



Image Processing - Challenge Project

OST — Ostscheizer Fachhochschule

Lukas Ribi, Dominik Castelberg, Pascal Christen

April 22, 2021

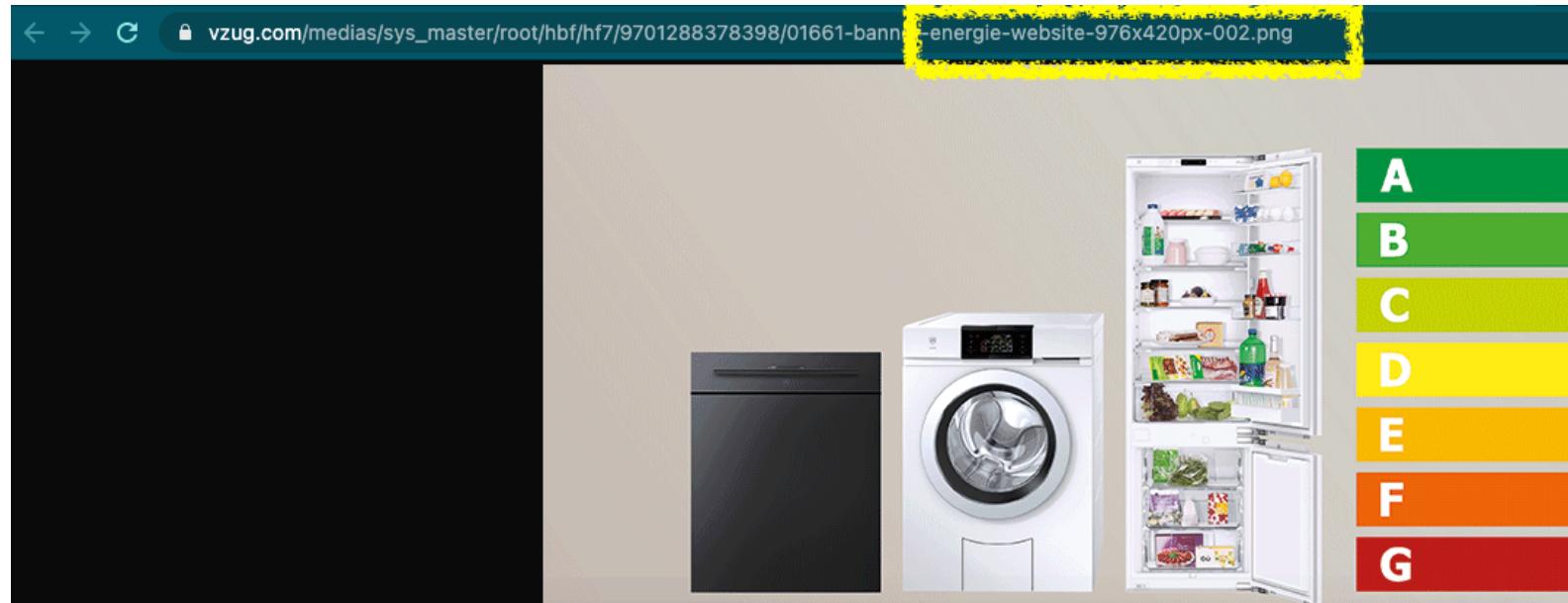
DS1 - Thomas Bocek

1. Introduction

Goal

Problem: Often you need to resize, adjust quality, and convert on demand:

- Responsive Websites
- Upload of an image



Solution: Our ImgProcessing service converts the image **on-the-fly** and provides a simple *REST API*.

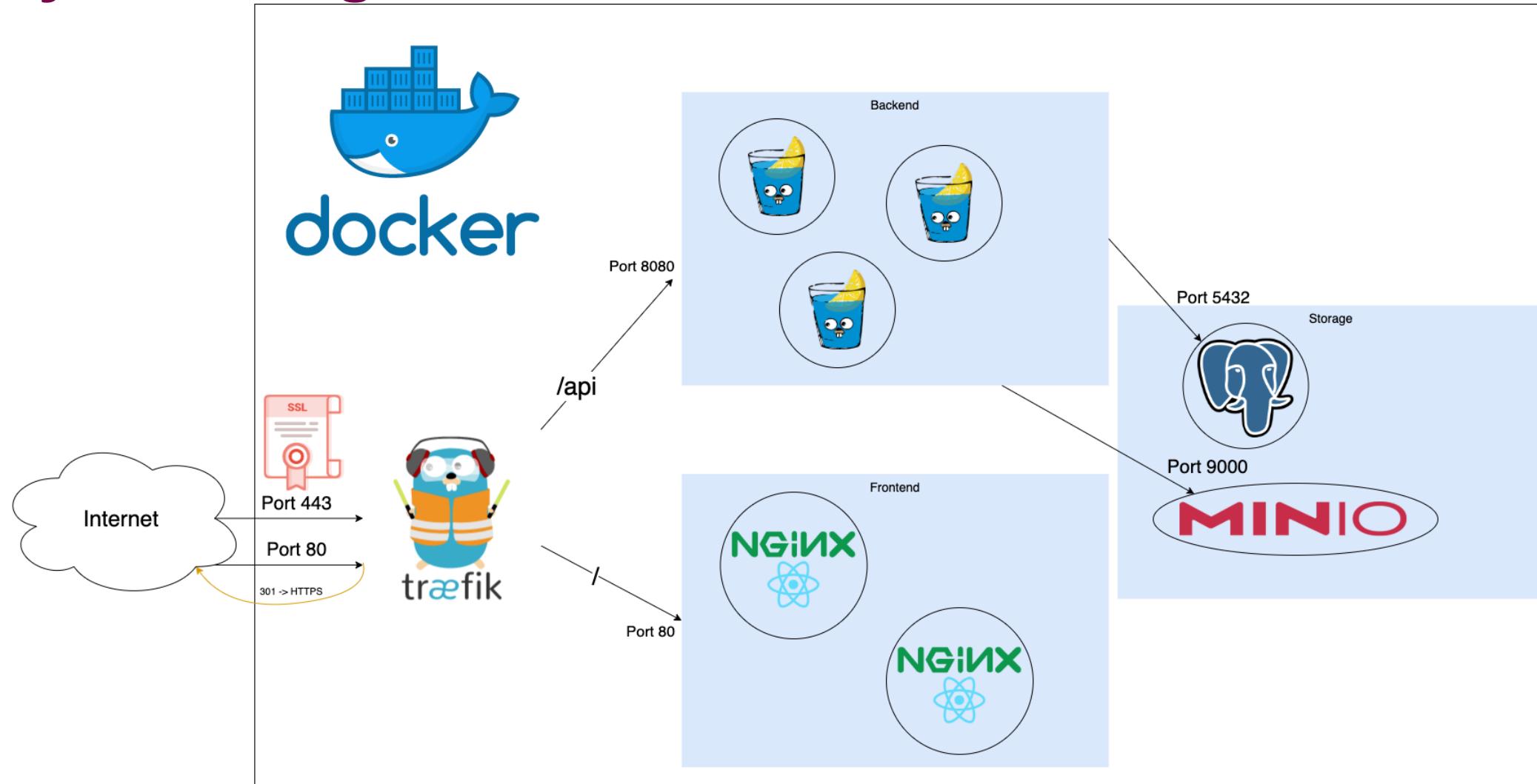
2. Technology

Setup

- Backend
 - `go` (1.16.3)
 - `gin` (1.6.2)
 - JWT
- Frontend
 - `ReactJS` (17.0.2)
 - Material-UI
 - Served by `Nginx` (1.19.9)
- Loadbalancer
 - `Traefik` (2.4.8)
- Storage
 - `PostgreSQL` (13.2)
 - `MinIO - Object Storage` (RELEASE.2021-03-26)
- Präsentation
 - `LaTeX`

2. Technology

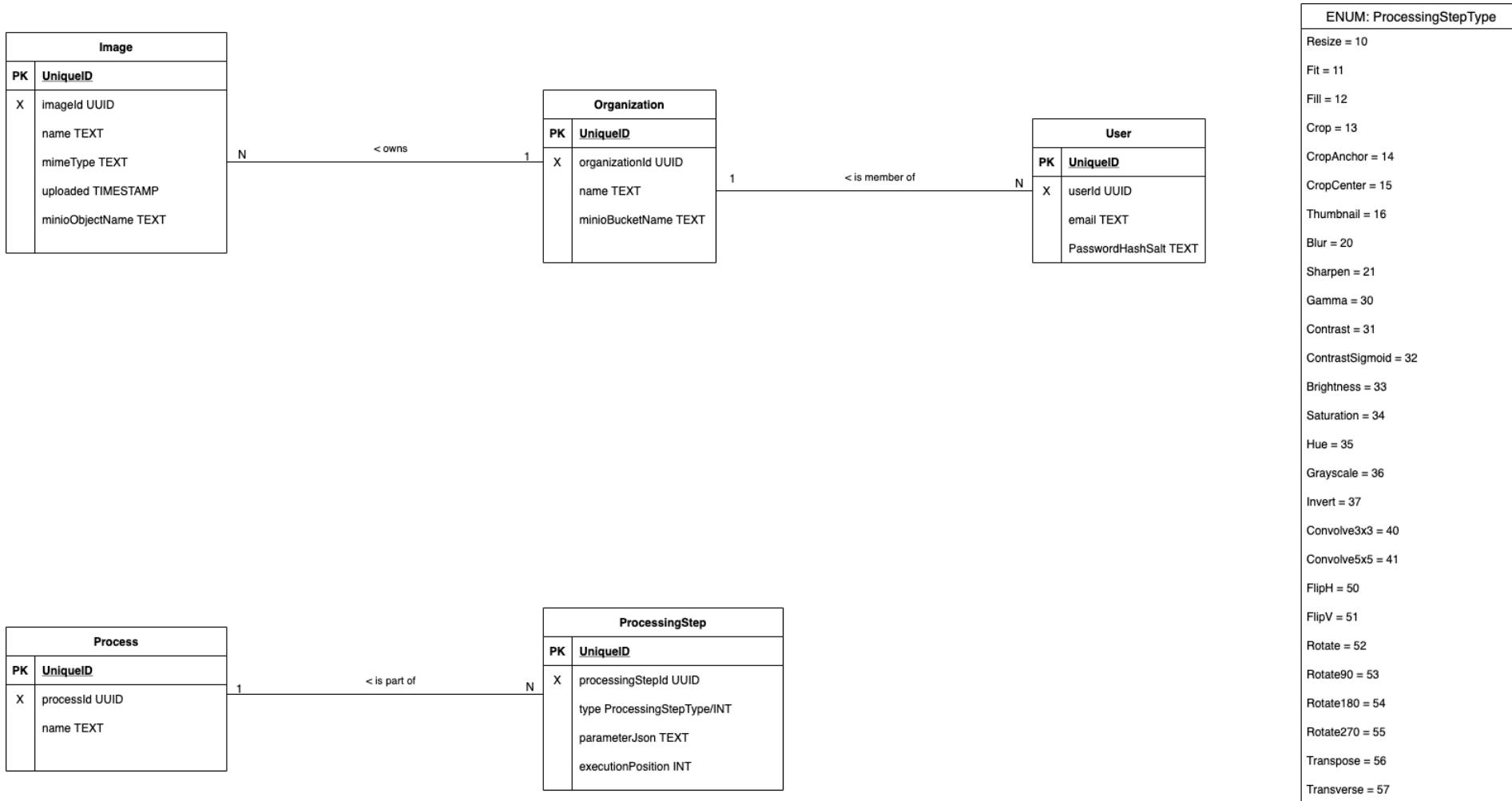
System Diagram



3. Development

Data Model

imgProcessing Entity Relations



3. Development

Github Actions

- Docker Images - builded and versioned by Github Actions
 - Frontend
 - Backend
- Security by scanning for **CVE**

📦 2 packages

📦 frontend latest	⬇️ 276
Published 8 days ago by imgProcessing in imgProcessing/backend	
📦 backend latest	⬇️ 270
Published 8 days ago by imgProcessing in imgProcessing/backend	

3. Development

docker-compose

- Prod and Dev environment
- Restricted access on prod

3. Development

Loadbalancer

Traefik:

- Docker Socket mounted:
 - + Autodiscover
 - - Security
- HTTP -> HTTPS Middleware
- Self-signed cert with **mkcert** for development
- Healthchecks

3. Development

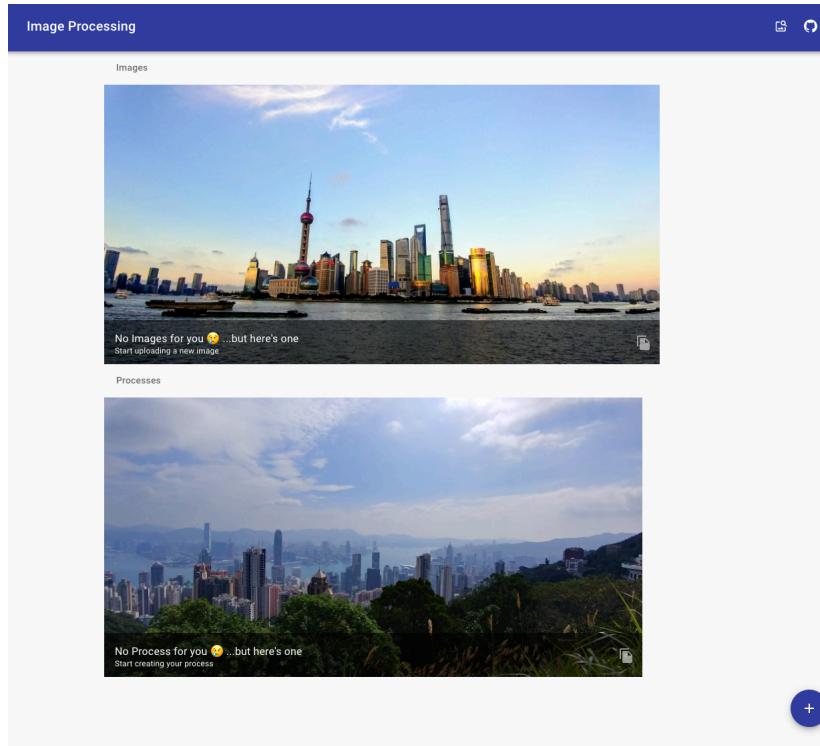
Backend

3. Development

Frontend

ReactJS with Material-UI:

- Basic functions
- Responsive design
- Simple error handling



3. Development

Storage

MinIO:

- High Performance, Kubernetes Native Object Storage
- Compatible with Amazon S3

4. Legal

Fulfillment of Requirements

- Dockerized: Used (images hosted on Github)
- Load balancing: Trafik
- Scalable service: Backend and Frontend
- JWT Authentication: Used
- Latest FOSS: Used

Demo

<https://imgprocessing.pesc.xyz>

End

Source Code