	4	b	Code-Verständnis		a				
6,67%	5	1	c	zz. Installationen	b		d			
6,67%	5	1	d	Gestaltung		b		e		
33,33%	1	5	e	Risiko	c		e		a	
13,33%	3	2	f	Wartbarkeit		e		b		
100,00%		15			e		f			

Tabelle 1: Präferenzmatrix

Die festgelegten Kriterien sind Zeitaufwand, Code-Verständnis, zusätzliche Installationen, Gestaltung, Risiko und Wartbarkeit.

Unter Zeitaufwand ist der geschätzte Zeitaufwand zu verstehen. Mit Code-Verständnis ist das Verständnis für den Code bzw. die verwendete Programmiersprache aufzufassen. Weil dies nicht nur beim Entwickeln, sondern auch bei der Analyse und der Anpassung von Software, wichtig ist. Zusätzliche Installationen bewertet den Aufwand von zusätzlich nötiger Software. Beim Kriterium Gestaltung geht es darum wie einfach die GUI designt werden kann. Mit Risiko wird die mögliche Gefahr, die durch den Code entstehen kann, wie Schadsoftware bewertet. Unter Wartbarkeit ist zu verstehen, wie einfach die Lösung von der Abteilung BB angepasst und womöglich neu implementiert werden kann.

2.7 Anwendung einer Nutzwertanalyse auf die Lösungen

Mit den aus dem vorherigen Kapitel *Anwendung einer Präferenzmatrix* bestimmten Kriterien und Gewichtungen wird eine Nutzwertanalyse auf die Lösungsvorschläge angewandt, um den besten Vorschlag zu ermitteln.

Kriterium	Gew.	Arista Switch Web UI		Arista_WEB_GUI		Eigenlösung	
		P	Gew. P	P	Gew. P	P	Gew. P
Zeitaufwand	13,33%	2	0,27	3	0,40	1	0,13
Code-Verständnis	26,67%	1	0,27	2	0,53	2	0,53
zz. Installationen	6,67%	1	0,07	3	0,20	3	0,20
Gestaltung	6,67%	3	0,20	2	0,13	2	0,13
Risiko	33,33%	1	0,33	1	0,33	3	1,00
Wartbarkeit	13,33%	1	0,13	2	0,27	3	0,40
Gesamt	100%		1,27		1,87		2,40
Rang			3		2		1

Gew. für Gewichtung, **P** für Punkte; von 1 (schlecht) zu 3 (gut)

(erstellt in Excel)

Tabelle 2: Nutzwertanalyse

Laut Nutzwertanalyse ist der beste Vorschlag die Eigenlösung mit 2,40 gewichteten Punkten. Darauf folgen die Weboberflächen Arista_WEB_GUI mit 1,87 und Arista Switch Web UI mit 1,27.

2.8 Kostenvergleich mit Personalkostenberechnung

Nachdem feststeht welcher Lösungsvorschlag umgesetzt wird, muss in Betracht gezogen werden, von wem die Eigenlösung umgesetzt werden soll. Zur Auswahl steht, dass ich als Azubi, ein Angestellter der Abteilung BB oder eine externe Person die Lösung entwickeln.

Die Entscheidung soll mit einem Kostenvergleich beziehungsweise einem Personalkostenvergleich getroffen werden. Dabei muss beachtet werden, dass, bei der Einbindung einer anderen Person für die Entwicklung der Lösung, die geplanten 40 Arbeitsstunden in 18 Stunden für mich als Azubi und 23 Stunden für die andere Arbeitskraft aufgeteilt werden (Siehe *Tabelle 3: Zeitliche Aufgabenaufteilung zwischen Azubi und Intern/Extern* im Anhang auf Seite ii).

Die daraus folgenden berechneten Personalkosten sind in der nächsten Abbildung dargestellt. Die genaue Berechnung dieser Ergebnisse beziehungsweise der Personalkosten befindet sich unter *Berechnung der Personalkosten* im Anhang auf Seite ii.

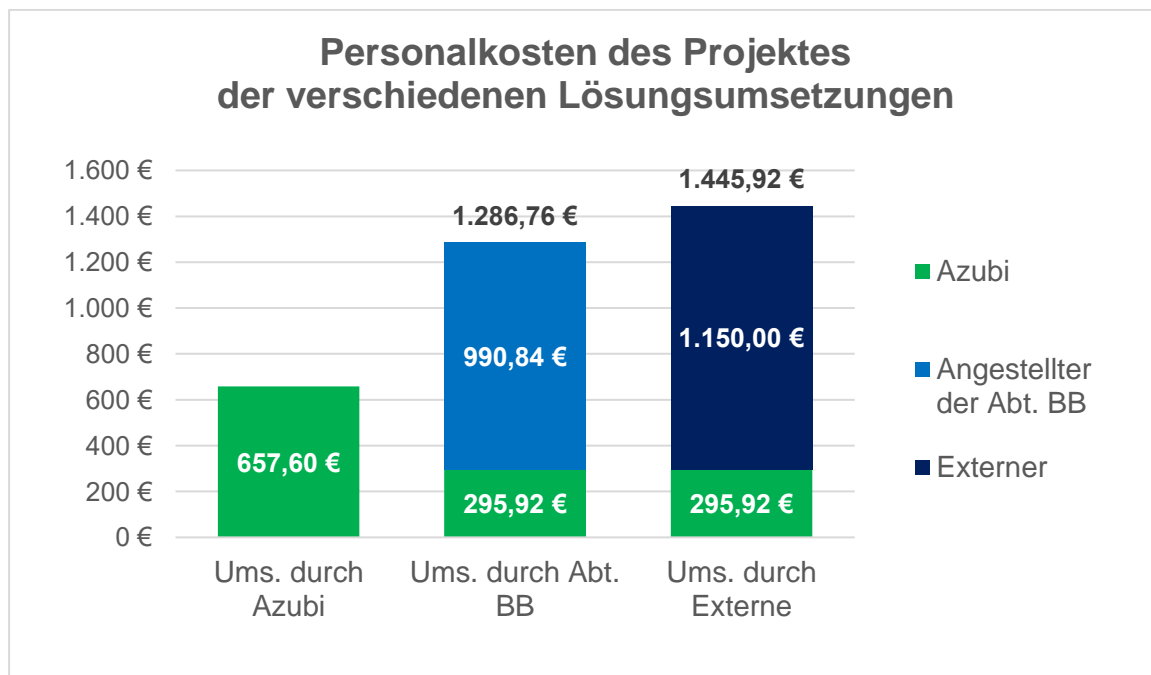


Abbildung 2: Vergleich der Personalkosten

Aus dem obigen Vergleich lässt sich erkennen, dass die alleinige Umsetzung durch mich als Azubi mit 657,60 € (6) die kostengünstigste Option ist. Darauf folgt die Lösungsumsetzung durch einen Mitarbeiter der Abteilung BB mit 1286,76 € (9) und durch eine externe Arbeitskraft mit 1445,92 € (11).

3 Verwirklichung der Lösung

Nachdem eine Lösung gewählt wurde, muss der Zugriff auf das Admin-Netz gewährleistet werden. Dafür richtete die Abteilung BB einen SSL-Zugang mit entsprechenden Berechtigungen ein, was die Anmeldung an dem SSL-VPN und somit den Zugriff auf das Admin-Netz ermöglichte. Zudem stellte die Abteilung BB Zugangsdaten zu den relevanten Netzkomponenten bereit.

Dies ist wichtig und ist nötig für den Zugriff auf den L1S, die Entwicklung der Lösung und die spätere Integration in die Backbone-Tools.

Zugriffe auf den L1S und die Server fanden mit dem Programm *PuTTY* über SSH beziehungsweise bei der Integration auch mit dem Programm *VSCodium* statt.

3.1 Konfiguration des L1-Verschalters

Für eine funktionierende Verwaltungsoberfläche ist die erfolgreiche Kommunikation zwischen dieser und dem L1S wichtig. Das kann über SSH an die CLI oder über eine API geschehen.

Die Übertragung und Ausführung von Kommandobefehlen über SSH an die CLI funktioniert gut, jedoch werden angefragte Daten als ASCII-Text zurückgegeben. Der ASCII-Text muss erst entsprechend formatiert werden, damit dieser für die Datenverarbeitung brauchbar ist. Das ist ein Ausschlusskriterium für diese Option, da die vom L1S gestellten APIs in XML oder JSON antworten.

Die vom L1S gestellten APIs sind mit unter anderem NETCONF, RESTCONF und eAPI. Alle der genannten APIs bis auf NETCONF kommunizieren mit Clients über HTTP/HTTPS und im

Format JSON. NETCONF nutzt stattdessen das Protokoll SSH und das Format XML. [[W/API/1](#), [W/API/2](#), [P/API/1](#)]

Bei NETCONF und RESTCONF, die beide auf dem Datenmodell von OpenConfig beruhen, wurde festgestellt, dass wichtige Daten zu Layer 1, wie verschaltete Ports, nicht abfragbar sind. Was die beiden APIs als alleinige Option unmöglich macht. Eine gleichzeitige Nutzung von zwei APIs wie RESTCONF und eAPI ist nicht möglich, da es sich beim Testen herausstellte, dass nur eine der beiden aktiv sein kann. [[W/API/1](#), [W/API/2](#), [P/API/1](#)]

Infolgedessen kommt nur die Schnittstelle eAPI für den Datenverkehr zwischen dem L1S und der Verwaltungsoberfläche infrage.

Die Konfiguration der Schnittstelle eAPI belief sich auf die Aktivierung der API, die Wahl des Protokolls HTTPS und die Überprüfung, ob die Schnittstelle fehlerfrei läuft (Siehe *API-Konfiguration* im Anhang auf Seite v). [[W/API/3](#)]

3.2 Entwicklung der Lösung

Die Entwicklung der Webseitenlösung findet lokal auf dem eigenen Laptop statt. Dafür wird ein lokaler PHP-Testserver benötigt, welcher durch die Installation des Programmes *XAMPP* bewerkstelligt wird.

Das Coding auf dem lokalen PHP-Server hat den Vorteil, dass von fehlerhaftem Code verursachte Probleme vor der Integration ausgemerzt werden können und der Betriebsserver nicht beeinträchtigt wird. Das hätte unter anderem zum Ausfall der Backbone-Tools führen können.

Für das Coding wird das Programm *VSCodium* genutzt. Die erste Datei, *layer_1_control.php*, ist eine PHP-Datei, die die Grundstruktur der Webseite vorgibt und die anderen Dateien einbindet. Darüber hinaus ist die Datei für die API-Kommunikation, wie das Abfragen von Daten und das Übermitteln von Konfigurationsanpassungen, zuständig, teilt die erhaltenen Daten auf und speichert diese als JavaScript-Variablen. Diese Variablen werden in der JavaScript-Datei, *layer_1.js*, verwendet. Die JavaScript-Datei ist für die Datenverarbeitung, die Datendarstellung und die Funktionalitäten, die sich nicht direkt auf die API-Kommunikation beziehen, verantwortlich. Die letzte Datei, *layer_1.css*, ist eine CSS-Datei und legt die Stile der Webelemente fest (Siehe *PHP-Server und Quelldateien* im Anhang auf Seite vi).

Die Webseite besteht aus mehreren Elementen, wobei einklappbare Elemente als Sektion zu verstehen sind. Als Sektionen gelten die Elemente Legende, Ethernet Ports, Management Ports, Descriptions, Verschaltungen, Komponenten, System, Temperaturen und Alarme. Andere Elemente sind Header und mehrere Tooltips. Sektionen, die nicht für die grafische Portdarstellung nötig sind, sind standartmäßig eingeklappt und es wird maximal eine dieser Sektionen offen dargestellt.

3.2.1 API-Kommunikation

Die API-Kommunikation wird in der PHP-Datei geregelt und funktioniert über die CLI-Befehle des Switches. Dafür wurde eine Funktion formuliert, die die CLI-Befehle mit einer HTTPS-Anfrage übermittelt. Für das Senden der Anfrage wird das Curl-Modul verwendet, wobei vorher Parameter wie URL, User, Passwort und Anfragetyp festgelegt wurden. Die Message der Anfrage erfordert bestimmte Parameter und Werte im JSON-Format, damit die Übertragung der Befehle an die API gelingt. Diese wurden der eAPI-Dokumentation entnommen (Siehe *Kommunikation mit der API* im Anhang auf Seite vii).

Beim Ausführen der PHP-Datei wird die genannte Funktion mit einer Liste von Befehlen ausgeführt, um die nötigen Daten zu holen. Diese werden danach aufgeteilt und verschiedenen Variablen zugeordnet.

3.2.2 Datenverarbeitung und -darstellung

Nach der Aufteilung werden die Daten als JavaScript-Variablen gespeichert, damit diese für den Quellcode der JavaScript-Datei erreichbar ist (Siehe *Verarbeitung und Darstellung der Daten* im Anhang auf Seite viii).

Die Daten werden in der JS-Datei für die einzelnen Webelemente entsprechend verarbeitet und in eine HTML-Tabellenstruktur gebracht. Diese Tabellen werden dann jeweils über den Index des Elementes an die richtige Stelle eingebunden und kann auf der Webseite betrachtet werden (Siehe *Beispiel: Descriptions* im Anhang auf Seite ix).

Um den Portstatus in den Portlayouts farblich darzustellen, werden vier CSS-Klassen mit verschiedenen Farben erstellt. Die Namen der Klassen entsprechen der Kleinschreibung der möglichen Portzustände. Die Zuweisung der Klassen erfolgt über eine JS-Funktion, die die Zustände den Ports zuweist (Siehe *Portstatus farblich darstellen* im Anhang auf Seite ix).

3.2.3 PHP-Formulare

Es gibt zwei Formulare, die Anpassungen am Switch ausführen können. Das eine ermöglicht die Anpassung von Portbeschreibung und Admin-Status des Portes, das andere hat zwei Darstellungen (Siehe *PHP-Formulare* im Anhang auf Seite x).

Die erste Darstellung kann neue Port-Verschaltungen erstellen. Die Auswahl der Ports findet über Dropdown-Elemente statt. Nach der Auswahl wird auf Verbindungen hingewiesen, die durch eine Erstellung der neuen Verschaltung überschrieben werden. Beim Erstellen einer neuen Verschaltung, werden überschriebene Verbindungen komplett gelöscht, um einseitige Verschaltungen zu verhindern. Die zweite Ansicht kann Port-Verschaltungen löschen und zeigt die Ports der zu löschende Verschaltung an und erwartet eine Löschbestätigung.

Beide Formulare erscheinen nur durch Klick auf dafür bestimmte Textelemente.

Wenn ein Formular abgeschickt wird, übermittelt es die Formulardaten über POST und ein für dieses Formular einzigartigen GET-Parameter über die URL. Dieser GET-Parameter ist entscheidend für den Typ der Konfigurationsanpassung. Wird *change_port* als Parameter genutzt, werden mit den POST-Daten Portdetails angepasst. Wird *change_connection* verwendet, wird mit den POST-Daten eine Verbindung erstellt oder gelöscht.

Nachdem die Anpassungen an die API gesendet wurden, wird die PHP-Seite ohne den GET-Parameter in der URL neugeladen.

3.2.4 Element Header

Im Element Header wird die Bezeichnung des Switches ausgegeben. Der Stand der Daten wird mit dem Serverzeitpunkt, zu dem die PHP-Seite ausgeführt wird, datiert. (Siehe *Header* im Anhang auf Seite xii).

3.2.5 Sektion Legende

In der Sektion Legende wird die farbliche Gestaltung der Ports beziehungsweise der Portstatus erklärt. So steht grün für *up*, orange für *not present*, rot für *down* und dunkelrot für *adminDown* (Siehe *Legende* im Anhang auf Seite xii).

3.2.6 Sektion Ethernet Ports

In der Sektion Ethernet Ports werden die Ethernet Ports, die nicht für das Management sind, graphisch präsentiert. Das Layout und die Beschriftung der Ports wird in PHP mit einer Funktion gestaltet und ähnelt dem physikalischen Patchfeld des Switches. Mit einem Klick auf einen Port, werden relevante Port- und SFP-Details angezeigt (Siehe *Ethernet Ports* im Anhang auf Seite xiii).

3.2.7 Sektion Management Ports

In der Sektion Management Ports werden die Management Ports des Switches anklickbar, wie in der *Sektion Ethernet Ports*, dargestellt. Das Layout der Ports wird über die im vorherigen Kapitel *Sektion Ethernet Ports* erwähnte Funktion erstellt (Siehe *Management Ports* im Anhang auf Seite xiii).

3.2.8 Sektion Descriptions

In der Sektion Descriptions wird eine Liste mit Portdaten aufgeführt. So wird in jeder Zeile ein Port mit seiner Beschreibung (Description), seinem Admin-Status und seinem Protokoll-Status angezeigt. Jeder Listeneintrag besitzt ein Änderungsicon, welches durch einen Klick darauf das Formular für Portänderungen einblendet (Siehe *Descriptions* im Anhang auf Seite xiv).

3.2.9 Sektion Verschaltungen

In der Sektion Verschaltungen wird die Port-Verschaltungen aufgelistet. Dabei werden jeweils die beiden verschalteten Ports und die Beziehung (uni- oder bidirektional) zueinander angezeigt. Aktive Verschaltungen werden in grüner Farbe und einseitige Verschaltungen, die ungewollt sind, mit einer dunkelroten Farbe hinterlegt gekennzeichnet. Zudem hat jede Verschaltung am rechten Rand ein x zum Löschen der Verschaltung. Zum Erstellen einer neuen Port-Verschaltung kann der Text *Add Connection* am Anfang der Liste geklickt werden, wodurch das entsprechende Formular eingeblendet wird (Siehe *Verschaltungen* im Anhang auf Seite xv).

3.2.10 Sektion Komponenten

In der Sektion Komponenten werden Details zu Systemkomponenten, wie Energieversorgung, Speicher, Kühlung und Ventilatoren, abgebildet. Dabei sind die Details der einzelnen Komponenten aus- und einklappbar (Siehe *Komponenten* im Anhang auf Seite xv).

3.2.11 Sektion System

In der Sektion System werden Systemdetails, wie Systembezeichnung, Hersteller und Seriennummer, präsentiert (Siehe *System* im Anhang auf Seite xv).

3.2.12 Sektion Temperaturen

In der Sektion Temperaturen werden zum Beispiel die Temperatur von CPUs angezeigt. Es wird nicht nur die aktuelle Temperatur, aber auch Schwellenwerte, bei denen Alarmer ausgelöst werden, dargestellt (Siehe *Temperaturen* im Anhang auf Seite xvi).

3.2.13 Sektion Alarmer

In der Sektion Alarmer werden Port-Meldungen, wie Alarmer, Warnungen und Signalverlust, abgebildet (Siehe *Alarmer* im Anhang auf Seite xvi).

3.2.14 Element Tooltip

Das Element Tooltip wird mehrfach verwendet und zeigt Informationstexte durch die Aktion Mausover auf das Infosymbol an. Diese Texte fördern den Umgang mit der Webseite, indem die Bedienung der Funktionen erklärt werden (Siehe *Tooltip* im Anhang auf Seite xvi).

3.3 Integration der Lösung in die bestehende IT-Landschaft

Nachdem festgestellt wurde, dass die GUI-Lösung auf dem lokalen Testserver vollfunktionsfähig ist, kann die Integration in die Backbone-Tools beginnen.

Um die Quelldateien der GUI auf den Webserver, auf dem auch die Backbone-Tools ausgeführt werden, zu übertragen, wird *VSCodium* genutzt. Dafür wird die Erweiterung *Open Remote - SSH* von *jeanp413* installiert, was den Zugriff auf den Server über SSH ermöglicht. Nach der Installation der Erweiterung muss nur noch der Server mit Daten wie Hostname, IP und Nutzerkonto in die SSH-Konfiguration eingetragen werden und eine Verbindung zum Server gestartet werden (Siehe *VSCodium: Zugang zum Server* im Anhang auf Seite xvii).

Besteht die Verbindung zum Server, so kann auf dessen Verzeichnisstruktur zugegriffen werden und das richtige Arbeitsverzeichnis ausgewählt werden. Im ausgewählten Verzeichnis werden die drei Quelldateien – *layer_1_control.php*, *layer_1.js* und *layer_1.css* – über *VSCodium* angelegt und übertragen (Siehe *VSCodium: übertragenen Dateien* im Anhang auf Seite xvii).

Um die Weboberfläche auch als Unterseite in die Backbone-Tools zu integrieren, muss der Pfad der PHP-Datei der Abteilung BB übermittelt werden. Dieser Pfad wird von der Abteilung BB in die Datenbank, die für Verwaltung der Unterseiten der Backbone-Tools zuständig ist, überführt und die GUI in das Menü der Backbone-Tools einpflegt. Des Weiteren wird die Content-Nummer der GUI zurückübertragen, da die Backbone-Tools-Seiten nur über die entsprechende Content-Nummer anstelle des Pfades in der URL erreichbar sind.

Da der entwickelte PHP-Code eher pfadbasierend ist, muss dieser angepasst werden. Die Anpassungen sind gering, so muss bei URL-Aufrufen wie bei der Seiten-Umleitung und in Formularaktionen der GET-Parameter *content* mit der erhaltenen Content-Nummer hinzugefügt werden. Auch müssen die Verlinkungspfade der anderen Quelldateien um den Elternordner erweitert werden, damit diese auch geladen werden (Siehe *VSCodium: PHP-Code Anpassung* im Anhang auf Seite xviii).

Nach der Anpassung konnte festgestellt werden, dass die Verwaltungsoberfläche über die Backbone-Tools erreichbar ist und vollfunktionsfähig ist (Siehe *Integrierte Verwaltungsoberfläche* im Anhang auf Seite xix). Um Änderungen am Code zu erschweren, werden die Dateirechte der Quelldateien auf das Minimum reduziert. So erhielten die Dateien nur Leserechte für Alle (Siehe *Dateirechte anpassen* im Anhang auf Seite xviii).

3.4 Sind die Anforderungen erfüllt?

Nachdem die Lösung verwirklicht wurde, sollte mit Hilfe der Anforderungsliste kontrolliert werden, ob die gewünschten Vorgaben erfüllt sind.

- ☒ Die Realisierung soll graphisch sein

Die Realisierung ist durch die Web-Form der Verwaltungsoberfläche und Nutzung von CSS graphisch.

- ☒ Die Realisierung soll nicht auf CLI basieren

Es wird die CLI des Switches nicht für den Datenzugriff genutzt und ist nicht in die Lösung eingebaut.

- ☒ Die Realisierung soll von Personen ohne Kenntnisse über den L1S bedienbar sein

Nutzer benötigen zum Bedienen der Lösung keine besonderen Kenntnisse über den L1S. Tooltips vereinfachen die Nutzung der Webseite.

- ☒ Die Realisierung soll aktive Verschaltungen anzeigen

Die aktiven Verschaltungen werden in der Liste der Sektion Verschaltungen in grün angezeigt.

- ☒ Die Realisierung soll Betriebsparameter anzeigen

Die Betriebsparameter werden in den verschiedenen Sektionen angezeigt.

- ☒ Die Realisierung soll Portzustände und Portbeschreibungen ändern

In Sektion Descriptions können Portdetails wie Portbeschriftung und Admin-Status verändert werden.

- ☒ Die Realisierung soll Port-Verschaltungen anpassen

Durch die Lösung können in Sektion Verschaltungen Port-Verschaltungen erstellt und gelöscht werden.

- ☒ Die Realisierung soll nur bidirektionale Verbindungen ermöglichen

Auf der Webseite unter Sektion Verschaltungen können nur zweiseitige Verbindungen erstellt werden. Beim Verschalten werden überschriebene Verschaltungen komplett gelöscht, was durch die Lösung verursachte unidirektionale Verbindungen verhindert.

- ☒ Die Realisierung soll bei der Bedienung keine CLI-Anmeldedaten nutzen

Der Nutzer benötigt auf der Webseite keine CLI-Anmeldedaten, diese sind nur im Quellcode nötig.

- ☒ Die Realisierung soll in die bestehende IT-Landschaft integriert werden

Die Verwaltungsoberfläche wurde erfolgreich in die Backbone-Tools integriert.

- ☒ Die Realisierung soll eine Anmeldung über das bestehende Rollen- und Berechtigungsmanagement ermöglichen

Durch die Integration in Backbone-Tools ist die Anmeldung über Backbone-Tools gegeben.

Somit sind alle Vorgaben eingehalten worden. Da sich die Lösung im Admin-Netz befindet, wurden auch die Einschränkungen eingehalten.

4 Testen und Fehlerprüfung

Das Testen sowie die Fehlerprüfung fanden größtenteils während der Implementierung von L1GUI statt. Nach der Integration wurden alle Funktionen ein zweites Mal überprüft.

4.1 Überprüfung der Funktionsfähigkeit der Lösung

Es wurden mehrere Aspekte überprüft, um sicherzustellen, dass die Projektrealisierung voll funktionstüchtig ist.

Bei der graphischen Portstatus-Anzeige wurde untersucht, ob die angezeigten Farben mit den richtigen Link-Status beziehungsweise Admin-Status korrespondieren und ob die relevanten Port- und Optikdetails durch den Klick auf den gewollten Port angezeigt werden.

Des Weiteren wurde beurteilt, ob die Listen über die Portdaten mit Description, Admin-Status und Protokoll-Status sowie den Verschaltungen im beabsichtigten Zustand dargestellt werden. Wobei bei den Verschaltungen kontrolliert wurde, ob die aktiven Verschaltungen und die Beziehung der Verschaltungen angezeigt werden.

Für die Verwaltung des L1S war es wichtig, dass die Popup-Formulare für Ports und Verschaltungen nur durch das Klicken auf die dafür vorgesehenen Schaltflächen ein- und ausblendbar sind.

Zudem wurde erprobt, ob das Ändern von Portbeschreibung (Description) und -Admin-Status möglich ist, und Verschaltungen gelöscht sowie Verschaltungen angelegt werden können.

Zuletzt wurde getestet, ob die Tooltips durch Mausover eingeblendet und richtig angezeigt werden. Abschließend wurde geprüft, ob die obigen Punkte auch nach der Integration in die bestehende IT-Landschaft bestehen.

5 Dokumentation

Um die Herangehensweise bei der Entwicklung, Umsetzung sowie die Anwendung der Projektlösung entsprechend festzuhalten, wurden eine Betriebsdokumentation und eine Anwenderdokumentation erstellt. Teile beider Dokumentationen wurden in diese Dokumentation integriert, und sollen das Verstehen des Projektvorgehens vereinfachen. Daher wurde auf eine separate Einpflegung der Dokumentationen verzichtet.

5.1 Anfertigung der Betriebsdokumentation

In der Betriebsdokumentation wurden verschiedene für die GUI-Realisierung notwendige Projektdetails festgehalten. Darunter das Projektproblem, der Ist-Zustand, der Soll-Zustand beziehungsweise die Anforderungsliste sowie die Umsetzung der Lösung mit Konfiguration, Code und Integration in die bestehende IT-Landschaft. Das beinhaltet auch relevante Anmeldedaten.

Die Dokumentation soll eine Neuaufsetzung oder Anpassung der L1S-Weboberfläche ermöglichen.

5.2 Anfertigung der Anwenderdokumentation

Für die Anwender wurde eine textgestützte Bilderdokumentation erstellt. Da die Funktionen der Weboberfläche durch die integrierten Tooltips erklärt werden, reicht diese Art von Dokumentation für Nutzer vollkommen aus.

5.3 Einweisen der Administratoren und Anwender in die Lösung

Innerhalb einer Schulung wurden die Administratoren und Anwender über die Funktionen und Anwendung der L1S-GUI informiert. Dabei wurde auf die Dokumentationen hingewiesen.

6 Abschluss

Zum Schluss muss noch überprüft werden, ob die Projektziele eingehalten wurden, aber auch, ob sich das Projekt rentieren wird. Zu diesem Zweck wird ein Soll-Ist-Vergleich und eine Wirtschaftlichkeitsanalyse durchgeführt.

6.1 Wirtschaftlichkeitsanalyse

Da, mit der Nutzung des L1S, nicht direkt Gewinn erwirtschaftet wird, ist die genaue Wirtschaftlichkeit des Projektes schwer zu ermitteln.

Da die Amortisation von diversen Faktoren abhängt, wird diese anhand von mehreren Beispielszenarios berechnet. Faktoren sind dabei: die stündlichen Kosten durch die nutzenden Personen, die stündlichen Kosten durch den Mitarbeiter der Abteilung BB, der stattdessen die Änderung durchgeführt hätte, und die Nutzungshäufigkeit beziehungsweise die Änderungen am L1S pro Woche. Für diese Werte gibt es derzeit noch keine genaue Durchschnitte. Daher werden in den folgenden Absätzen diverse Annahmen getroffen, um eine vereinfachte Schätzung zu erhalten.

Vor dem Projekt haben L1S-unerfahrene Personen die Abteilung BB beauftragt, Änderungen am L1S über die CLI umzusetzen. Nun können diese Personen mit der Verwaltungswebseite selbst Anpassungen am L1S tätigen. Das spart Arbeitszeit durch den Wegfall der Kommunikation mit der Abteilung BB ein und führt zu einer schnelleren Durchführung der Anpassungen.

Dabei ist anzunehmen, dass das Durchführen von Änderungen über die CLI und Verwaltungsoberfläche ungefähr dieselbe Zeit in Anspruch nimmt. Was zur Folge hat, dass nur durch L1S-unerfahrene Personen Arbeitszeit eingespart werden kann, was zur Amortisation beiträgt. Daher bezieht sich der zu ermittelnde Break-Even-Point nur auf die Nutzung der Webseite durch L1S-unerfahrene Personen.

Bei der Ermittlung des Break-Even-Point werden die entstehenden Stromkosten der API und Webseite sowie die dabei entstehende Netzwerklast nicht miteinbezogen. Diese können vernachlässigt werden, da diese wesentlich kleiner als die Personalkosten sind und im Vergleich zur CLI kaum zusätzliche Kosten verursachen. Daher sind nur die ermittelten Personalkosten von 657,60 € Teil der Projektkosten. Diese Projektkosten werden sich mit der eingesparten Arbeitszeit durch den Wegfall der Kommunikation amortisieren, dabei werden der Einfachheit halber stündliche Kosten von 43,08 € (Mitarbeiter der Abt. BB) angenommen.

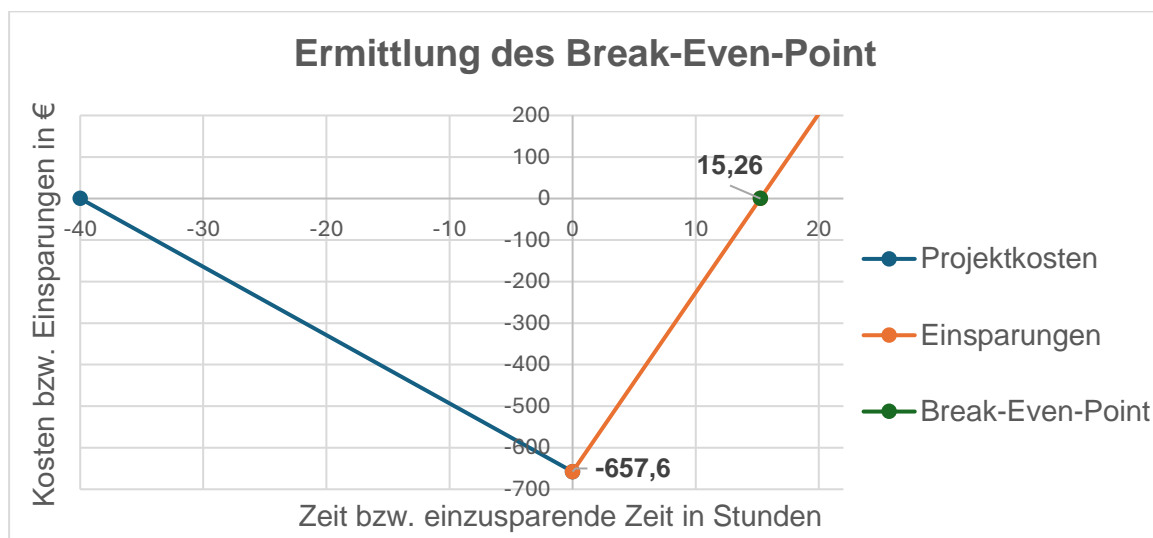


Abbildung 3: Zeit-Euro-Diagramm: Ermittlung des Break-Even-Point

So wären die Projektkosten nach 15,26 eingesparten Stunden wieder ausgeglichen. Mit 12 Minuten Einsparung pro Änderung entspräche, dass circa 77 Änderungen beziehungsweise mit durchschnittlich 3 Änderungen pro Woche, ungefähr ein halbes Jahr (Siehe *Ermittlung des Break-Even-Point* im Anhang auf Seite xx).

Da der Break-Even-Point beziehungsweise die Amortisation sich auf Annahmen stützt, ist davon auszugehen, dass dieser Wert stark von der Realität abweichen kann.

Auch wenn unklar ist wann sich das Projekt wirklich amortisieren wird, rentiert es sich, indem es die Abteilung BB durch weniger Anfragen entlastet und Konfigurationsfehler durch die Eingabebeschränkungen der Webseite unterbindet.

6.2 Soll-Ist-Vergleich

Nachdem die Wirtschaftlichkeitsanalyse gezeigt hat, dass sich das Projekt rentiert. Ist zu überprüfen, ob alle Projektziele beziehungsweise Anforderungen eingehalten worden sind. Diese fand bereits im Kapitel *Sind die Anforderungen erfüllt?* auf Seite 12 statt. Dabei konnte festgestellt werden, dass alle Anforderungen erfüllt sind.

Die zeitliche Planung konnte nur in Teilen eingehalten werden. So gab es ein paar zeitliche Abweichungen. Bei den Phasen *Vorbereitung und Planung* und *Abschluss* kam es insgesamt zu einer Abweichung von drei Stunden, die eingespart wurden. Diese flossen in die Phase *Verwirklichung der Lösung* für die Entwicklung der Lösung. Die festgesetzte Gesamtzeit von 40 Stunden hat sich dabei nicht geändert (Siehe *Tabelle 7: Soll-Ist-Zeitvergleich* im Anhang auf Seite xx).

6.3 Fazit und Ausblick

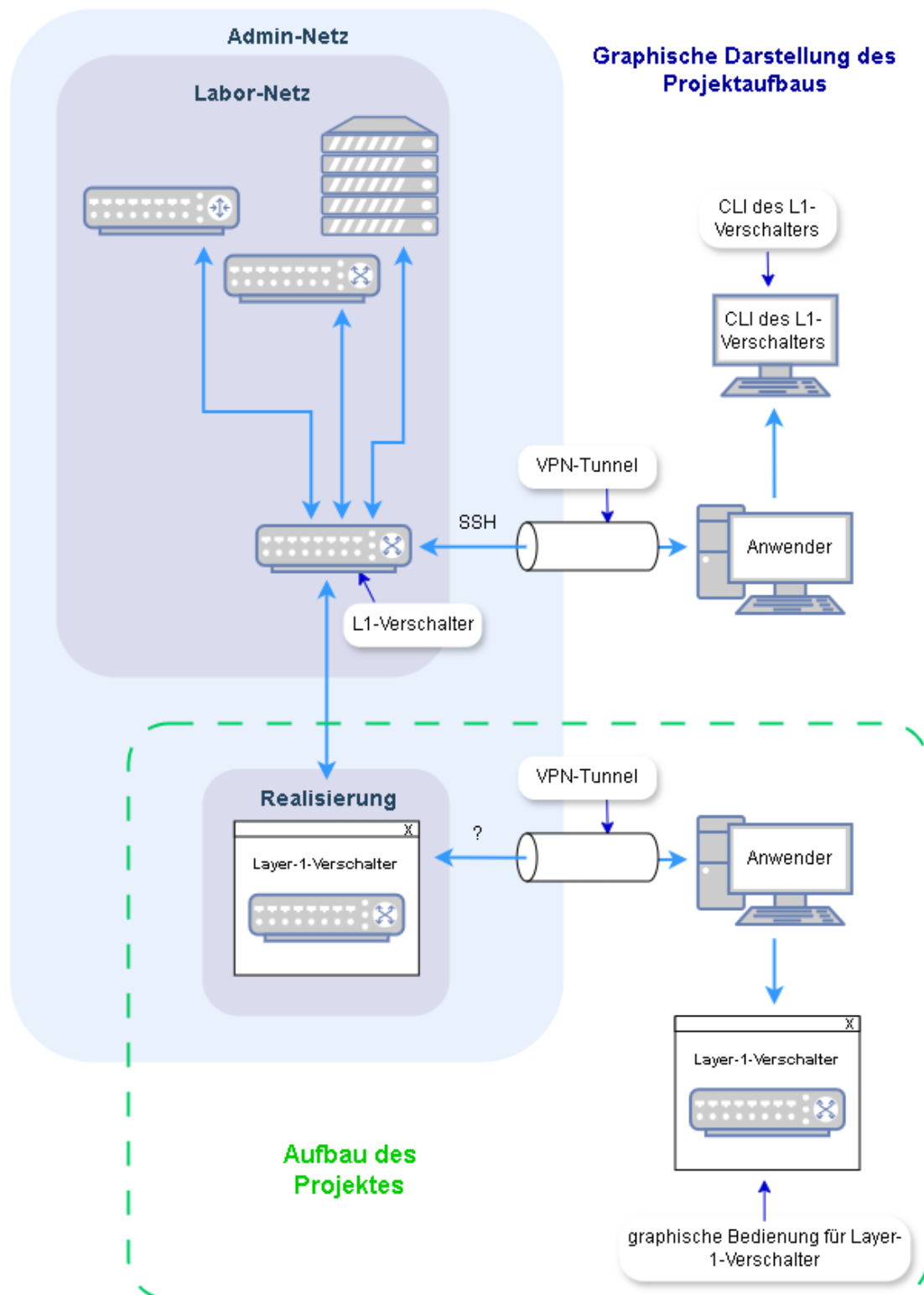
Das Projekt an sich lief gut ab und ermöglichte mir ein komplettes Projekt selbst durchzuführen. Normalerweise war ich nur mit der Entwicklung oder Umsetzung einer Lösung beschäftigt. Zudem lernte ich kennen, was ein L1S ist und wie eine Präferenzmatrix und Wirtschaftlichkeitsanalyse funktioniert.

Was nicht so ideal war, war, dass es keine Fertiglösung gab, die als Lösungsvorschlag genutzt werden hätte können. Auch die nur ungenaue und auf Annahmen basierende Ermittlung des Break-Even-Point der Wirtschaftlichkeitsanalyse war nicht zufriedenstellend.

Zusammenfassend wird die Verwaltungsoberfläche für den L1S erfolgreich in der Abteilung BB und den anderen Abteilungen verwendet. Dabei kamen bisher keine Probleme auf. Da die Nutzung der Oberfläche gut läuft, plant die Abteilung BB den Zugang auf die CLI des L1S so einzuschränken, dass nur noch Administratoren Zugriff auf diese haben. So sollen Anpassungen des L1S nur noch über die Webseite möglich sein.

Quellenverzeichnis

- P|API|1** Arista, „Arista eAPI,“ 2023. [Online]. Available: https://www.arista.com/assets/data/pdf/Whitepapers/Arista_eAPI_FINAL.pdf. [Zugriff am 07 03 2025].
- P|EWE|1** M. Tienken, „Projektarbeit - TranSpot - Ein Langzeit Monitoring Solution zur Überwachung von Traffic-Fluss & -Menge in einem Providernetz,“ 11 05 2022. [Online]. Available: <https://s3.eu-central-1.amazonaws.com/fiae/Beispiele/ProjektdokumentationFachinformatikerSystemintegration2022MartinTienken.pdf>. [Zugriff am 03 03 2025].
- P|L1|g** Arista, „A layman's guide to Layer 1 Switching,“ 2018. [Online]. Available: <https://www.arista.com/assets/data/pdf/Whitepapers/Laymans-Guide-White-Paper.pdf>. [Zugriff am 03 03 2025].
- P|L1|d** Arista, „7130S Connect Series Layer 1 Switch - Data Sheet,“ 22 November 2024. [Online]. Available: <https://www.arista.com/assets/data/pdf/Datasheets/7130-Connect-S-Series-Datasheet.pdf>. [Zugriff am 03 03 2025].
- W|API|1** Arista, „NETCONF over SSH - Open Management,“ [Online]. Available: https://aristanetworks.github.io/openmgmt/examples/netconf/netconf_over_ssh/. [Zugriff am 07 03 2025].
- W|API|2** Arista, „RESTCONF with cURL - Open Management,“ [Online]. Available: <https://aristanetworks.github.io/openmgmt/examples/restconf/curl/>. [Zugriff am 07 03 2025].
- W|API|3** QuickHostUK, „How to enable eAPI – Arista – Knowledge Base,“ [Online]. Available: <https://help.quickhost.uk/index.php/knowledge-base/how-to-enable-eapi-arista/>. [Zugriff am 07 03 2025].
- W|EWE|1** LeadIQ, „EWE TEL GmbH Company Overview, Contact Details & Competitors | LeadIQ,“ [Online]. Available: <https://leadIQ.com/c/ewe-tel-gmbh/5a1d7dbe24000024005735e9>. [Zugriff am 03 03 2025].
- W|EWE|2** Windhoff Group, „EWE TEL | Windhoff Group,“ [Online]. Available: <https://windhoff-group.de/referenz/telekommunikation-und-medien/ewe-tel/>. [Zugriff am 03 03 2025].
- W|GUI|1** Verttigo28, „GitHub - Verttigo28/arista-switch-webui: This project is a web-based user interface designed to easily configure and manage Arista switches. The Web UI provides network administrators with a clear, intuitive interface to interact with their network device,“ 16 11 2024. [Online]. Available: <https://github.com/Verttigo28/arista-switch-webui?tab=readme-ov-file>. [Zugriff am 07 03 2025].
- W|GUI|2** YonatanPorat, „GitHub - YonatanPorat/Arista_WEB_GUI: Simple Web interface for Arista switches,“ 31 01 2022. [Online]. Available: https://github.com/YonatanPorat/Arista_WEB_GUI. [Zugriff am 07 03 2025].
- W|TAG|1** S. Banse, „Arbeitstage Niedersachsen 2025,“ 01 10 2024. [Online]. Available: https://www.smart-rechner.de/arbeitstage/infothek/arbeitstage_in_ni.php. [Zugriff am 10 03 2025].
- W|TAG|2** Statistisches Bundesamt, „Krankenstand - Statistisches Bundesamt,“ 2025. [Online]. Available: <https://www.destatis.de/DE/Themen/Arbeit/Arbeitsmarkt/Qualitaet-Arbeit/Dimension-2/krankenstand.html>. [Zugriff am 10 03 2025].
- W|PY|1** Arista, „Introducing the Python Client for eAPI (pyeapi),“ 26 01 2025. [Online]. Available: <https://arista.my.site.com/AristaCommunity/s/article/introducing-the-python-client-for-eapi-pyeapi>. [Zugriff am 07 03 2025].
- W|PY|2** J. Marshall, „jsonrpclib-pelix · PyPI,“ 03 03 2025. [Online]. Available: <https://pypi.org/project/jsonrpclib-pelix/>. [Zugriff am 07 03 2025].

A Anhang**A.1 Zielschema**

(Quelle: Eigene Darstellung, Draw.io)

Abbildung 4: Zielschema

A.2 Anforderungsliste

Die Realisierung soll ...

- ☐ graphisch sein
- ☐ nicht auf CLI basieren
- ☐ von Personen ohne Kenntnisse über den L1S bedienbar sein
- ☐ in die bestehende IT-Landschaft integriert werden
- ☐ aktive Verschaltungen anzeigen
- ☐ Betriebsparameter anzeigen
- ☐ Portzustände ändern
- ☐ Portbeschreibungen ändern
- ☐ Port-Verschaltungen anpassen
- ☐ nur bidirektionale Verbindungen ermöglichen
- ☐ bei der Bedienung keine CLI-Anmeldedaten nutzen
- ☐ eine Anmeldung über das bestehende Rollen- und Berechtigungsmanagement (RADIUS-Authentifizierung) ermöglichen

A.3 Berechnung der Personalkosten

	Azubi	Andere	Gesamt
Vorbereitung und Planung	13		13
Besprechung des Problems / Projektumfang	1		1
Erfassung des Ist-Zustand	1		1
Definition des Soll-Zustands	1		1
Erstellung der Lösungsvorschläge	3		3
Kostenvergleich mit Personalkostenberechnung	3		3
Anwendung einer Präferenzmatrix	2		2
Anwendung einer Nutzwertanalyse auf die Lösungen	2		2
Verwirklichung der Lösung	1	15	16
Konfiguration des Layer-1-Verschalters	0,5	0,5	1
Entwicklung der Lösung		14	14
Integration der Lösung in die bestehende IT-Landschaft	0,5	0,5	1
Testen und Fehlerprüfung	0,5	2,5	3
Überprüfung der Funktionsfähigkeit der Lösung	0,5	2,5	3
Abschluss	3	5	8
Anfertigung der Betriebsdokumentation	0,75	2,25	3
Anfertigung der Anwenderdokumentation		2	2
Einweisen der Administratoren und Anwender	0,25	0,75	1
Wirtschaftlichkeitsanalyse	2		2
Gesamtzeit	18	23	40

(in Excel erstellt)

Tabelle 3: Zeitliche Aufgabenaufteilung zwischen Azubi und Intern/Extern

$$\frac{39 \frac{\text{Stunden}}{\text{Woche}}}{5 \frac{\text{Arbeitstage}}{\text{Woche}}} = 7,8 \frac{\text{Stunden}}{\text{Arbeitstag}} \quad (1)$$

Jahr	Arbeitstage	Jahr	Krankheitstage
2018	251	2014	9,5
2019	251	2015	10,0
2020	255	2016	10,8
2021	256	2017	10,6
2022	253	2018	10,6
2023	251	2019	10,9
2024	252	2020	11,2
2025	251	2021	11,2
2026	254	2022	14,8
2027	256	2023	15,1
Ø	253	Ø	11,47

[W/TAG/1]

[W/TAG/2]

(erstellt in
Excel)

Tabelle 4: Durchschnittliche Arbeits- und Krankheitstage pro Jahr von Beschäftigten in Niedersachsen

$$253 \frac{\text{Arbeitstage}}{\text{Jahr}} - 30 \frac{\text{Urlaubstage}}{\text{Jahr}} - 11 \frac{\text{Krankheitstage}}{\text{Jahr}} = 212 \frac{\text{Arbeitstage}}{\text{Jahr}} \quad (2)$$

$$7,8 \frac{\text{Stunden}}{\text{Arbeitstag}} * 212 \frac{\text{Arbeitstage}}{\text{Jahr}} \approx 1.654 \frac{\text{Stunden}}{\text{Jahr}} \quad (3)$$

	%	Faktor	Kosten
Gehalt pro Monat			1.390,00 €
* Monate im Jahr		12	
= Gehalt pro Jahr			16.680,00 €
+ Urlaubsgeld			1.390,00 €
+ Weihnachtsgeld			1.390,00 €
= Gehaltskosten pro Jahr			19.460,00 €
+ Gemeinkostenzuschlagssatz	20,0%		3.892,00 €
+ Arbeitgeberanteil	19,7%		3.833,62 €
= Kosten pro Jahr für Azubi			27.185,62 €

(in Excel erstellt)

Tabelle 5: Berechnung der Jahreskosten durch einen Azubi

$$\frac{27.185,62 \frac{\text{€}}{\text{Jahr}}}{1.654 \frac{\text{Stunden}}{\text{Jahr}}} \approx 16,44 \frac{\text{€}}{\text{Stunde}} \quad (4)$$

	%	Faktor	Kosten
Gehalt pro Monat			4.250,00 €
* Monate im Jahr		12	
= Gehalt pro Jahr			51.000,00 €
+ Gemeinkostenzuschlagssatz	20,0%		10.200,00 €
+ Arbeitgeberanteil	19,7%		10.047,00 €
= Kosten pro Jahr für Angestellter der Abt. BB			71.247,00 €

(in Excel erstellt)

Tabelle 6: Berechnung der Jahreskosten durch einen Angestellten der Abteilung BB

$$\frac{71.247,00 \frac{\text{€}}{\text{Jahr}}}{1.654 \frac{\text{Stunden}}{\text{Jahr}}} \approx 43,08 \frac{\text{€}}{\text{Stunde}} \quad (5)$$

$$16,44 \frac{\text{€}}{\text{Stunde}} * 40 \text{ Stunden} = 657,60 \text{ € Pers. – K. Azubi} \quad (6)$$

$$16,44 \frac{\text{€}}{\text{Stunde}} * 18 \text{ Stunden} = 295,92 \text{ € Teilpers. – K. Azubi} \quad (7)$$

$$43,08 \frac{\text{€}}{\text{Stunde}} * 23 \text{ Stunden} = 990,84 \text{ € Teilpers. – K. Intern} \quad (8)$$

$$990,84 \text{ €} + 295,92 \text{ €} = 1.286,76 \text{ € Pers. – K. Intern} \quad (9)$$

$$50,00 \frac{\text{€}}{\text{Stunde}} * 23 \text{ Stunden} = 1.150 \text{ € Teilpers. – K. Extern} \quad (10)$$

$$1.150 \text{ €} + 295,92 \text{ €} = 1.445,92 \text{ € Pers. – K. Extern} \quad (11)$$

A.4 API-Konfiguration

API-Konfiguration

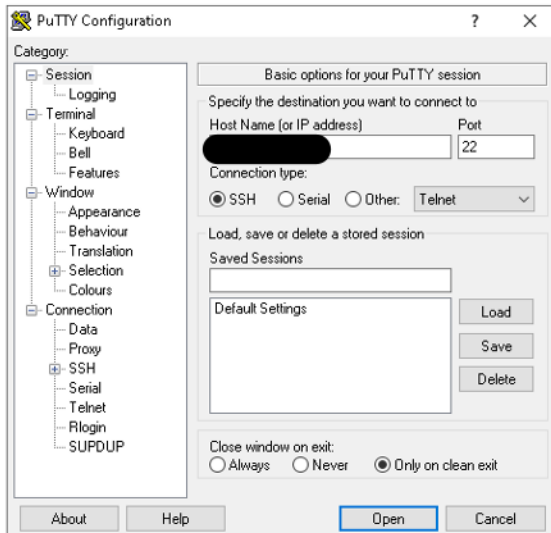


Abbildung 5: Zugriff auf den L1S mit PuTTY

```

L1CONNECT-90730699-LAB>enable
L1CONNECT-90730699-LAB#configure terminal
L1CONNECT-90730699-LAB(config)#management api http-commands
L1CONNECT-90730699-LAB(config-mgmt-api-http-cmds)#protocol https
L1CONNECT-90730699-LAB(config-mgmt-api-http-cmds)#no shutdown
L1CONNECT-90730699-LAB(config-mgmt-api-http-cmds)#end
L1CONNECT-90730699-LAB#wr
Copy completed successfully.

```

Abbildung 6: Aktivierung und Konfiguration der eAPI auf dem L1S

```

L1CONNECT-90730699-LAB#show management api http-commands
Enabled:                Yes
HTTPS server:           running, set to use port 443
HTTP server:            shutdown, set to use port 80
Local HTTP server:      shutdown, no authentication, set to use port 8080
Unix Socket server:     shutdown, no authentication
VRFs:                   default
Hits:                   3067
Last hit:                1329 seconds ago
Bytes in:                1364040
Bytes out:               366102688
Requests:               2468
Commands:               25393
Duration:               2265.810 seconds
SSL Profile:             none
FIPS Mode:              No
QoS DSCP:               0
Log Level:              none
CSP Frame Ancestor:     None
TLS Protocols:          1.0 1.1 1.2

```

User	Requests	Bytes in	Bytes out	Last hit
admin	2468	1364040	366102688	1329 seconds ago

```

URLs
-----
Management1 : https://[redacted]:443

```

Abbildung 7: Anzeige und Überprüfung des eAPI-Status auf dem L1S

A.5 Entwicklung der Verwaltungsoberfläche

PHP-Server und Quelldateien

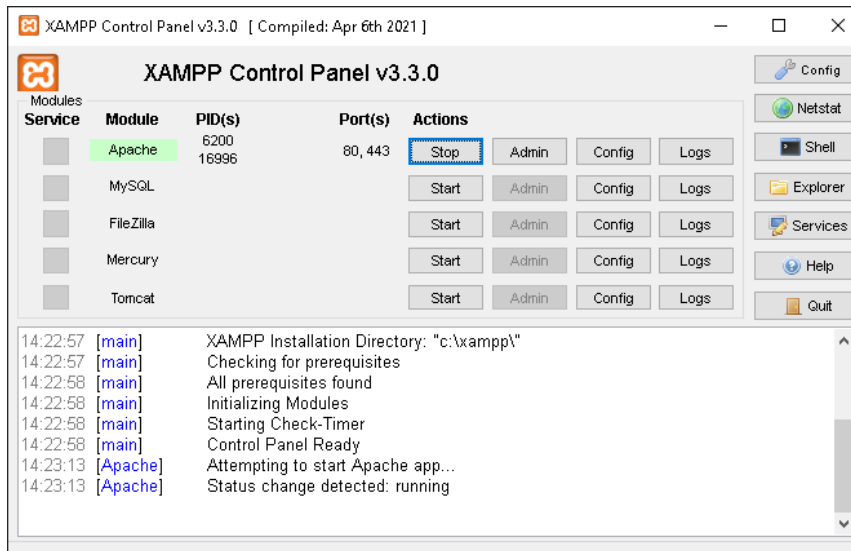


Abbildung 8: Start des PHP-Servers über das Programm XAMPP

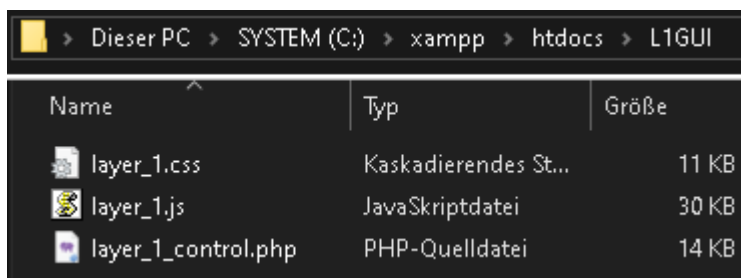


Abbildung 9: Lokal angelegte Quelldateien

Kommunikation mit der API

ARISTA
Command API

eAPI Explorer

Commands

Usage

Overview

On-Box Usage

Websockets

Cookies

Versioning

The Spec

Configuration

Overview

Server Certificates

Client Certificates

eAPI Explorer

This page lets you craft a single eAPI request, and explore the returned JSON. Note that this form creates real eAPI requests, so any configuration you perform will apply to this switch. Don't know where to start? Read the [API overview](#) or try one of these examples: [Check version](#) [Create an ACL](#)

[Show virtual router](#) or [View running-config](#)

Request Creator Request Viewer

API Endpoint:

https://.../command-api

{
 "cmd": "enable",
 "input": "my_enable_passwd"
}
show running-config

Version: 1

Format: json text

Timestamps: ☐

AutoComplete: ☐

ExpandAliases: ☐

StopOnError: ☒

Streaming: ☐

IncludeErrorDetail: ☐

ID: EapiExplorer-1

Submit POST request

eAPI Response:

Abbildung 10: Webseite der eAPI: Anfrage: show running-config

Request Creator Request Viewer

```
{  
  "jsonrpc": "2.0",  
  "method": "runCmds",  
  "params": {  
    "version": 1,  
    "cmds": [  
      {  
        "cmd": "enable",  
        "input": "my_enable_passwd"  
      },  
      "show running-config"  
    ],  
    "format": "json",  
    "timestamps": false,  
    "autoComplete": false,  
    "expandAliases": false,  
    "stopOnError": true,  
    "streaming": false,  
    "includeErrorDetail": false  
  },  
  "id": "EapiExplorer-1"  
}
```

Submit POST request

Abbildung 11: Webseite der eAPI: Anfrage mit JSON-Parametern: show running-config

```

<?php
...
// Sende Anfrage an L1-Verschalter, gib L1-Verschalter-Antwort zurück
function use_api($query) {
    // Wenn mehrere Befehle als Liste übermittelt werden
    if (is_array($query)) {
        $query_str = "";
        foreach ($query as $cmd) {
            $query_str .= ", \"$cmd\"";
        }
    }
    else $query_str = ", \"$query\"";
    $post = '{"jsonrpc": "2.0", "method": "runCmds", "params": {"version": 1, "cmds":
[{"cmd": "enable", "input": "my_enable_password"}$query_str], "format": "json",
"timestamps": false, "autoComplete": false, "expandAliases": false, "stopOnError": true,
"streaming": false, "includeErrorDetail": false}, "id": "EapiExplorer-1"}';

    $url = "https://[REDACTED]/command-api";
    $username = <user>;
    $password = <password>;

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post);
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_USERPWD, "$username:$password");
    curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
    curl_setopt($ch, CURLOPT_NOPROXY, '*'); // kein Proxy verwenden

    $output = curl_exec($ch);
    $info = curl_getinfo($ch);
    curl_close($ch);
    return json_decode($output, true);
}
...
?>

```

Abbildung 12: PHP: Funktion für API-Zugriff

```

<?php
...
$data = use_api(['show running-config', 'show interfaces description', 'show interfaces
transceiver', 'show version', 'show inventory', 'show transceiver status', 'show hostname',
'show system storage statistics', 'show environment power', 'show system environment cooling',
'show system environment temperature', 'show l1 path'])["result"];
...
$interface_descriptions = $data[2]["interfaceDescriptions"];
...
$hostname = $data[7]["hostname"];
...
?>

```

Abbildung 13: PHP: API-Daten mit CLI-Kommandos über Funktion holen und aufteilen (Auszug)

Verarbeitung und Darstellung der Daten

```

<script>
...
let interface_descriptions = <?php echo json_encode($interface_descriptions) ?>;
...
</script>

```

Abbildung 14: PHP: Daten als JavaScript-Variable speichern: Beispiel Interface Descriptions

Beispiel: Descriptions

```
function use_desc_data(data) {
  infos = "<table class='table'><tr><th>Port</th><th>Description</th><th>Admin-Status</th><th>Protokoll</th><th>Change?</th></tr>";
  for (let port in data) {
    let desc = data[port]["description"];
    let admin_status = data[port]["interfaceStatus"];
    let oper_status = data[port]["lineProtocolStatus"];

    infos += fill_description_list(port, desc, admin_status, oper_status);
    display_port_status(port, oper_status, admin_status);
  }
  infos += "</table>";
  return infos;
}

function fill_description_list(port, desc, admin_status, oper_status) {
  return "<tr><td>" + port + "</td><td>" + desc + "</td><td>" + admin_status + "</td><td>" + oper_status + "</td><td>"
    + "class='change_port_data svgchange'></td></tr>";
}

document.querySelector('div#descriptions').innerHTML = use_desc_data(interface_descriptions);
```

Abbildung 15: JS: Anhängen der verarbeiteten Daten an Dokumentelement für Sektion Descriptions

Portstatus farblich darstellen

```
#start td.up {
  background-color: lime;
}

#start td.down {
  background-color: red;
}

#start td.notpresent {
  background-color: orange;
}

#start td.admindown {
  background-color: rgb(101, 1, 1);
}
```

Abbildung 16: CSS: Stile der Klassen, die für die farbliche Darstellung des Portstatus nötig sind

```
function display_port_status(port, oper_status, admin_status) {
  try {
    document.getElementById(port).classList.add(oper_status.toLowerCase());
    document.getElementById(port).classList.add(admin_status.toLowerCase());
  } catch (err) {}
}
```

Abbildung 17: JS: Portstatus zuweisen über gleichnamige Klassen

PHP-Formulare

The screenshot shows a dark gray popup window titled 'Ethernet2' with an information icon. It contains two checkboxes: 'Change Description to' followed by a text input field, and 'Change Admin-Status to' followed by a dropdown menu currently set to 'adminDown'. At the bottom is a yellow button labeled 'Change Port Data'.

Abbildung 18: Popup-Formular für das Anpassen der Porteigenschaften

The screenshot shows a dark gray popup window titled 'Add Connection' with an information icon. It features two dropdown menus for selecting ports, currently showing 'Ethernet4' and 'Ethernet5', with a double-headed arrow between them. Below these is a section titled 'Verbindungen, die überschrieben werden:' (Connections to be overwritten:) showing two rows of connections: 'Ethernet4' to 'Ethernet45' and 'Ethernet5' to 'Ethernet14'. At the bottom is a yellow button labeled 'Add Connection'.

Abbildung 19: Popup-Formular für das Erstellen von Verschaltungen

The screenshot shows a dark gray popup window titled 'Delete Connection' with an information icon. It features two dropdown menus for selecting ports, currently showing 'Ethernet2' and 'Ethernet46', with a double-headed arrow between them. At the bottom is a yellow button labeled 'Delete Connection'.

Abbildung 20: Popup-Formular für das Löschen von Verschaltungen

```

<?php
...
function create_change_form() {
    return
    "<div id='overlay' class='overlay hide'>
    <div id='change_form' class='change_form shadow'>
    <form action='".$_SERVER['PHP_SELF']."'?content=106&change_port=true' method='POST'>
        <div class='flexbox'>
            <input name='input_port' id='input_port' value='NaN' placeholder='Port' readonly>
            <span class='info'><span>Um Description oder Up-Status zu verändern, muss die jeweilige Checkbox angehakt
            sein. Der Standardwert des Description-Feldes ist die bisherige Description.\n\nVorsicht! Längere
            Descriptions werden teilweise vom Inputfeld ausgeblendet.\n\nVerlassen durch Klick außerhalb des Feldes.</
            span></span>
        </div>
        <input name='is_change_desc' id='is_change_desc' type='checkbox'>
        <label for='input_desc'>Change Description to </label>
        <input name='input_desc' id='input_desc' value='NaN' placeholder='Description'>
        <br>
        <input name='is_change_status' id='is_change_status' type='checkbox'>
        <label id='label_status' for='input_status'>Change Admin-Status to </label>
        <input name='input_status' id='input_status' value='NaN' readonly>
        <br>
        <button type='submit'>Change Port Data</button>
    </form>
    </div>
    </div>";
}

function create_change_form_l1_connections() {
    return
    "<div id='l1_overlay' class='overlay hide'>
    <div id='l1_change_form' class='change_form shadow'>
    <form action='".$_SERVER['PHP_SELF']."'?content=106&change_connection=true' method='POST'>
        <div>
            <select name='l1_port' id='l1_port'></select>
        </div>
        <label id='l1_change_form_arrow' for='l1'>&#8597;</label>
        <div>
            <select name='l1_source' id='l1_source'></select>
        </div>
        <div id='info_form_l1_connections'></div>
        <input name='delete' id='delete' type='checkbox' class='hide'>
        <input name='l1_bidirection' id='l1_bidirection' type='checkbox' class='hide'>
        <input name='l1_reset_port1' id='l1_reset_port1' type='text' class='hide'>
        <input name='l1_reset_port2' id='l1_reset_port2' type='text' class='hide'>
        <div class='flexbox'>
            <button id='l1_submit_button' type='submit' onclick='enable_select_elements()'>Add Connection</button>
            <span class='info'><span>Hier können neue L1-Verschaltungen angelegt werden.\n\nDie Reihenfolge der Ports
            ist egal, da die Verbindung von beiden Seiten aus erstellt wird.\n\nVorsicht! Bestehende Verschaltungen
            werden ohne weitere Warnung überschrieben.\n\nVerlassen durch Klick außerhalb des Feldes.</span></span>
        </div>
    </form>
    </div>
    </div>";
}
...
?>

```

Abbildung 21: PHP: HTML-Code für die Formulare

```

function add_options_to_verschaltung_form(interfaces) {
    for (let port in interfaces) {
        if (port != 'Management1' && port != 'Management2') {
            document.querySelector('#l1_port').innerHTML += "<option value='" + port + "'>" + port + " - " + interfaces[port].description + "</option>";
            document.querySelector('#l1_source').innerHTML += "<option value='" + port + "'>" + port + " - " + interfaces[port].description + "</option>";
        }
    }
}

add_options_to_verschaltung_form(interface_descriptions);

```

Abbildung 22: JS: Hinzufügen der Auswahlmöglichkeiten bei Verschaltungsformular

```

change_port_data = document.querySelectorAll('.change_port_data');
change_port_data.forEach(elm => { elm.addEventListener('click', (e) => {
    document.querySelector('#input_port').value = e.target.parentNode.children[0].innerHTML;
    document.querySelector('#input_desc').value = e.target.parentNode.children[1].innerHTML;
    document.querySelector('#input_status').value = (e.target.parentNode.children[2].innerHTML == "adminDown") ? "up" : "adminDown";

    document.querySelector('#overlay').classList.remove('hide');
}}});

```

Abbildung 23: JS: Anzeigen der Daten aus Sektion Descriptions für Portanpassungsformular

```
<?php
if ((isset($_GET['change_port']) && $_GET['change_port'] == true) || (isset($_GET['change_connection']) && $_GET
['change_connection'] == true)){
    if (isset($_POST['input_port']) && (isset($_POST['is_change_desc']) || isset($_POST['is_change_status']))) {
        change_port_data($_POST['input_port'], $_POST['is_change_desc'] ?? false, $_POST['input_desc'], $_POST
['is_change_status'] ?? false, $_POST['input_status']);
    }
    else if (isset($_POST['l1_port'])) {
        change_l1_connection($_POST['l1_port'], $_POST['l1_source'], $_POST['delete'] ?? false, $_POST
['l1_bidirection'] ?? false, $_POST['l1_reset_port1'] ?? false, $_POST['l1_reset_port2'] ?? false);
    }
    echo "<script>window.location.replace('".$_SERVER['PHP_SELF']."'."?content=106');</script>";
    die();
}
...
function change_port_data($port, $is_change_desc, $desc, $is_change_status, $status) {
    if ($is_change_desc || $is_change_status) {
        if ($is_change_desc) $desc = ($desc) ? "description ".str_replace("'", '\'', $desc) : "no description";
        $status = ($status == "up") ? 'no shutdown' : 'shutdown';
        if ($is_change_desc == false) $query = ["configure", "interface $port", $status];
        else if ($is_change_status == false) $query = ["configure", "interface $port", $desc];
        else $query = ["configure", "interface $port", $desc, $status];
        use_api($query);
    }
}

function change_l1_connection($port, $source, $delete, $bidirection, $reset_port1, $reset_port2) {
    // Delete
    if ($delete) {
        $query = ["configure", "interface $port", "no l1 source interface"];
        if ($bidirection) array_push($query, "interface $source", "no l1 source interface");
    }
    // Add
    else {
        $query = ["configure", "l1 connect $port $source"];
        // Reset ports of now incomplete connections
        if ($reset_port1) array_push($query, "interface $reset_port1", "no l1 source interface");
        if ($reset_port2) array_push($query, "interface $reset_port2", "no l1 source interface");
    }
    use_api($query);
}
...
?>
```

Abbildung 24: PHP: Logik und Funktionen für die Formulare

Header

L1CONNECT-90730699-LAB Stand: 12.03.2025 - 12:26

Abbildung 25: Header mit Zeit

```
<?php
...
echo "<div id='start'><div id='hostname'><h1>$hostname <span>Stand: ".date("d.m.Y - H:i", time())."</span></h1></div>";
...
?>
```

Abbildung 26: PHP: HTML-Code für Header

Legende

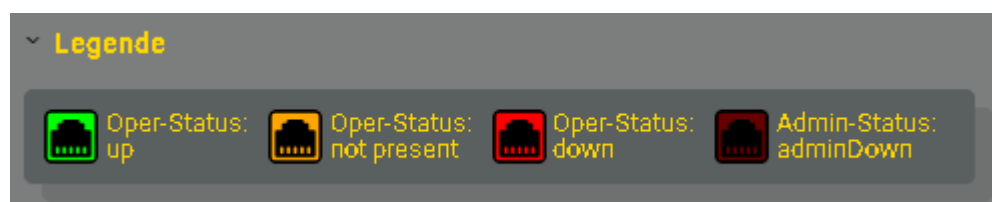


Abbildung 27: Sektion Legende: Legende für Portstatus

Management Ports



Abbildung 28: Sektion Management Ports

Ethernet Ports

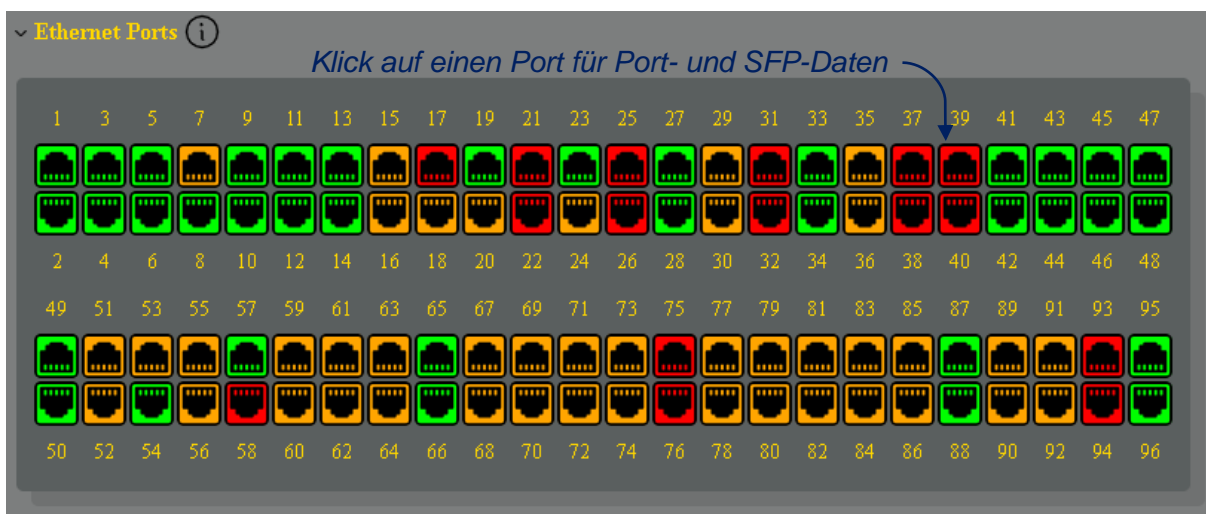


Abbildung 29: Sektion Ethernet Ports

```
<?php
...
function create_port_layout($what = 'Ethernet', $ports_count = 96, $rows = 2, $row_double = true, $port_nums = true) {
    $step = $row_double ? 2 : 1;

    echo "<table class='table' id='layout_$what'>";
    for ($z = 0; $z < $rows; $z++) {
        for ($y = 0; $y < $rows; $y++) {
            echo "<tr>";
            if ($port_nums && $y % 2 == 0) {
                for ($x = 1+$y+$z*($ports_count/$rows); $x <= $ports_count/($rows-$z); $x += $step) {
                    echo "<td class='port_num'>$x</td>";
                }
                echo "</tr><tr>";
            }
            for ($x = 1+$y+$z*($ports_count/$rows); $x <= $ports_count/($rows-$z); $x += $step) {
                echo "<td class='port noborder' id='$what$x'></td>";
            }
            if ($port_nums && $y % 2 != 0) {
                echo "</tr><tr>";
                for ($x = 1+$y+$z*($ports_count/$rows); $x <= $ports_count/($rows-$z); $x += $step) {
                    echo "<td class='port_num'>$x</td>";
                }
            }
            echo "</tr>";
            if (!$row_double) break;
        }
    }
    echo "</table>";
}
...
?>
```

Abbildung 30: PHP: Funktion für Port-Layout



Abbildung 31: Popup: Port- und SFP-Daten

Descriptions



Abbildung 32: Sektion Descriptions: Liste der Ports mit Eigenschaften (Auszug)

Verschaltungen

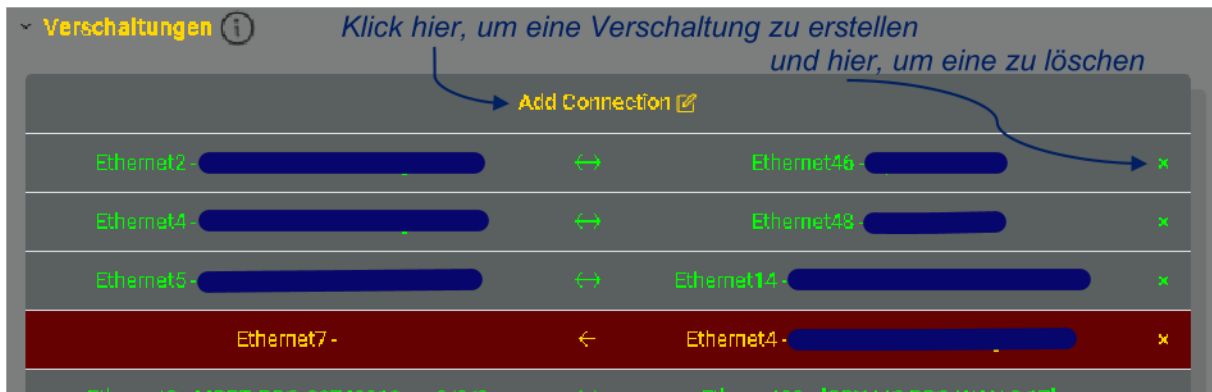


Abbildung 33: Sektion Verschaltungen: Liste der Verschaltungen (Auszug)

Komponenten

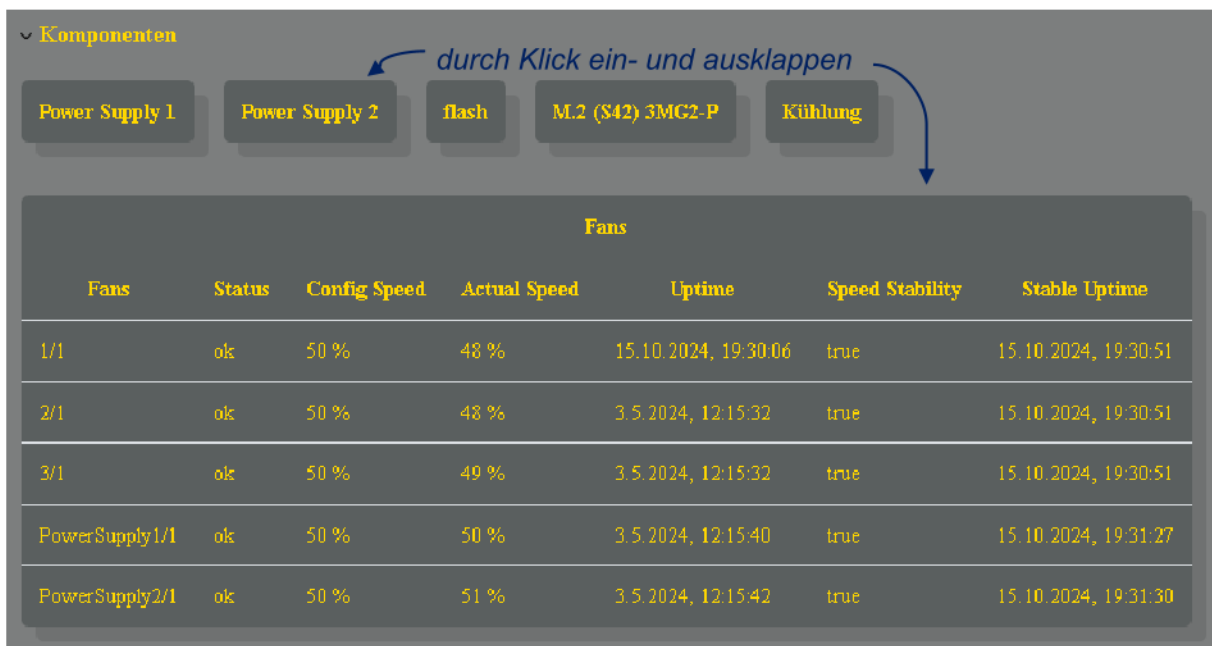


Abbildung 34: Sektion Komponenten: Nur Komponente Fans aufgeklappt

System



Abbildung 35: Sektion System (Auszug)

Temperaturen

~ Temperaturen

Status: temperatureOk

Sensor	Description	Temp (°C)	Setpoint (°C)	Alert Limit (°C)	Critical Limit (°C)
1	CPU0 temperature	29	(75) 77	90	95
2	CPU1 temperature	29	(75) 77	90	95
3	CPU2 temperature	29	(75) 77	90	95
4	CPU3 temperature	29	(75) 77	90	95

Abbildung 36: Sektion Temperaturen (Auszug)

Alarmer

~ Alarmer

Port 21	Port 22	Port 25	Port 26	Port 31
rxLos true	rxLos true	rxLos true	rxLos true	rxLos true
rxPowerLoAlarm true	rxPowerLoAlarm true			rxPowerLoAlarm true
rxPowerLoWarn true	rxPowerLoWarn true			rxPowerLoWarn true

Abbildung 37: Sektion: Alarmer (Auszug)

Tooltip

~ Ethernet Ports ⓘ

Portinformationen werden durch einem Klick auf den Port angezeigt. Mit einem Klick außerhalb der Infos werden die Infos ausgeblendet.

1 3 5 7 9 11 13 15 17 19 21 23 25

Abbildung 38: Tooltip: Tooltip zu Portdetails anzeigen und ausblenden

A.6 Integration mit VSCodium und PuTTY

VSCodium: Zugang zum Server

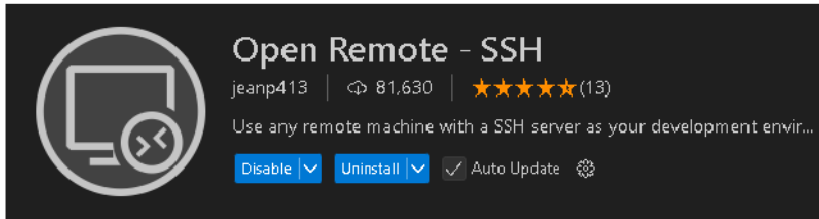


Abbildung 39: Installierte SSH-Erweiterung für VSCodium

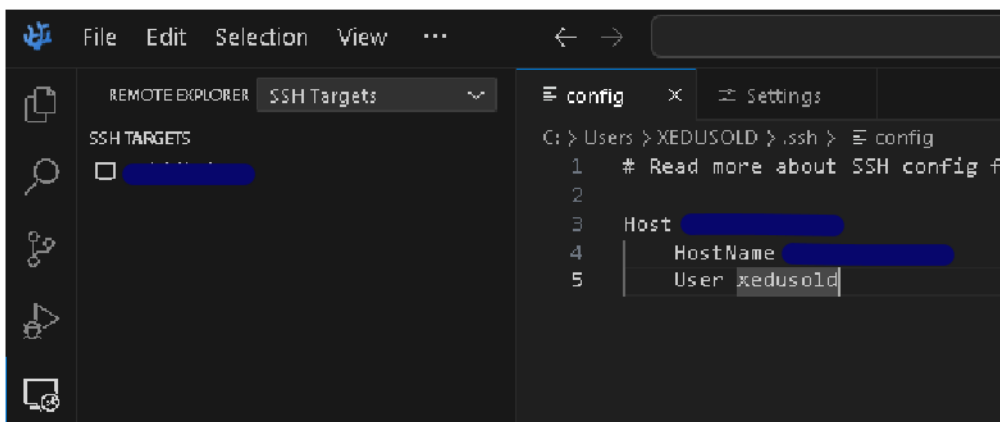


Abbildung 40: Anlegen des SSH-Zuganges zum Server

VSCodium: übertragenen Dateien

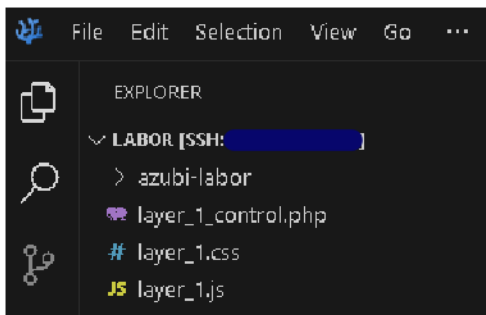


Abbildung 41: Über VSCodium angelegte und übertragene Quelldateien

VSCodium: PHP-Code Anpassung

```

<?php
...
echo "<script>window.location.replace('".$_SERVER['PHP_SELF']."?content=106');</script>";
...
<form action='".$_SERVER['PHP_SELF']."?content=106&change_port=true' method='POST'>
...
<form action='".$_SERVER['PHP_SELF']."?content=106&change_connection=true' method='POST'>
...
?>

```

Abbildung 42: PHP-Code Anpassung für Integration: Hinzufügen der Content-Nummer

```

<?php
...
echo "<link rel='stylesheet' href='layer_1.css' type='text/css'>";
...
echo "<script src='layer_1.js'></script>";
?>

```

Abbildung 43: PHP-Code Anpassung für Integration: Einbindung der verlinkten Dateien anpassen

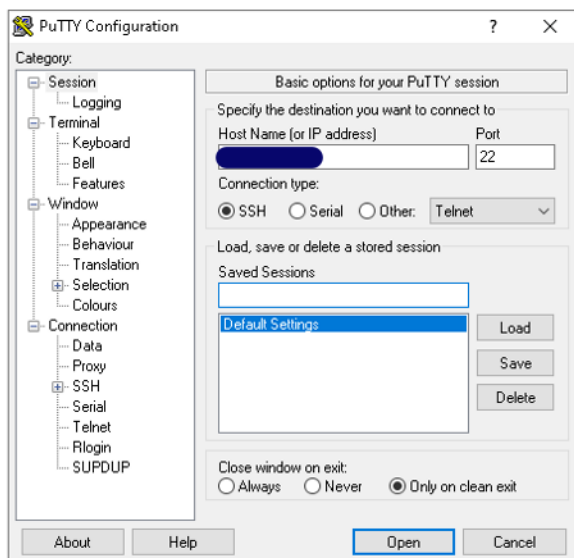
Dateirechte anpassen

Abbildung 44: Zugriff auf Server mit PuTTY

```

xedusold@:/labor$ ls -l
total 64
drwxrwxrwx 3 domänen-benutzer 4096 Mar  5 12:15 azubi-labor
-rw-r--r-- 1 xedusold domänen-benutzer 13867 Mar 12 12:10 layer_1_control.php
-rw-r--r-- 1 xedusold domänen-benutzer 10051 Mar 12 12:10 layer_1.css
-rw-r--r-- 1 xedusold domänen-benutzer 29736 Mar 12 12:10 layer_1.js

```

Abbildung 45: Dateirechte der über VSCodium angelegten Dateien

```

xedusold@: /labor$ chmod 004 layer*
xedusold@: /labor$ ls -l
total 64
drwxrwxrwx 3 domänen-benutzer 4096 Mar  5 12:15 aruba-labor
-----r-- 1 xedusold domänen-benutzer 13867 Mar 13 12:10 layer_1_control.php
-----r-- 1 xedusold domänen-benutzer 10051 Mar 12 12:10 layer_1.css
-----r-- 1 xedusold domänen-benutzer 29736 Mar 12 12:10 layer_1.js

```

Abbildung 46: Anpassung der Dateirechte der angelegten Dateien

A.7 Integrierte Verwaltungsoberfläche

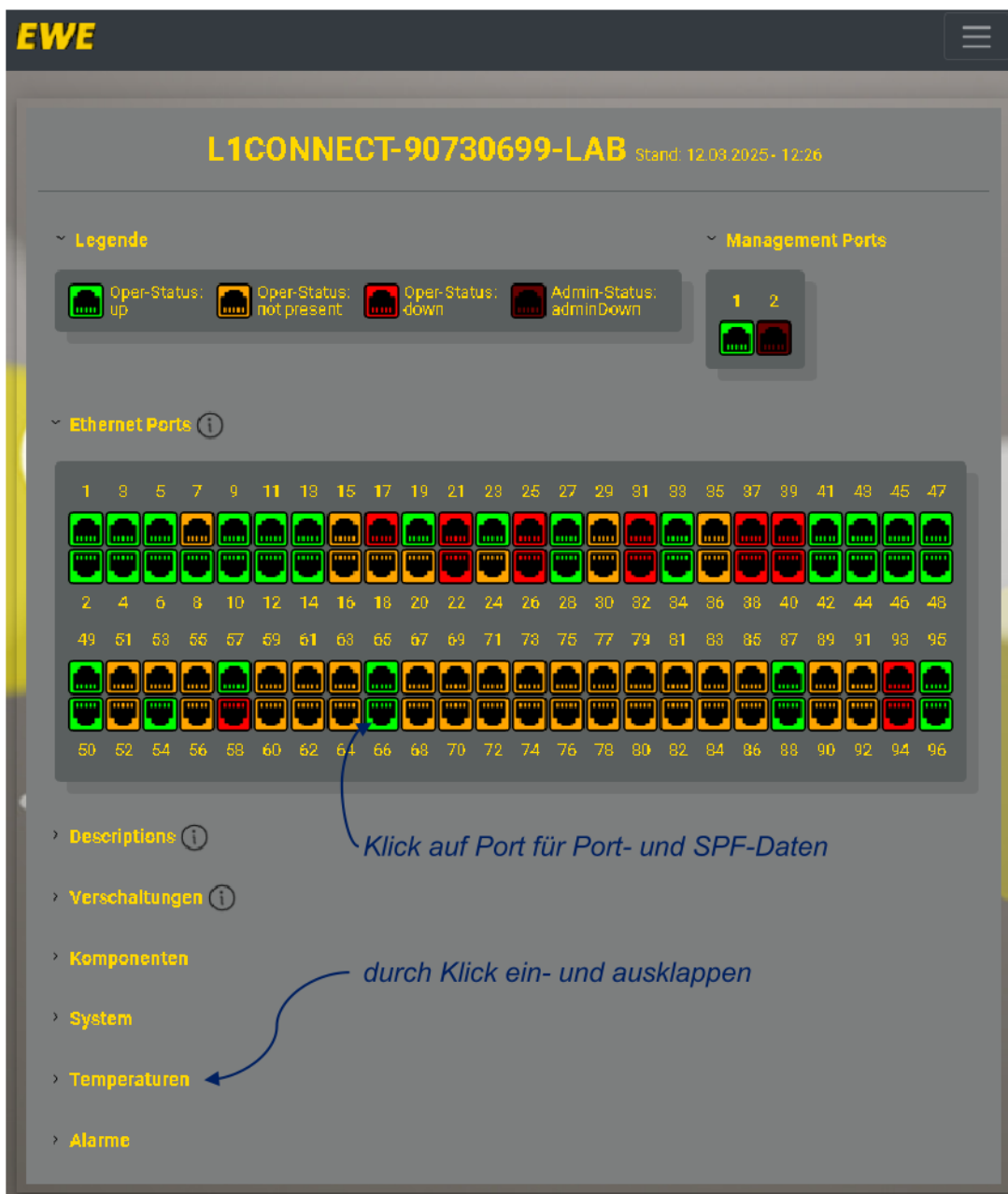


Abbildung 47: Integrierte Verwaltungsoberfläche für den L1S

A.8 Soll-Ist-Zeitvergleich

	SOLL	IST	DIFF
Vorbereitung und Planung	13	11,0	-2,0
Besprechung des Problems / Projektumfang	1	0,5	-0,5
Erfassung des Ist-Zustand	1	0,5	-0,5
Definition des Soll-Zustands	1	1	
Erstellung der Lösungsvorschläge	3	4	+1
Kostenvergleich mit Personalkostenberechnung	3	3	
Anwendung einer Präferenzmatrix	2	1,5	-0,5
Anwendung einer Nutzwertanalyse auf die Lösungen	2	0,5	-1,5
Verwirklichung der Lösung	16	19	+3
Konfiguration des L1-Verschalters	1	1	
Entwicklung der Lösung	14	17	+3
Integration der Lösung in die bestehende IT-Landschaft	1	1	
Testen und Fehlerprüfung	3	3	
Überprüfung der Funktionsfähigkeit der Lösung	3	3	
Abschluss	8	7	-1,0
Anfertigung der Betriebsdokumentation	3	3	
Anfertigung der Anwenderdokumentation	2	1	-1,0
Einweisen der Administratoren und Anwender	1	1	
Wirtschaftlichkeitsanalyse	2	2	
Gesamtzeit	40	40,0	

(in Excel erstellt)

Tabelle 7: Soll-Ist-Zeitvergleich

A.9 Ermittlung des Break-Even-Point

$$\frac{657,6 \text{ €}}{43,08 \frac{\text{€}}{\text{Stunden}}} \approx 15,26 \text{ einzusparende Stunden bis Break – Even – Punkt} \quad (12)$$

$$\frac{15,26 \text{ Stunden}}{12 \text{ Minuten}} \approx 77 \text{ Änderungen bis Break – Even – Punkt} \quad (13)$$

$$\frac{77 \text{ Änderungen}}{3 \frac{\text{Änderung}}{\text{Woche}}} \approx 26 \text{ Wochen} \approx 0,5 \text{ Jahre bis Break – Even – Punkt} \quad (14)$$