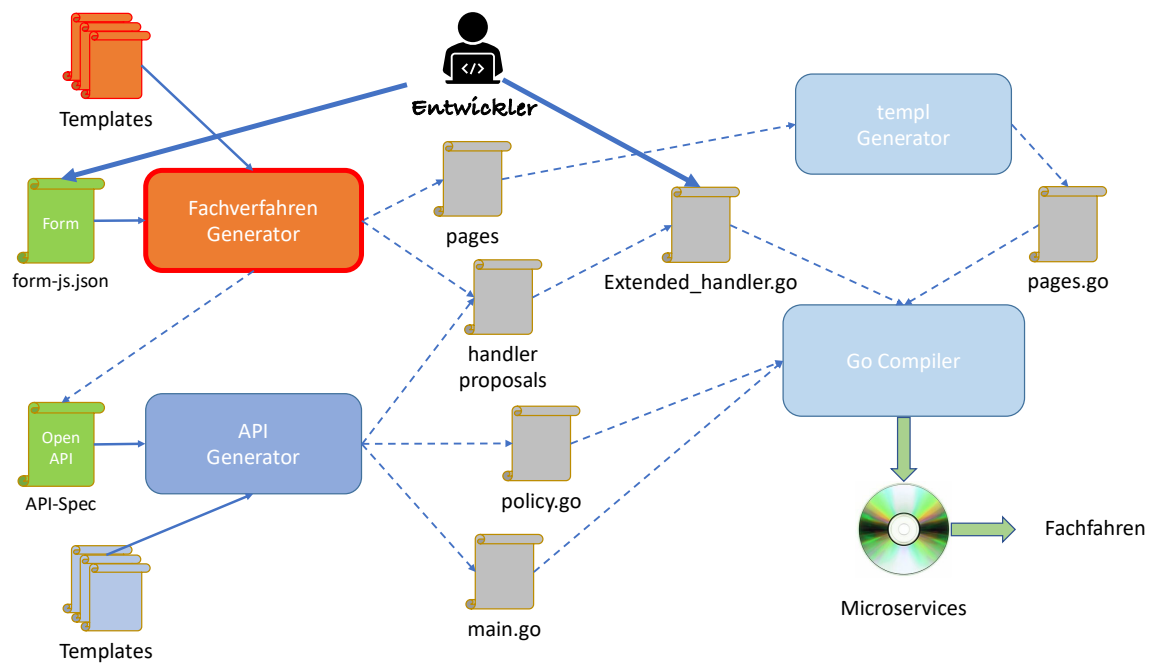


Programmierpraktikum: Moderne verteilte Anwendungen

Aufgabe „Generisches Fachverfahren“

Viele Fachverfahren (fachliche Anwendungen) der Öffentlichen Verwaltung folgen den gleichen, sehr ähnlichen Abläufen (Geschäftsprozessen), bspw. eine Baumfällgenehmigung[6] oder ein Parkausweis. Es wird ein Antrag gestellt, geprüft, genehmigt, und ein Bescheid versendet. Manchmal sind auch weitere Schritte notwendig, bspw. um Gebühren einzufordern. Die verschiedenen Fachverfahren unterscheiden sich vor allem in den Datenstrukturen, die sie benötigen, vor allem für ihre Stammdaten und die Antragsdaten. Darüber hinaus unterscheiden sie sich in den Entscheidungskriterien (Geschäftsregeln), was hier im Rahmen des Projekts nicht weiter betrachtet werden muss. Im Rahmen des Projektes soll ein (minimales) „generisches Fachverfahren“ entwickelt werden, bei dem die anzupassenden Code-Teile durch Generatoren aus Modellen erzeugt werden sollen. Die verteilte Anwendung soll ein web-basiertes Portal für den Bürger und vor allem das web-basierte Fachverfahren für den Sachbearbeitenden umfassen. Letzterer kann die Stammdaten der Bürger:innen pflegen, Anträge sichten, prüfen und genehmigen, wobei er durch Aufgabenlisten unterstützt werden soll.

Technisch soll ein modernes, verteiltes System bestehend aus Microservices (Dienste) implementiert werden. Die Microservices sollen miteinander über APIs (REST) kommunizieren. Die Web-Oberfläche soll als modernes Hypermediasystem (erweitertes HTML [7]) realisiert werden. Die Daten sollen in einer SQL/NoSQL-Datenbank persistiert werden. Der Code soll möglichst komplett generiert werden, aber auch erweiterbar sein, um bspw. zusätzliche Funktionalität realisieren zu können. Dabei sollen vorhandene Generatoren oder deren Ansätze/Ideen genutzt und bei Bedarf erweitert werden, beispielsweise um aus einer OpenAPI-Spezifikation den Basis-Code für einen Microservice zu generieren, oder um aus einer Formular-Spezifikation, die mit einem Form-Editor[3] erstellt wurde, einen Service zur HTML-Ansicht zu generieren.



Als Technologie-Stack sollen die Programmiersprache Go mit den Bibliotheken Echo-Webserver und HTMX sowie die Datenbank CockroachDB genutzt werden, um eine Nachnutzung im Low-Code-Kontext bei Fraunhofer FOKUS zu gewährleisten. Als zusätzliche Rahmenbedingung sollen die Best-Practices des „Clean Code“ [1] und der „Clean Architecture“ [2] umgesetzt werden. Der Generator soll an einem verständlichen Beispiel demonstriert sowie die generierten Microservices mit BDD-basierten Tests [5] auf Basis von dogdog getestet werden.

Der finale Umfang des Projektes orientiert sich an der Größe des Projekt-Teams.

[1] Jürgen Lampe: Clean Code für Dummies, Wiley, 2020.

[2] Robert C. Martin: Clean Architecture, mitp, 2018.

[3] BPMN.io: form-js, <https://bpmn.io/toolkit/form-js/> .

[4] Go, <https://go.dev/> und <https://pkg.go.dev/> .

[5] Echo <https://echo.labstack.com/docs>

[6] Baumschutzverordnung: <https://www.berlin.de/sen/uvk/natur-und-gruen/naturschutz/baumschutz/> .

[7] Hypermedia Systems: <https://hypermedia.systems/> .