

Crowdsourced E-Accessibility Dashboard

Graduate

Michael Hofmann

Pascal Lehmann

**Introduction:** According to the Swiss Federal Statistical Office (BFS), almost 18% of the population have disabilities as defined under the Disability Discrimination Act (BehiG). The Federal Council of Switzerland announced in December 2023 that they are working on a revision of these regulations also obligating private companies to make their digital services accessible. Therefore, a far greater number of actors will now have to adhere to these rules. Despite existing regulations, there are clear deficits with the accessibility of digital services in Switzerland, especially with government services at the cantonal and municipal levels. However, the extent of these deficits cannot currently be determined. For this reason, our project aims to: record accessibility issues on websites, score these websites based on their accessibility and classify them by region and other categories. The collected data is made available to the public to raise awareness of web accessibility and help website maintainers resolve such issues.

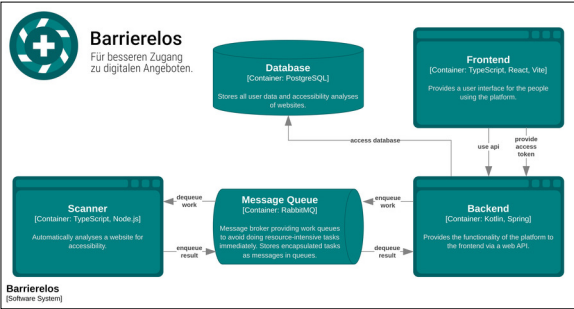
**Result:** The result of this thesis is a software system to automatically determine the accessibility of websites and a web dashboard to highlight these results. Crowdsourcing is employed to collect and categorize websites. The system was developed from scratch but uses the axe-core accessibility engine to search for accessibility issues. The work on this thesis included software engineering, requirements engineering, software architecture, database modeling, wireframing, UX design and the development of a scoring algorithm. The score quantifies the accessibility of websites (from 0 to 100) and was given the name Barrierelos-Score. It is based on the number of accessibility issues found on the website and the severity of these issues.

The frontend is a web application providing an overview of the state of web accessibility in Switzerland and Liechtenstein, that also allows collaboration and moderation. It is a React single-page application with client-side rendering written in TypeScript. The backend implements the business logic that commissions website scans, assesses the results and calculates the score. It is a Spring Boot application written in Kotlin that provides a RESTful web API with OAuth 2.0 offered for authorization. A PostgreSQL database and JPA is used to manage persistence. The scanner receives jobs from the backend via a RabbitMQ message broker, to temporally decouple the resource-intensive scanning tasks. It searches for accessibility issues and returns the findings to the backend via RabbitMQ. The scanner is implemented in TypeScript and runs on Node.js.

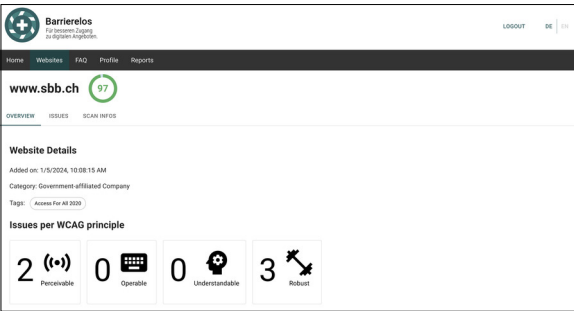
**Conclusion:** Thanks to the containerization and temporal decoupling of resource intensive tasks, the system is easily scalable. The highly modularized architecture is designed so that new requirements

can be implemented with little effort. Large suites of automated unit and integration tests were written to improve quality and maintainability of the code. An agile workflow was fully embraced, with an automated pipeline for building, testing and deployment with GitLab CI/CD, and Flyway for automated database migration, enabling both continuous integration and continuous deployment. Writing the documentation with RST and Sphinx allowed making it available online as a web-documentation through GitLab Pages.

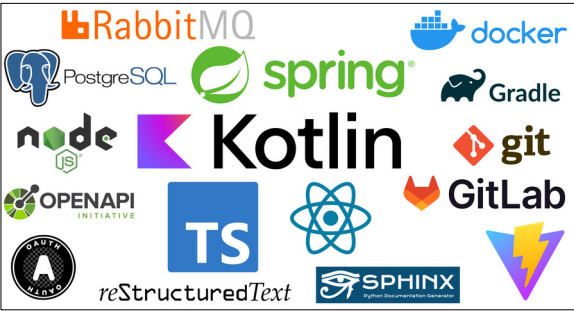
The software architecture as a C4 container diagram  
Own illustration



Screenshot of the website assessment result page  
Own illustration



Technologies used for the project  
Own illustration



**Advisor**  
Prof. Dr. Markus Stolze

**Co-Examiner**  
Markus Flückiger,  
Zühlke Engineering AG,  
Schlieren, ZH

**Subject Area**  
Software