Fachhochschule Nordwestschweiz
Hochschule für Technik

Studiengang Systemtechnik

Remote-Screening

Erhöhung des Sicherheitsstandards von über Paket- und Briefpost eingeführten gefährlichen Gegenständen

Ausgangslage

Viele Schweizer Kunden der Firma Max C. Meister AG betreiben ihre Röntgenanlagen durch einen eigenen Angestellten, welcher sich nicht im Detail mit dem Auswerten von Röngtenbildern auskennt. Die Firma Custodio AG kann nun mit Ihren gut ausgebildeten Screenern, diese Verantwortung übernehmen.

Zielsetzung

Es soll eine Remote-Screening-Lösung geschaffen werden, die es ermöglicht, die beim Kunden anfallenden Röntgenbilder Remote aus dem Büro der Firma Custodio AG auszuwerten und die Bilder zusammen mit dem Resultat zurück an den Kunden zu senden.

Konzept

Das Konzept sieht vor, einen Server des Schweizer Hosters Exoscale, in Betrieb zu nehmen, welcher zentral die Bilder der in der Schweiz verteilten Rapiscan-Röntgenanlagen aufnimmt und weiter verarbeitet.

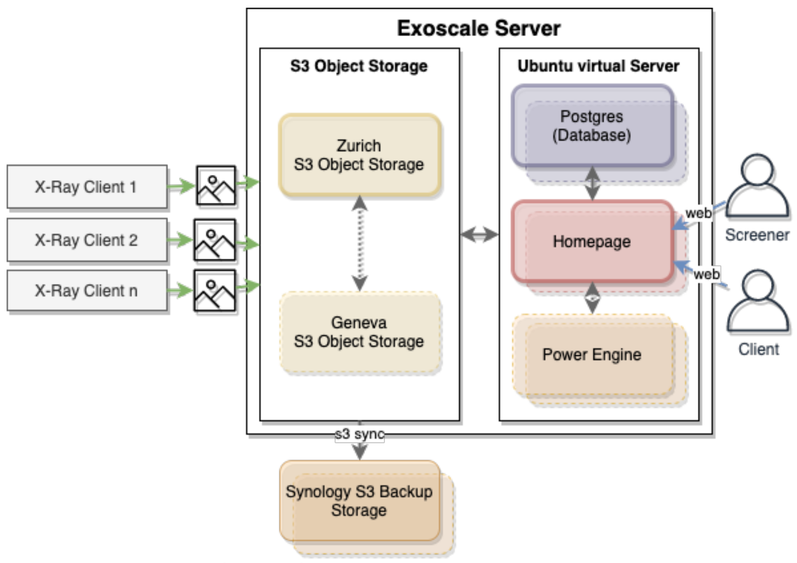

Fig.1: Konzeptübersicht

Mit einem auf Python basierenden Uploadskript werden die Bildarchive der Röntgenanlagen auf neue Bilder geprüft und, falls Neue vorhanden sind, hochgeladen. Die Bilder sind aber in einem herstellereigenen Format geschrieben und müssen auf dem Server, zuerst in ein standardisiertes Bildformat umgewandelt werden. Zwei weitere Skripts erzeugen in der verwendeten PostgreSQL-Datenbank die Einträge der anfallenden Dateien.

```
...
def upload_folder(folder):
    #List all O  [Linter] Error running Pylint
    for obj in   See Console for more info.
        #If Obj
        argumen
        if (os.   Open Console  Cancel
            upload_folder(folder+"\\"+obj)
        #If Object is a file, ends with .RCF
        elif (obj.endswith('.RCF')):
            #If Object has not been uploaded
            if (obj not in uploaded):
                #Check if Object is already in bucket
                objs =
                list(bucket.objects.filter(Prefix=bucket_prefix))
                if any([w.key == obj for w in objs])==False:
                    #Upload Object to Bucket
                    s3resource.meta.client.upload_file(folder+"\\"+ob
                    j, bucket_name, bucket_prefix+obj)
                    print(folder+"\\"+obj)
                    #Add names of newly uploaded files to list
                    uploaded.append(obj)
            else:
                print("File already uploaded")
        else:
            print("No Directory or different Filetype")

local_Path="D:\\ImageArchive\\"
local_Log="uploaded.txt"
...
```

Fig.2: Auszug aus dem Uploadskript

Bevor das Konvertierungsskript erstellt werden kann, muss das Bild in der binären Ebene analysiert werden. Anhand der Analyse können die binären Daten manipuliert werden, bis die Pixelwerte in einem zweidimensionalen Array vorliegen und als Bild angezeigt werden können.

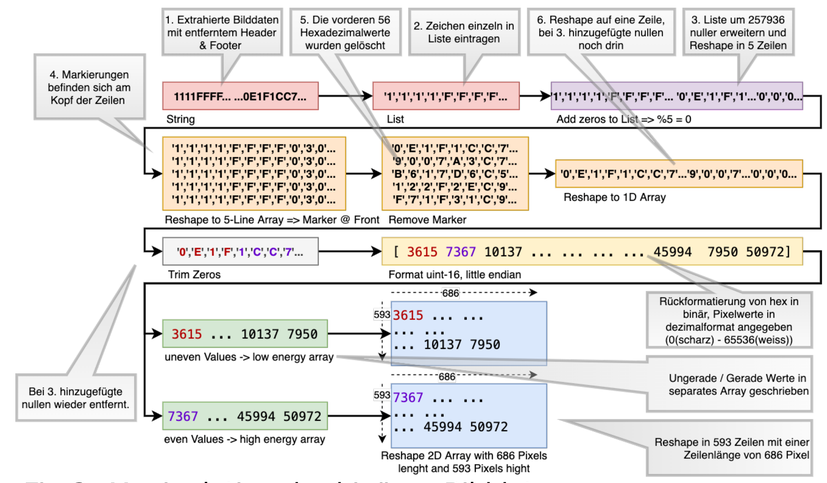

Fig.3: Manipulation der binären Bilddaten

Das Django-Web-Framework bildet das Fundament der auf dem Server gehosteten Webseite. Mit diversen Erweiterungen lässt sich das Framework mit dem verwendeten S3-Objektspeicher und der PostgreSQL-Datenbank verbinden.

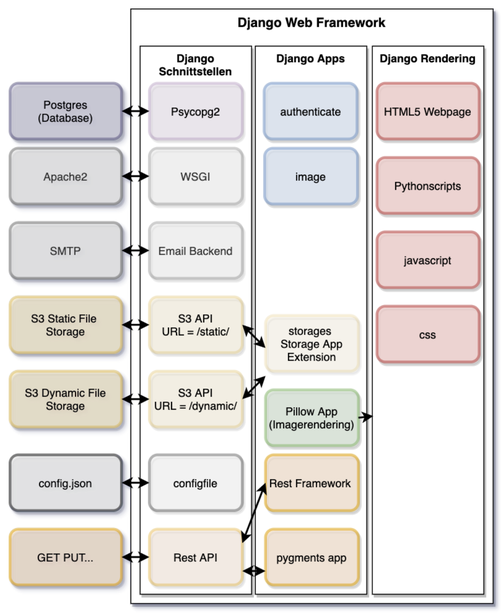

Fig.4: Django-Web-Framework

Ergebnis

Mit dem erstellen Prototyp wird demonstriert, wie ein Server auf Exoscale in Betrieb genommen wird und wie der Verarbeitungsprozess eines Röntgenbildes, vorerst auf dem manuellen Weg, durchgeführt werden kann. Über die Webseite remote-screening.ch werden nach einer Authentifizierung Funktionen angeboten, wie formatierte Bilder einzeln oder gebündelt anzeigen zu lassen.

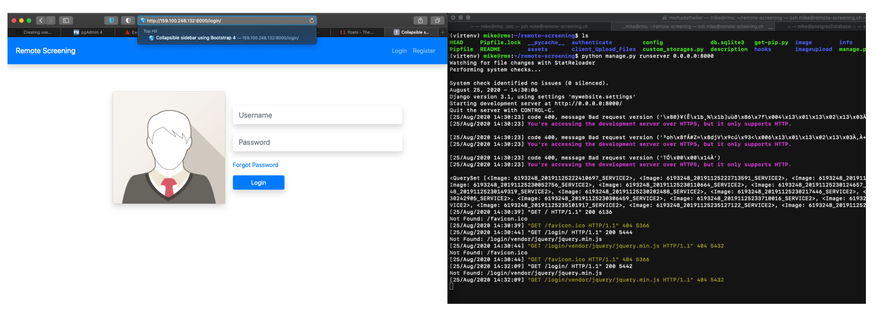

Fig.4: Django Development Server

Studierender: Micha Dettwiler
Projektbetreuer: Prof. Jürg Keller

Ort/Datum: **Zürich, 31.08.2020**

CUSTODIO Airport Security Solutions

MAX C. MEISTER AG

www.fhnw.ch/technik