GetraenkeIO: Eine Getränkelagerverwaltung mit Bestandsstatistik für Vereinsheime

Technischer Bericht: CL-TR-2025-42, Juli 2025

Dotzler Martin, Ehrles Andreas, Taach Eduard, Wegerer Nikolas,
Weinhut Justin, Christoph P. Neumann
CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik
Ostbayerische Technische Hochschule Amberg-Weiden
Amberg, Deutschland

***Zusammenfassung*—Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. {⚠The abstract does neither mention a teaching module nor a team/project, it is a summary of the content, thus, the objectives and architecture. Do NOT remove the abstract ⚠, this section is mandatory. You should consider comparing your self-written abstract with the result of a generative AI that summarizes your content after you have written a nearly stable draft version. However, do not use a verbatim copy to replace your abstract, just use generative AI for inspirational purposes.}**

***Index Terms*—D.2.0.c Software engineering for Internet projects, D.2.2.c Distributed/Internet based software engineering tools and techniques**

I. Einleitung und Projektziele

The cs-techrep formatting is adopted both from IEEE [1] and IARIA [2] styles. The cs-techrep LaTeX class is based on IEEEtran class [3]. In addition, be aware of the supplementary IARIA editorial rules [4] ⚠ that provide a beginner-friendly set of further advices. It is recommended to use a grammar tool, e. g., the LanguageTool [5] browser plugin in combination with Overleaf [6].

The title of your paper should not exceed two lines ⚠. In exceptional cases, three lines might be allowed. A four-line-title is absolutely forbidden (hint: use the longer form in the abstract).

For capitalization of titles and section headings, use a web tool like Capitalize My Title ⚠ with the option `Chicago` for capitalization rules by Chicago Manual of Style (CMOS).

The pipe symbol „|“ in the section headings represents alternatives! Choose one and remove the others ⚠. The selectively provided quoted terms are special German alternatives. You may deviate from the structure of this example document and its exemplary section headings.

The problem statement needs to be written from perspective of a subject-matter expert („Fachkonzept“). Like an elevator pitch / mission statement ⚠ and NOT from a technical perspective.

II. Optional: Related Work | State of the Art | Methods | Data Acquisition

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

III. Architekturziele

Provides (1) a visualization of the external systems and users with which the system interacts („Kontextabgrenzung“), (2) the most important technical and organizational preconditions („Rahmenbedingungen“), (3) quality/non-functional requirements („Qualitätsziele“), and/or (4) architectural style design decisions with formative patterns of the solution („Architekturstil“) as well as (5) the applied programming language(s).

IV. Architektur von GetraenkeIO

A. Technologie-Stack

Die GetraenkeIO-Anwendung soll auf einer dreischichtigen Architektur, bestehend aus Frontend, Backend und Datenhaltungsschicht bestehen. Jede Schicht soll von den anderen abgegrenzt auf einem eigenen Container [7] laufen. Das Frontend besteht aus einer React-Anwendung [8]. Als Backend wird eine Python-Anwendung [9] verwendet. Diese benutzt das Web-Framework FastAPI [10] und stellt damit dem Frontend eine RESTful-Schnittstelle [11] zum Datenaustausch zur Verfügung. Zur Datenhaltung wird die relationale Datenbank PostgreSQL [12] genutzt. Die Kommunikation zwischen Datenbank und Backend übernimmt das Framework SQLModel [13], welches Mittels Object-Relational Mapping (ORM) die Datenbanktabellen als Python-Objekte zur Verfügung stellt.

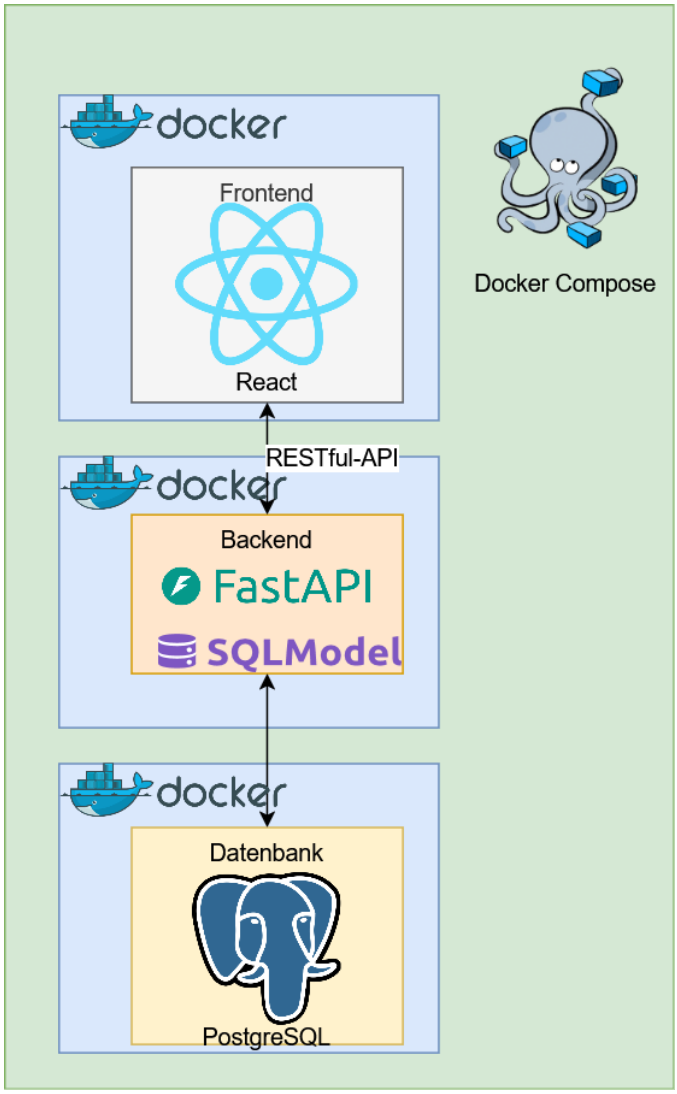

Abbildung 1. Bausteinsicht über den Technologie-Stack und die verwendeten Technologien und deren Zusammenspiel.

1) Vorlage Hr. Neumann: Provides (1) design decisions based on the previously defined requirements and (2) a visualization of the functional structure at top level including relationships („Grobe Zerlegung“), thus, gives an overview on modules, frameworks, and middleware.

In discussions of multi-tier architecture, layer is often used interchangeably – and mistakenly – for tier. They aren't the same. A „layer“ refers to a functional division of the software, but a „tier“ refers to a functional division of the software that runs on infrastructure separate from the other divisions. The Contacts app on your phone, for example, is a three-layer application, but a single-tier application, because all three layers run on your phone.

In discussions concerning multi-tier architecture, the term „layer“ is frequently misused interchangeably with „tier“, despite their distinct meanings. A layer denotes a functional partition within the software, whereas a tier signifies a functional division that operates on separate infrastructure from other divisions/tiers. For instance, the Camera app or Settings app on your phone exemplifies a three-layer application but remains a single-tier application since all three layers run on your phone.

B. Frontend

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

C. Backend

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

D. Datenhaltung

Zur Datenhaltung in der Produktionsumgebung wird eine relationale PostgreSQL-Datenbank verwendet. Die Kommunikation mit dieser findet ausschließlich über das SQLModel-Framework statt. Da ein ORM-Framework verwendet wird, könnte die Datenbank relativ einfach gegen alle anderen von diesem Framework ausgetauscht werden. In der Anfangsphase der Entwicklung wurde diese Möglichkeit genutzt, um SQLite [0] als Entwicklungsdatenbank verwenden zu können und durch die einfache Konfiguration schnell ein lauffähiges System erzeugen zu können.

V. Discussion | Evaluation | Lessons Learned | Impediments

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

VI. Fazit und Ausblick

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Literatur

[1] IEEE. *Conference Template and Formatting Specifications*. 2018. URL: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/conferences/Conference-template-A4.doc.

[2] IARIA. *Formatting Rules*. 2014. URL: http://www.iaria.org/formatting.doc.

[3] Michael Shell. *How to Use the IEEEtran LaTeX Class*. 2015. URL: http://mirrors.ctan.org/macros/latex/contrib/IEEEtran/IEEEtran_HOWTO.pdf.

[4] IARIA. *Editorial Rules*. 2009. URL: https://www.iaria.org/editorialrules.html.

[5] LanguageTooler GmbH. *LangueTool*. URL: https://languagetool.org/overleaf.

[6] Digital Science UK Limited. *Overleaf*. URL: https://www.overleaf.com.

[7] IBM. *Was sind Container?* 2025. URL: https://www.ibm.com/de-de/topics/containers.

[8] Facebook. *React*. [Online]. URL: https://react.dev/.

[9] Guido van Rossum. *Python*. [Online]. URL: https://www.python.org.

[10] Sebastián Ramírez. *FastAPI*. [Online]. URL: https://fastapi.tiangolo.com/.

[11] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.

[12] Andrew Yu und Jolly Chen. *PostgreSQL: Open Source Relational Database*. [Online]. URL: https://www.postgresql.org/.

[13] Sebastián Ramírez. *SQLModel*. 2025. URL: https://https://sqlmodel.tiangolo.com/.

[0] Richard Hipp. *SQLite: Features*. [Online]. URL: https://sqlite.org/features.html.