

CESC

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für Informatik

Eingereicht von:

Elias Mahr
Leopold Mistelberger
Timon Schmalzer

Betreuer:

Thomas Stütz

Projektpartner:

Herbsthofer

Leonding, 4. April 2025

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, 4. April 2025

Elias Mahr & Leopold Mistelberger

Abstract

Brief summary of our amazing work. In English. This is the only time we have to include a picture within the text. The picture should somehow represent your thesis. This is untypical for scientific work but required by the powers that are.



Zusammenfassung

Zusammenfassung unserer genialen Arbeit. Auf Deutsch. Das ist das einzige Mal, dass eine Grafik in den Textfluss eingebunden wird. Die gewählte Grafik soll irgendwie eure Arbeit repräsentieren. Das ist ungewöhnlich für eine wissenschaftliche Arbeit aber eine Anforderung der Obrigkeit. *Bitte auf keinen Fall mit der Zusammenfassung verwechseln, die den Abschluss der Arbeit bildet!*



Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 1 |
| 1.1 | Ausgangslage [Elias Mahr] | 1 |
| 1.2 | Ist-Zustand [Elias Mahr] | 1 |
| 1.3 | Problem [Leopold Mistelberger] | 1 |
| 1.4 | Aufgabenstellung [Leopold Mistelberger] | 2 |
| 2 | Umfeldanalyse | 3 |
| 3 | Technologien | 4 |
| 3.1 | Hardware und Betriebssystem [Leopold Mistelberger] | 4 |
| 3.2 | Kommunikation und Automatisierung [Leopold Mistelberger] | 5 |
| 3.3 | Container- und Laufzeitumgebung [Leopold Mistelberger] | 5 |
| 3.4 | Server- und Infrastrukturtechnologien [Elias Mahr] | 5 |
| 3.5 | Reverse Proxy [Elias Mahr] | 5 |
| 3.6 | Keycloak [Elias Mahr] | 6 |
| 3.7 | Reverse Proxy [Elias Mahr] | 7 |
| 3.8 | Cronjob | 9 |
| 3.9 | Angular | 9 |
| 3.10 | nager Api | 9 |
| 3.11 | JsPdf | 9 |
| 4 | Umsetzung | 10 |
| 4.1 | Keycloak [Elias Mahr] | 10 |
| 4.2 | Reverse Proxy [Elias Mahr] | 14 |
| 5 | Zusammenfassung | 15 |
| | Glossar | V |
| | Literaturverzeichnis | VI |

| | |
|------------------------------|-------------|
| Abbildungsverzeichnis | VII |
| Tabellenverzeichnis | VIII |
| Quellcodeverzeichnis | IX |
| Anhang | X |

1 Einleitung

1.1 Ausgangslage [Elias Mahr]

Das seit 1870 Familien geführte Unternehmen Herbsthofer plant, liefert, montiert und wartet HKLS-Anlagen für Industrie, Gesundheitswesen und Forschung. Beispielhafte Projekte wären die dm Zentrale, das MSD Krems und die Techbase in

Linz. Das Unternehmen nutzt 3D-Planung und viele moderne, digitale Tools zur Kosten- und Qualitätskontrolle. Nationale und internationale Projekte werden vom Hauptsitz Linz aus umgesetzt. Hier mehr dazu: <https://www.herbsthofer.at/leistungsspektrum>



Abbildung 1: Herbsthofer Logo

1.2 Ist-Zustand [Elias Mahr]

Das Unternehmen betreibt zahlreiche Baustellen im In- und Ausland. Für die Dauer der Projekte werden auf diesen Baustellen Containeranlagen genutzt, die als temporäres Büro, Aufenthalts- und Lagerräume dienen. Die Container sind je nach Nutzung mit W-Lan, Heizkörpern, Kühlsystem und Beleuchtung ausgestattet. Um einen durchgehenden Betrieb und geeignete Arbeitsbedingungen für das Personal zu gewährleisten wird durchgehend geheizt/gekühlt.

1.3 Problem [Leopold Mistelberger]

Baustellencontainer werden häufig ohne technische Überwachung betrieben, wodurch hohe Energiekosten entstehen, da hauptsächlich die Heizung unabhängig von der tatsächlichen Nutzung betrieben wird. Eine Kontrolle über den Zustand des Containers ist nicht gegeben, weshalb die Firma keinen Zugriff auf aktuelle Informationen über Heizung, Raumklima, geöffnete Türen, Nutzung, Anwesenheit, Temperatur, Luftfeuchtigkeit und Lichtstatus hat.

Dieser Mangel erschwert dem Betrieb eine effiziente Steuerung der Baustellencontainer und führt zu erhöhten Betriebskosten. Darüber hinaus stellen Container ein gewisses Sicherheitsrisiko dar. Einbrüche bleiben oftmals unbemerkt, da weder eine automatische Erkennung noch eine sofortige Benachrichtigung erfolgt. In solchen Fällen ist eine zeitnahe Reaktion nicht möglich, und Vorfälle können nur im Nachhinein nachvollzogen werden, was die Suche nach dem Täter erheblich erschwert.

1.4 Aufgabenstellung [Leopold Mistelberger]

Ziel dieser Diplomarbeit ist die Entwicklung des Systems CESC zur Überwachung und Steuerung eines Baustellencontainers. Das System soll Temperatur, Luftfeuchtigkeit, Türstatus und Anwesenheit erfassen und visualisieren sowie bei kritischen Ereignissen, wie etwa einem Einbruch, Benachrichtigungen senden.

Zusätzlich soll CESC die Steuerung der Heizung, der Klimatisierung und der Beleuchtung ermöglichen. Bei kritischen oder sicherheitsrelevanten Ereignissen sollen automatisch Benachrichtigungen versendet werden. Durch den Einsatz von CESC sollen Energiekosten gesenkt und die Sicherheit auf Baustellen erhöht werden.

2 Umfeldanalyse

Citing [1] properly.

Was ist eine GUID? Eine GUID kollidiert nicht gerne.

Kabellose Technologien sind in abgelegenen Gebieten wichtig [2].

3 Technologien

3.1 Hardware und Betriebssystem [Leopold Mistelberger]

3.1.1 Raspberry Pi mit Ubuntu [Leopold Mistelberger]

3.1.2 Beschreibung und Rolle

Der Raspberry Pi 5 ist ein kleiner, kompakter Computer, der als zentrales Gehirn eines oder mehrerer umliegende Baustellencontainer dient. Er übernimmt die Erfassung sämtlicher Sensordaten, die Steuerung der Aktoren und die Kommunikation mit der Datenbank sowie dem Proxmox-Server im Unternehmen.

3.1.3 Technische Eigenschaften

- Leistungsfähiger Prozessor
- 16 Gigabyte Arbeitsspeicher
- Netzwerkanschluss
- 4x USB-Anschlüsse
- GPIO-Pins
- Kompakt
- Langlebig
- Energieeffizient
- Kompatibel mit Ubuntu

3.1.4 Begründung der Wahl

Der Raspberry Pi war für CESC die perfekte Wahl, da er alle unsere technischen Voraussetzungen erfüllt. Dazu zählen ein USB-Anschluss für den Zigbee-Dongle, ein

Netzwerkanschluss für die Kommunikation mit Servern und der Datenbank sowie ein leistungsstarker Prozessor für die Ausführung unserer Programme. Ubuntu wurde als Betriebssystem gewählt, um eine bessere Wartbarkeit und Zuverlässigkeit aller eingesetzten Softwaredienste zu gewährleisten. Der Raspberry Pi ermöglicht somit eine zuverlässige Steuerung der Baustellencontainer.

3.1.5 Proxmox [Leopold Mistelberger]

3.2 Kommunikation und Automatisierung [Leopold Mistelberger]

3.2.1 Zigbee [Leopold Mistelberger]

3.2.2 MQTT [Leopold Mistelberger]

3.2.3 Home Assistant [Leopold Mistelberger]

3.3 Container- und Laufzeitumgebung [Leopold Mistelberger]

3.3.1 Docker [Leopold Mistelberger]

3.3.2 GitHub Container Registry (ghcr.io) [Leopold Mistelberger]

3.4 Server- und Infrastrukturtechnologien [Elias Mahr]

3.5 Reverse Proxy [Elias Mahr]

Ein Revers Proxy steht zwischen dem Webserver und dem Internet. Im Grunde fungiert es wie ein Türsteher für den Server. Anfragen werden von außen entgegen genommen und zum richtigen Dienst weitergeleitet(Frontend, Backend, Keycloak,...). Durch dies werden die echten Adressen verschleiert.

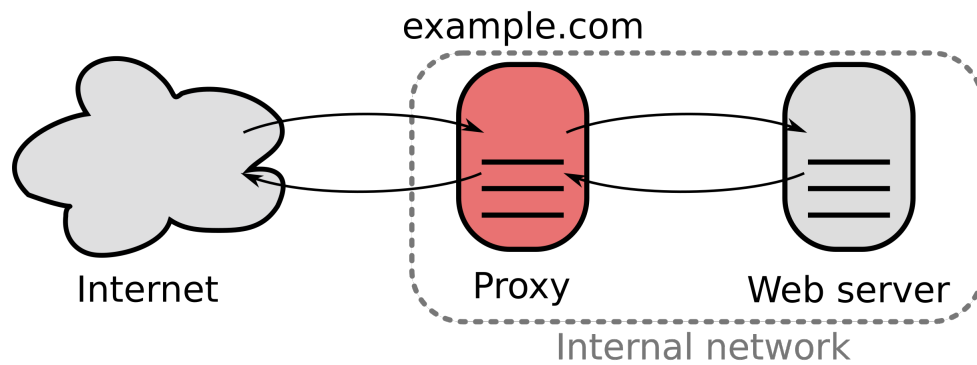


Abbildung 2: Reverse Proxy (Quelle: https://commons.wikimedia.org/wiki/File:Reverse_proxy_h2g2bob.svg, Lizenz: CC0)

3.5.1 Nutzen in der Diplomarbeit

Hauptsächlich gibt es drei Hauptgründe weshalb wir ein Reversproxy einbauen wollten/mussten.

- Keycloaks https Redirects: Keycloak verwendet moderne Standards wie OpenID Connect. Diese Standards setzen auch eine HTTPs Redirect URL voraus. Https setzt im Gegensatz zu http die nötigen Sicherheitsstandarts um zum Beispiel Man in the Middle Angriffe zu verhindern.
- Port Chaos: Mit vielen verschiedenen Ports wie 80 für das Frontend, 8080 für Keycloak und 5000 für das Backend wäre alles sehr unübersichtlich. Der User müsste sich 3 URLs merken und im Docker Netzwerk würde es ständig Konflikte geben.
- Sicherheit: Umso mehr Ports öffentlich zugänglich sind umso mehr Angriffsfläche gibt es für eine Potenzielle Bedrohung. Ports wie 5000 für das Backend sind von außen gar nicht erreichbar so bleiben die internen Dienste unsichtbar. Auch wenn die Webapplikation nur durch das Intranet erreicht werden kann ist eine zusätzliche Sicherheitsstufe nie ein Fehler.

3.6 Keycloak [Elias Mahr]



Abbildung 3: Keycloak Logo

Keycloak ist ein Open Source System, das die sichere Anmeldung und die Rechteverwaltung für unsere Anwendungen übernimmt. Die Anwendungen schicken die Anmeldung an Keycloak und müssen das Speichern und das Prüfen von Benutzerkonten nicht mehr selbst erledigen. So kann die Anwendung sich auf das Wesentliche konzentrieren, während Keycloak die Anmeldung und die Rechte im Hintergrund mit Sicherheitsstandards für Unternehmen regelt.

3.6.1 Grundidee und Zweck

Die Firma Herbsthofer legt sehr großen Wert auf Sicherheit. Die externe Bedienung von Heizung, Kühlung und Beleuchtung kann, wenn es ausgenutzt wird zu erheblichen Schäden führen.

- **Mitarbeiterschutz:** Bei voll eingeschalteter Kühlung an kalten Wintertagen kann es zu Gesundheitlichen Schäden in Form von verkühlungen bei Arbeitern/Arbeiterinnen führen, die auch mit einem Ausfall für ein paar Tage enden können und so der Firma Geld kosten.
- **Brandgefahr:** Nicht zu vergessen ist auch die erhöhte Brandgefahr. Unkontrollierte Heizbetriebe auf einer zu hohen Temperaturstufe kann im Schlimmsten Fall zu Bränden führen und so erheblichen Schaden anrichten und Personenschaden mit sich führen.
- **Kosten:** Strom wird immer Teurer. Bei durchgehender Nutzung auch an Tagen an denen niemand im Container ist, ist erstens der Schaden an der Umwelt durch extreme Ressourcen Verschwendung zu Beachten. Jedoch auch die immensen, unnötigen Kosten die, die Firma bei Ihrer Stromrechnung erwarten.

«««< HEAD =====

3.7 Reverse Proxy [Elias Mahr]

Ein Revers Proxy steht zwischen dem Webserver und dem Internet. Im Grunde fungiert es wie ein Türsteher für den Server. Anfragen werden von außen entgegen genommen und zum richtigen Dienst weitergeleitet(Frontend, Backend, Keycloak,...). Durch dies werden die echten Adressen verschleiert.

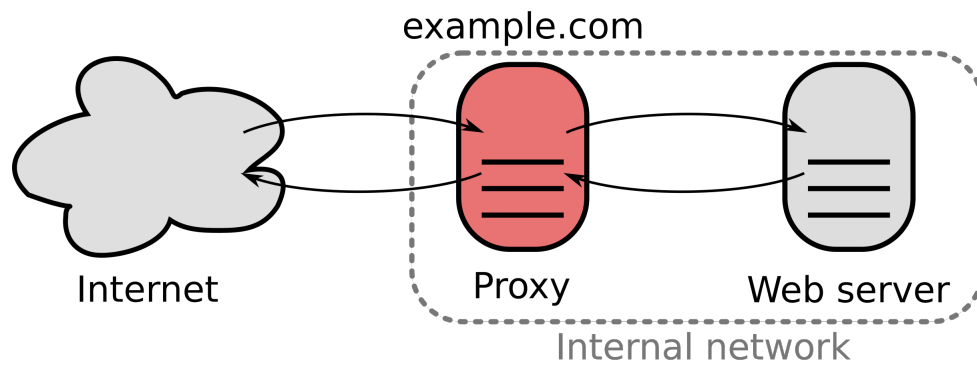


Abbildung 4: Reverse Proxy (Quelle: https://commons.wikimedia.org/wiki/File:Reverse_proxy_h2g2bob.svg, Lizenz: CC0)

3.7.1 Nutzen in der Diplomarbeit

Hauptsächlich gibt es drei Hauptgründe weshalb wir ein Reversproxy einbauen wollten/mussten.

- Keycloaks https Redirects: Keycloak verwendet moderne Standards wie OpenID Connect. Diese Standards setzen auch eine HTTPs Redirect URL voraus. Https setzt im Gegensatz zu http die nötigen Sicherheitsstandarts um zum Beispiel Man in the Middle Angriffe zu verhindern.
- Port Chaos: Mit vielen verschiedenen Ports wie 80 für das Frontend, 8080 für Keycloak und 5000 für das Backend wäre alles sehr unübersichtlich. Der User müsste sich 3 URLs merken und im Docker Netzwerk würde es ständig Konflikte geben.
- Sicherheit: Umso mehr Ports öffentlich zugänglich sind umso mehr Angriffsfläche gibt es für eine Potenzielle Bedrohung. Ports wie 5000 für das Backend sind von außen gar nicht erreichbar so bleiben die internen Dienste unsichtbar. Auch wenn die Webapplikation nur durch das Intranet erreicht werden kann ist eine zusätzliche Sicherheitsstufe nie ein Fehler.

3.8 Cronjob

3.9 Angular

3.10 nager Api

3.11 JsPdf

»»»> 00bd5f3ab94f5427a714e6891446744df4fd51c5

4 Umsetzung

4.1 Keycloak [Elias Mahr]

4.1.1 Einbindung ins Frontend

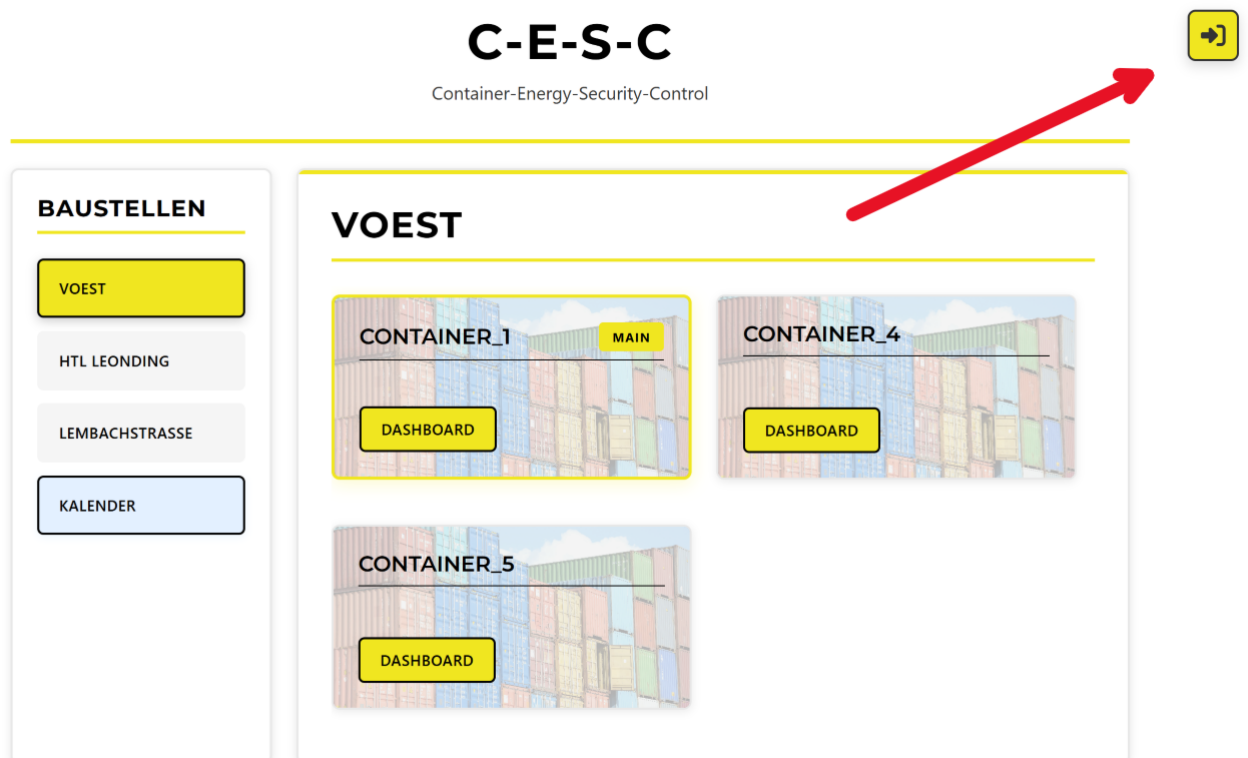


Abbildung 5: Button im Frontend

Um die Webapp möglichst unkompliziert und Benutzerfreundlich zu halten ist der Button wie im Screenshot rechts oben in der Ecke platziert. Unabhängig vom Scroll-Zustand oder der aktuell ausgewählten Page bleibt er auch immer dort, um sich sofort und schnell anmelden zu können.



Abbildung 6: Login- und Logout-Buttons

Das Icon stammt aus der Font Awesome Bibliothek:

```
<i class="fas fa-sign-in-alt"></i>
```

Bei Klick darauf wird man weitergeleitet zum Pfad:

<https://192.168.20.202/auth/realms/cesc/protocol/openid-connect>

/auth?client_id=cesc-frontend&redirect_uri=https... Hier kommt dann die Standard Keycloak Login Page. Nach erfolgreicher Anmeldung bekommt der User seinen Token, wird automatisch wieder umgeleitet zur vorher besuchten Seite und es werden alle Funktionen der Webseite freigeschaltet.

4.1.2 das Realm

Ein Realm in Keycloak ist ein isolierter Sicherheitsbereich. Er umfasst Benutzer, Rollen, Clients und Policies für eine bestimmte Anwendung. Wir haben ein Realm namens “cesc” erstellt. Es definiert die gesamte Identifikation und Authorisierung für die Weboberfläche. Zurzeit ist es möglichst klein gehalten, jedoch vollkommen ausreichend. Ein Admin User(herbsthofe) steuert alles, in Zukunft kann man jedoch leicht noch andere Benutzer/Rollen hinzu fügen.

Realm Einstellungen

Listing 1: Keycloak Realm Konfiguration

```

1  {
2    "realm": "cesc",
3    "enabled": true,
4    "sslRequired": "external",
5    "registrationAllowed": false,
6    "loginWithEmailAllowed": true,
7    "duplicateEmailsAllowed": false,
8    "resetPasswordAllowed": true,
9    "editUsernameAllowed": false,
10   "bruteForceProtected": true

```

```
11 }
```

In den wichtigsten Einstellungen wird unter anderem definiert das für alle externen Verbindungen SSL vorgeschrieben ist. Die Zeile `bruteForceProtected: true` ist eine einfache Absicherung gegen Brute Force Angriffe von Keycloak. Durch sie wird ein Account nach mehreren fehlgeschlagenen Anmeldungen Temporär gesperrt. Um zukünftige Admins der Webseite eine leichte und unkomplizierte Anmeldung zu ermöglichen ist das einloggen mit der email Adresse auch erlaubt.

Rollen

Listing 2: Keycloak Realm Konfiguration

```
1 {
2   "roles": {
3     "realm": [
4       {
5         "name": "admin",
6         "description": "Administrator role for CESC
          application",
7         "composite": false,
8         "clientRole": false
9       }
10    ]
11  }
12 }
```

Die wie oben erwähnt zurzeit einzige Rolle `admin` hat kurtzgesatz Rechte auf alles. Durch die Zeile `"clientRole : false` is sie zum Beispiel auch nicht auf einen Gewissen Client beschränkt sondern hat im ganzen Realm auf alles Zugriff.

Benutzer

Listing 3: Keycloak Realm Konfiguration

```
1 {
2   "users": [
3     {
4       "username": "herbsthofer",
5       "enabled": true,
6       "emailVerified": true,
7       "firstName": "Admin",
8       "lastName": "Herbsthofer",
9       "email": "herbsthofer@example.com",
10
11       "realmRoles": [
12         "admin"
13     ]
14   }
15 ]
```

```

14     }
15   ]
16 }

```

Der Zurzeit einzige User `herbsthofer` erhält durch die Zuweisung der `admin` Rolle, "`realmRoles` " : [`admin` "] alle Administrator rechte.

Clients

Listing 4: Keycloak Realm Client Konfiguration

```

1  {
2    "clients": [
3      {
4        "clientId": "cesc-frontend",
5        "name": "CESC Frontend Application",
6        "enabled": true,
7        "publicClient": true,
8        "protocol": "openid-connect",
9        "standardFlowEnabled": true,
10       "redirectUris": [
11         "https://192.168.20.202/*",
12         "http://localhost:4200/*"
13       ],
14       "webOrigins": [
15         "https://192.168.20.202",
16         "http://localhost:4200"
17       ]
18     },
19     {
20       "clientId": "cesc-backend",
21       "name": "CESC Backend API",
22       "enabled": true,
23       "publicClient": false,
24       "bearerOnly": true,
25       "protocol": "openid-connect"
26     }
27   ]
28 }

```

Für unsere Anwendung werden nur zwei Clients benötigt. Der `cesc-frontend` Client ist der öffentliche Client (`publicClient: true`) für die Angular Anwendung. Die `redirectUris` und `webOrigins` legen die erlaubten URLs für die Produktion (192.168.20.202) und die lokale Entwicklung (localhost:4200) fest. Der `cesc-backend` Client ist als vertraulicher Client gemacht, da er keine Login Page bereitstellt. (`publicClient: false`).

4.2 Reverse Proxy [Elias Mahr]

Nginx ist als eigener Docker Container mit nginx.conf implimentiert. Nur der Port 443 ist öffentlich mit Let's Encrypt-Zertifikaten. Die Ports werden Pfadbasiert weitergeleitet:

- http://Keycloak:8080 → location /auth

Listing 5: Nginx Konfiguration /auth

```

1      location /auth {
2          proxy_pass http://keycloak:8080;
3          proxy_set_header Host $host;
4          proxy_set_header X-Real-IP $remote_addr;
5          proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
6          proxy_set_header X-Forwarded-Proto https;
7          proxy_set_header X-Forwarded-Host $host;
8          proxy_set_header X-Forwarded-Port 443;
9          proxy_buffer_size 128k;
10         proxy_buffers 4 256k;
11         proxy_busy_buffers_size 256k;
12     }
```

- http://frontend:80 → location /

Listing 6: Nginx Konfiguration /

```

1      location / {
2          proxy_pass http://frontend:80;
3      }
```

- http://backend:5000 → location /api

Listing 7: Nginx Konfiguration /api

```

1      location /api {
2          proxy_pass http://backend:5000;
3      }
```

Siehe tolle Daten in Tab. 1.

Siehe und staune in Abb. ??.

Dann betrachte den Code in Listing ??.

| | Regular Customers | Random Customers |
|-----------|-------------------|------------------|
| Age | 20-40 | >60 |
| Education | university | high school |

Tabelle 1: Ein paar tabellarische Daten

5 Zusammenfassung

Aufzählungen:

- Itemize Level 1
 - Itemize Level 2
 - Itemize Level 3 (vermeiden)
- 1. Enumerate Level 1
 - a. Enumerate Level 2
 - i. Enumerate Level 3 (vermeiden)

Desc Level 1

Desc Level 2 (vermeiden)

Desc Level 3 (vermeiden)

Glossar

GUID Globally Unique Identifier

Literaturverzeichnis

- [1] P. Rechenberg, G. Pomberger *et al.*, *Informatik Handbuch*, 4. Aufl. München – Wien: Hanser Verlag, 2006.
- [2] Association for Progressive Communications, „Wireless technology is irreplaceable for providing access in remote and scarcely populated regions,” 2006, letzter Zugriff am 23.05.2021. Online verfügbar: <http://www.apc.org/en/news/strategic/world/wireless-technology-irreplaceable-providing-access>

Abbildungsverzeichnis

| | | |
|---|---|----|
| 1 | Herbsthofer Logo | 1 |
| 2 | Reverse Proxy (Quelle: https://commons.wikimedia.org/wiki/File:Reverse_proxy_h2g2bob.svg , Lizenz: CC0) | 6 |
| 3 | Keycloak Logo | 6 |
| 4 | Reverse Proxy (Quelle: https://commons.wikimedia.org/wiki/File:Reverse_proxy_h2g2bob.svg , Lizenz: CC0) | 8 |
| 5 | Button im Frontend | 10 |
| 6 | Login- und Logout-Buttons | 11 |

Tabellenverzeichnis

| | | |
|---|--|----|
| 1 | Ein paar tabellarische Daten | 14 |
|---|--|----|

Quellcodeverzeichnis

| | | |
|---|---|----|
| 1 | Keycloak Realm Konfiguration | 11 |
| 2 | Keycloak Realm Konfiguration | 12 |
| 3 | Keycloak Realm Konfiguration | 12 |
| 4 | Keycloak Realm Client Konfiguration | 13 |
| 5 | Nginx Konfiguration /auth | 14 |
| 6 | Nginx Konfiguration / | 14 |
| 7 | Nginx Konfiguration /api | 14 |

Anhang