HTL Leonding
next level

HÖHERE TECHNISCHE
BUNDESLEHRANSTALT
LEONDING

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 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]

Abbildung 2: Proxmox Virtual Environment Logo

Proxmox Virtual Environment (Proxmox VE) ist eine Plattform zur Servervirtualisierung, die Open Source ist. Sie erlaubt den Betrieb mehrerer virtueller Systeme auf einer einzigen physischen Hardware und findet häufig Anwendung in Unternehmensumgebungen.

Die Verwaltung wird zentral über eine webbasierte Benutzeroberfläche durchgeführt. Weitere Details finden Sie auf Proxmox offizieller Website (https://www.proxmox.com).

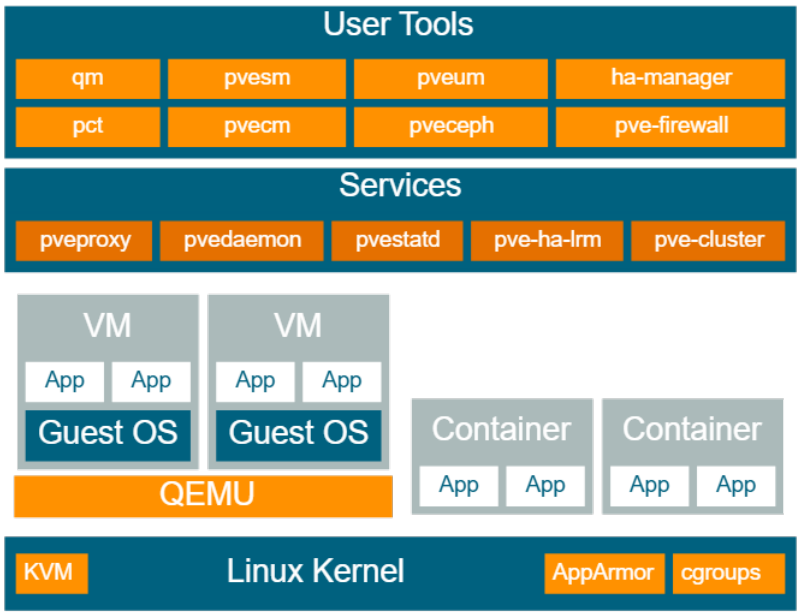

Abbildung 3: Architektur von Proxmox Virtual Environment

Proxmox läuft auf einem Linux-Hostsystem und verwendet verschiedene Virtualisierungstechnologien. Proxmox fügt zwischen der Hardware und den Anwendungen eine

Virtualisierungsschicht ein. Diese dient dazu, Systeme, die auf derselben Hardware laufen, voneinander zu trennen.

Virtuelle Maschinen (VMs) verhalten sich dabei jeweils wie ein eigener Computer mit einem eigenen Betriebssystem. Dadurch sind sie vollständig voneinander getrennt. CPU, Arbeitsspeicher und Speicherplatz werden im Vorhinein fest zugewiesen. Eine VM ist somit vollständig von anderen Systemen isoliert.

Zusätzlich unterstützt Proxmox Container auf Basis von Linux Containers (LXC). Container sind im Vergleich zu VMs leichter und schneller, da sie sich den Linux-Kernel des Servers teilen und arbeiten dadurch deutlich effizienter. Dadurch benötigen sie wenige Ressourcen und eignen sich sehr für den Betrieb Dienste oder Anwedungen.

Die Vorteile von Proxmox liegen unter anderem darin, dass keine Lizenzkosten anfallen, da Proxmox kostenlos verfügbar ist. Dadurch können die Betriebskosten eines Unternehmens gesenkt werden, insbesondere da Virtualisierungslösungen für virtuelle Maschinen zunehmend kostenintensiver werden. Zudem ermöglicht Proxmox eine effizientere Nutzung der vorhandenen Hardware, während die Systeme sauber voneinander getrennt bleiben. Durch das webbasierte Interface erhält der Benutzer eine gute Übersicht, da alle Systeme zentral verwaltet werden können.

3 dargestellt.

3.2 Kommunikation und Automatisierung [Leopold Mistelberger]

3.2.1 Zigbee [Leopold Mistelberger]

3.2.2 MQTT [Leopold Mistelberger]

Beschreibung

MQTT (Message Queuing Telemetry Transport) ist ein leichtgewichtiges Nachrichtenprotokoll, das nach dem sogenannten *Publish-Subscribe-Prinzip* arbeitet [**?**]. Dieses Protokoll wurde speziell für die Übertragung von Daten mit geringer Bandbreite oder instabiler Verbindung entwickelt. Es arbeitet ereignisbasiert und eignet sich besonders für Projekte, bei denen viele Geräte regelmäßig kleine Datenmengen wie Temperatur oder Luftfeuchtigkeit senden, wie beispielsweise in CESC.

Komponenten und Funktionsweise

Bei MQTT gibt es drei zentrale Komponenten, die nach dem Pub-Sub-Prinzip arbeiten:

- **Publisher:** Ein Gerät oder Programm, das bestimmte Daten an den Broker sendet. In den meisten Fällen handelt es sich hierbei um einen Sensor.
- **Broker:** Meist ein zentraler Server, der alle Nachrichten empfängt und an die richtigen Empfänger weiterleitet.
- **Subscriber:** Ein Gerät oder Programm, das bestimmte Daten abonniert und diese empfängt.

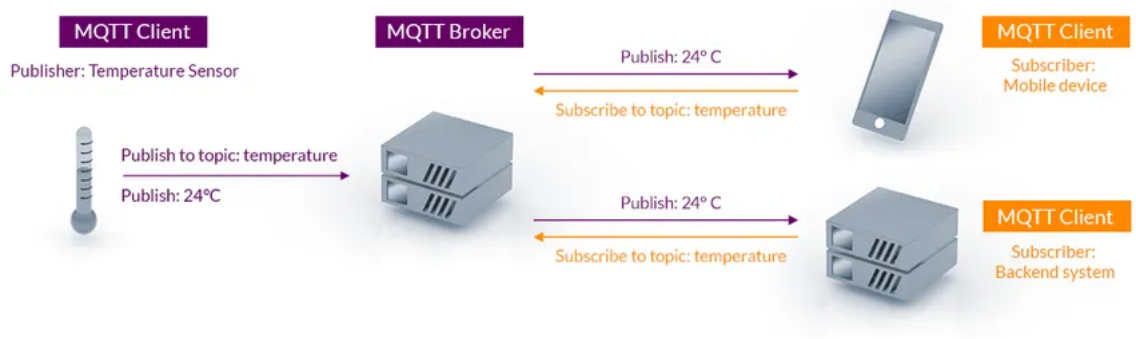

Abbildung 4: MQTT-Architektur: Publisher, Broker, Subscriber (Quelle: https://mqtt.org/assets/img/mqtt-publish-subscribe.png, Lizenz: CC0)

Die Daten werden in sogenannten *Topics* organisiert. Ein Publisher sendet Nachrichten an ein bestimmtes Topic, während Subscriber dieses Topic abonnieren. Der Broker übernimmt die Verteilung der Nachrichten, ohne dass sich Sender und Empfänger direkt kennen müssen.

Beispielhafte Nutzung

Um Daten zu empfangen, muss ein Subscriber ein bestimmtes Topic abonnieren. Beispielsweise möchte man die Raumtemperatur eines Containers auslesen. Dazu wird ein Subscriber auf das entsprechende Topic gesetzt:

Listing 1: Subscriber für Raumtemperatur des Containers 147

```
1  mosquitto_sub -h <Broker-IP> -t "container/147/raum1/temperatur"
```

Neben dem Empfangen von Daten kann man auch Aktoren steuern, wie zum Beispiel die Heizung des Containers. Um diese einzuschalten, verwendet man den folgenden Publisher-Befehl:

Listing 2: Heizung des Containers 147 einschalten

```
mosquitto_pub -h <Broker-IP> -t "container/147/raum1/heizung" -m "ON"
```

Vorteile von MQTT

MQTT bietet mehrere Vorteile gegenüber klassischen Kommunikationsprotokollen:

- Geringer Netzwerk- und Ressourcenverbrauch
- Zuverlässige Datenübertragung auch bei instabilen Verbindungen
- Klare Trennung von Sendern und Empfängern
- Hohe Skalierbarkeit bei vielen beteiligten Geräten
- Besonders geeignet für IoT- und Sensornetzwerke

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.

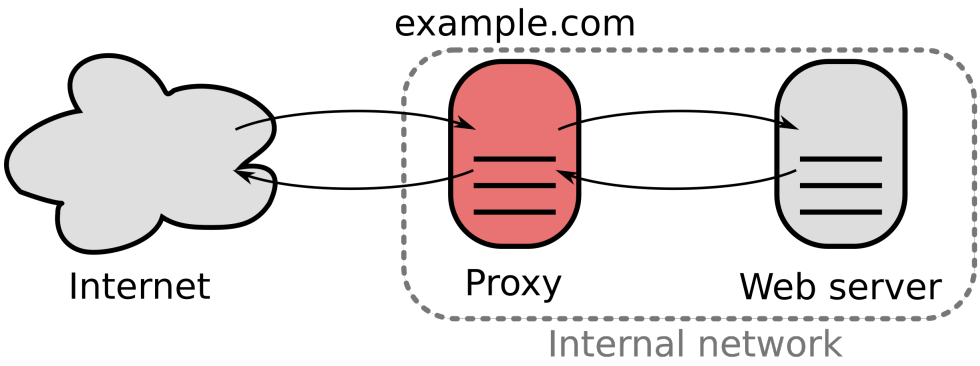

Abbildung 5: Reverse Proxy (Quelle: https://commons.wikimedia.org/wiki/File:Reverse_proxy_h2g2bob.svg, Lizenz: CC0)

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 Webaplikation nur durch das Intranet erreicht werden kann ist eine zusätzliche Sicherheitsstufe nie ein Fehler.

3.6 Keycloak [Elias Mahr]

Abbildung 6: 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 Verschwändung zu Beachten. Jedoch auch die immensen, unnötigen Kosten die, die Firma bei Ihrer Stromrechnung erwarten.

3.7 Cronjob

3.8 Angular

3.9 Api

4 Umsetzung

4.1 Keycloak [Elias Mahr]

4.1.1 Einbindung ins Frontend

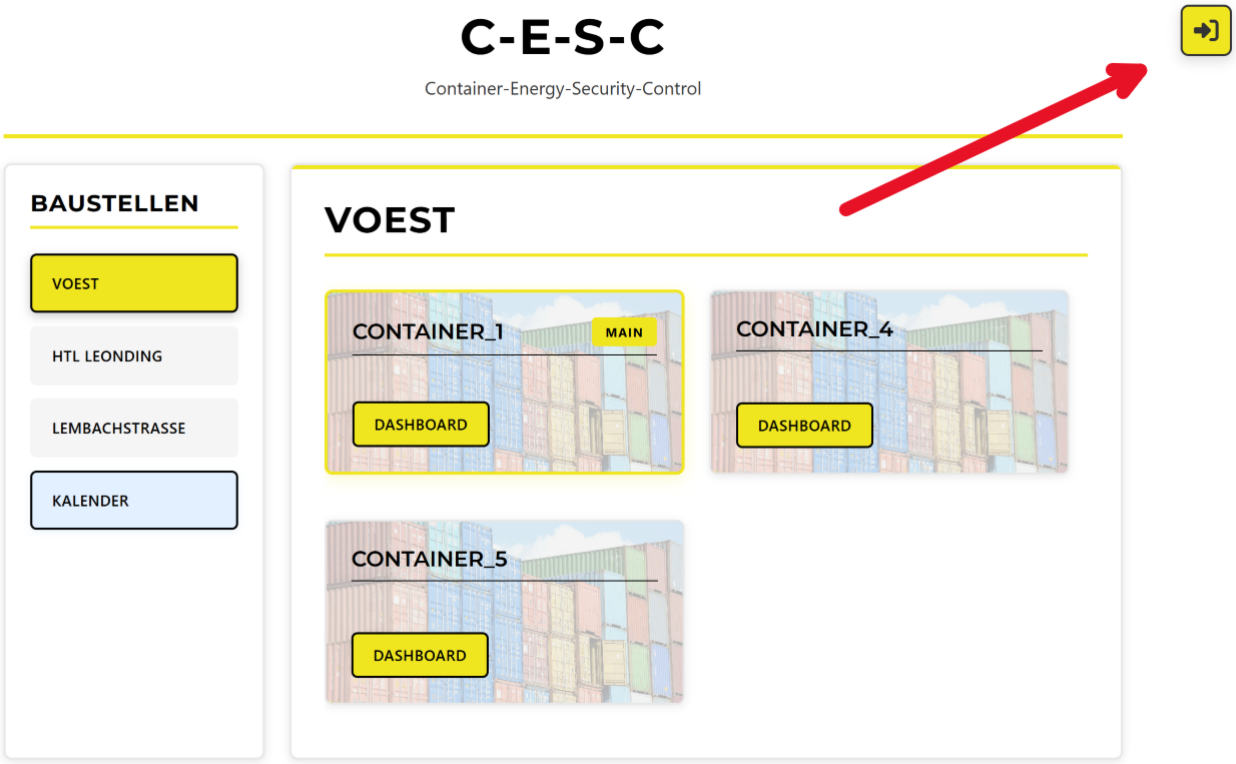

Abbildung 7: Button im Frontend

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.

(a) Anmelde-Button (b) Abmelde-Button

Abbildung 8: 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 Standart 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 freigeschalten.

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(herbsthofer) steuert alles, in Zukunft kann man jedoch leicht noch andere Benutzer/Rollen hinzu fügen.

Realm Einstellungen

Listing 3: Keycloak Realm Konfiguration

```
{
  "realm": "cesc",
  "enabled": true,
  "sslRequired": "external",
  "registrationAllowed": false,
  "loginWithEmailAllowed": true,
  "duplicateEmailsAllowed": false,
  "resetPasswordAllowed": true,
  "editUsernameAllowed": false,
  "bruteForceProtected": true
```

```
}
```

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 4: Keycloak Realm Konfiguration

```
{
  "roles": {
    "realm": [
      {
        "name": "admin",
        "description": "Administrator role for CESC
           application",
        "composite": false,
        "clientRole": false
      }
    ]
  }
}
```

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 beschränk sondern hat im ganzen Realm auf alles Zugriff.

Benutzer

Listing 5: Keycloak Realm Konfiguration

```
{
"users": [
    {
      "username": "herbsthofer",
      "enabled": true,
      "emailVerified": true,
      "firstName": "Admin",
      "lastName": "Herbsthofer",
      "email": "herbsthofer@example.com",

      "realmRoles": [
        "admin"
      ]
```

```
14       }
15     ]
16   }
```

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

Clients

Listing 6: 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 7: Nginx Konfiguration /auth

```
location /auth {
    proxy_pass http://keycloak:8080;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto https;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Port 443;
    proxy_buffer_size 128k;
    proxy_buffers 4 256k;
    proxy_busy_buffers_size 256k;
}
```

- http://frontend:80 → location/

Listing 8: Nginx Konfiguration /

```
location / {
    proxy_pass http://frontend:80;
}
```

- http://backend:5000 → location /api

Listing 9: Nginx Konfiguration /api

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

4.3 Angular

4.3.1 Components

start-list-container

Abbildung 9: Startlist Component

Die “StartListContainerComponent” dient als Startseite der Anwendung. Dort wird bei öffnen schon eine Komplette Übersicht über alles Verfügbaren Baustellen in der Datenbank gegeben. Die Seite ist in 3 Wesentliche Abschnitte unterteilt:

- **Titel**

  Ganz ober sieht man sofort den Titel unserer Arbeit(C-E-S-C) mit dem Vollausgeschriebenen Titel darunter(Container-Energy-Security-Controll).

- **Baustellen Übersicht**

  Links auf 1/4 der breite sind alle Baustellen aufgelistet. Bei der Auflistung werden die Namen der Baustellen aus der Datenbank genommen. Die ausgewählte Baustelle wird immer Gelb mit Ramen markiert, so auch wenn die Maus darüber Hovert. In diesem Fall wird das Element auch ein wenig eingerückt. Immer am Ende des Abteils ist das Blau eingefärbte Kalender Element. Dieses ist Blau damit man es immer gut von den Baustellen unterscheiden kann.

- **Container Übersicht**

  Rechts bis Mittig zu 3/4 der Breite ist die Übersicht der Jeweiligen Container zu einer Baustelle. Die Angezeigten Container werden bei Klicken auf eine anderen Baustelle gewechselt. Optisch wird unterschieden zwischen Main-Container und normalen Container.

(a) Main Container (b) normaler Container

Abbildung 10: unterschied der Container

  Durch Gelben Rand und der Kennzeichnung “MAIN” neben dem Container Namen. Bei Klick auf eine Componente wird man zum jeweiligen Dashboard weiter geleitet. Der Name(im Screenshot z.B. `Container_1`), wird aus der Datenbank genommen und kann natürlich auf Wunsch geändert werden. Deffaultmäßig ist es zurzeit “`Container_n`”.

calender-editor

dashboard

door-logs

kalender

login-modal

navbar

plotly-chart

- can- und Logout-Buttonslender-editor
- dashboard
- door-logs
- kalender
- login-modal

- navbar
- plotly-chart
- start-list-container

Services

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

Tabellenverzeichnis

Quellcodeverzeichnis

Anhang