

An abstract geometric design on the left side of the slide. It features a dark blue background with various geometric shapes and patterns. A white circle is positioned near the top left. Below it, a light blue semi-circle is visible. To the right of the semi-circle, there are concentric circles. Further down, there are several parallel lines forming a series of nested rectangles. The design is composed of various shades of blue, purple, and pink.

# AUSGABENTRACKER



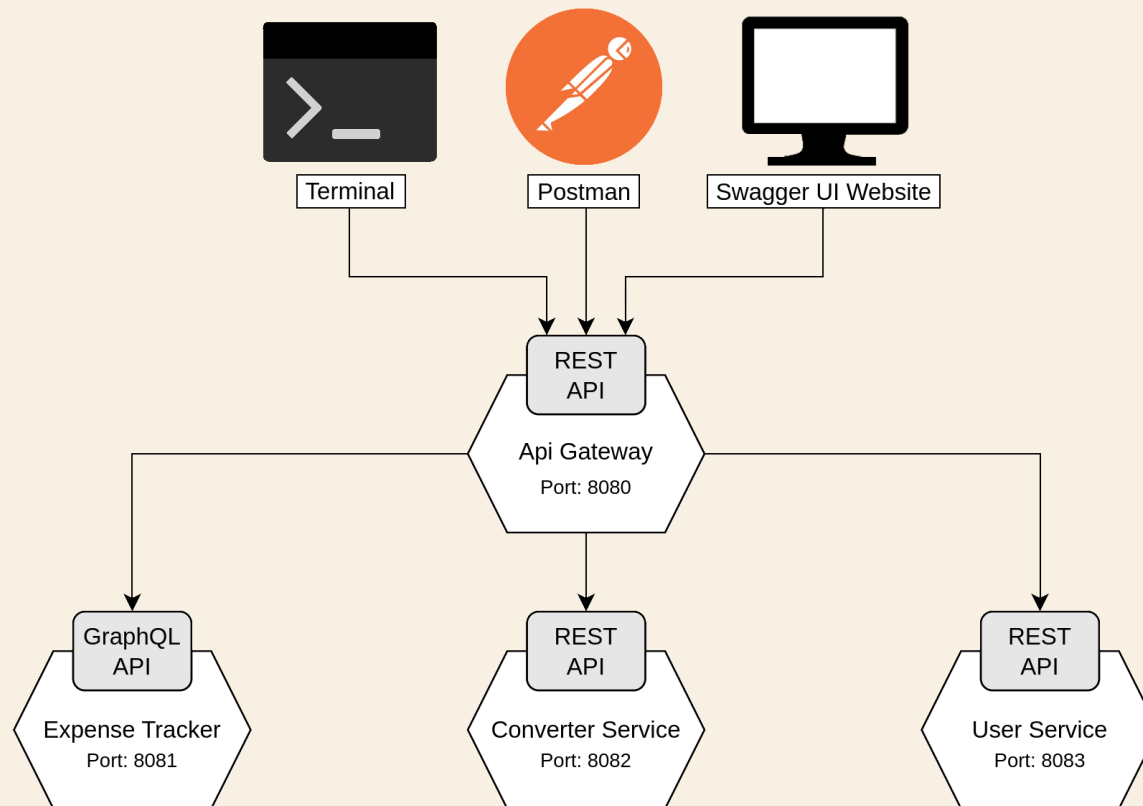
# AGENDA

- Was ist unsere Webanwendung?
- Architektur
- User Service
- Converter Service
- Expenses Service
- Demonstration

# WEBANWENDUNG

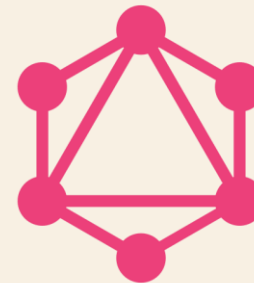
- Backend Entwicklung ohne Frontend
- User Verwaltung
- Ausgaben tracken
- Währungen umrechnen
- Nutzung über Postman, Konsole und Swagger UI
- Implementierung über VS Studios

# ARCHITEKTUR



# TECHNOLOGIEN

- Spring Boot 3.4.2
- Maven
- Rest
- GraphQL





# REST API USERSERVICE

# REST API USER

## **Rest API User Service**

- RESTful API für User zur Verwaltung von Benutzerdaten
- Entwickelt mit Spring Boot und Java 17
- Er bietet CRUD-Funktionalitäten

# CRUD

- Create / POST
- Read / GET
- Update / PUT
- Delete / DELETE



# FEHLERBEHANDLUNG IM USER SERVICE

- **Ungültige oder unvollständige Anfragen**  
(z. B. fehlende Felder im Request-Body)
- **Ressourcen nicht gefunden**  
(z. B. User mit ID existiert nicht)
- **Konflikte und doppelte Einträge**  
(z. B. User mit gleicher E-Mail existiert bereits)
- **Serverfehler**  
(z. B. unerwartete Ausnahmen)

Status Code	Bedeutung
400 Bad Request	Ungültige/Unvollständige Anfrage
404 Not Found	Ressource nicht gefunden
409 Conflict	Konflikt (Doppelte Einträge)
500 Internal Server Error	Unerwarteter Fehler

# CONVERTER SERVICE

- REST-API für Währungsumrechnungen
- Entwickelt mit Spring Boot und Java 17
- Verwendet die CurrencyLayer API für Echtzeit-Wechselkurse
- Ermöglicht die Umrechnung einzelner Währungen oder mehrerer Währungen gleichzeitig
- Einfache Integration in andere Systeme durch standardisierte API-Endpunkte

# ARCHITEKTUR

- **REST-API Controller (ConverterController.java)**
- Stellt Endpunkte zur Währungsumrechnung bereit
- Wichtige Routen:
  - **/api/converter/supported-currencies** → Liste aller Währungen
  - **/api/converter/convert?from=USD&to=EUR&amount=100** → Einzelne Umrechnung
  - **/api/converter/convertToMultipleCurrencies?from=USD&amount=100** → Mehrere Währungen

# ARCHITEKTUR

- **Service-Schicht (CurrencyService.java)**
- Sendet Anfragen an die CurrencyLayer API
- Berechnet Wechselkurse & verarbeitet API-Antworten
- Fehlerbehandlung für ungültige Eingaben

# KOMPONENTEN

- **Externe API (CurrencyLayer)**
- Liefert Echtzeit-Wechselkurse
- Kostenpflichtig mit einer begrenzten **kostenlosen Version**
- Datenabruf über **HTTP-Requests**

# VORTEILE / NACHTEILE

- Einfache Nutzung
- Unterstützt Einzel- & Mehrfach-Umrechnungen
- Modular aufgebaut und erweiterbar
- Aktuelle Wechselkurse
- Gute Integration in andere Systeme
- API-Schlüssel direkt im Code
- Keine Fehlertoleranz bei API-Ausfällen
- Keine Caching-Funktion
- Abhängigkeit von CurrencyLayer (Abo-Modell)

# EXPENSES SERVICE

- Erstellt in Grahql
- Mutation und query
- Flexibilität bei den abfragen

# ALL EXPENSES

- Zeigt alle Benutzerausgaben
- User erforderlich
- Überprüfung des Parameters
- Fehlermeldung bei ungültiger user



# ADD EXPENSE (MUTATION)

- Zeigt Benutzerausgaben an
- Benutzer-ID erforderlich
- Fehlermeldung bei Fehlern

# DELETE EXPENSE (MUTATION)

- Löscht Ausgaben basierend auf ID
- Nur eine ID erforderlich
- Komplette Löschung der Ausgabe
- Einfache Handhabung bei Fehlern

# UPDATE EXPENSE (QUERY)

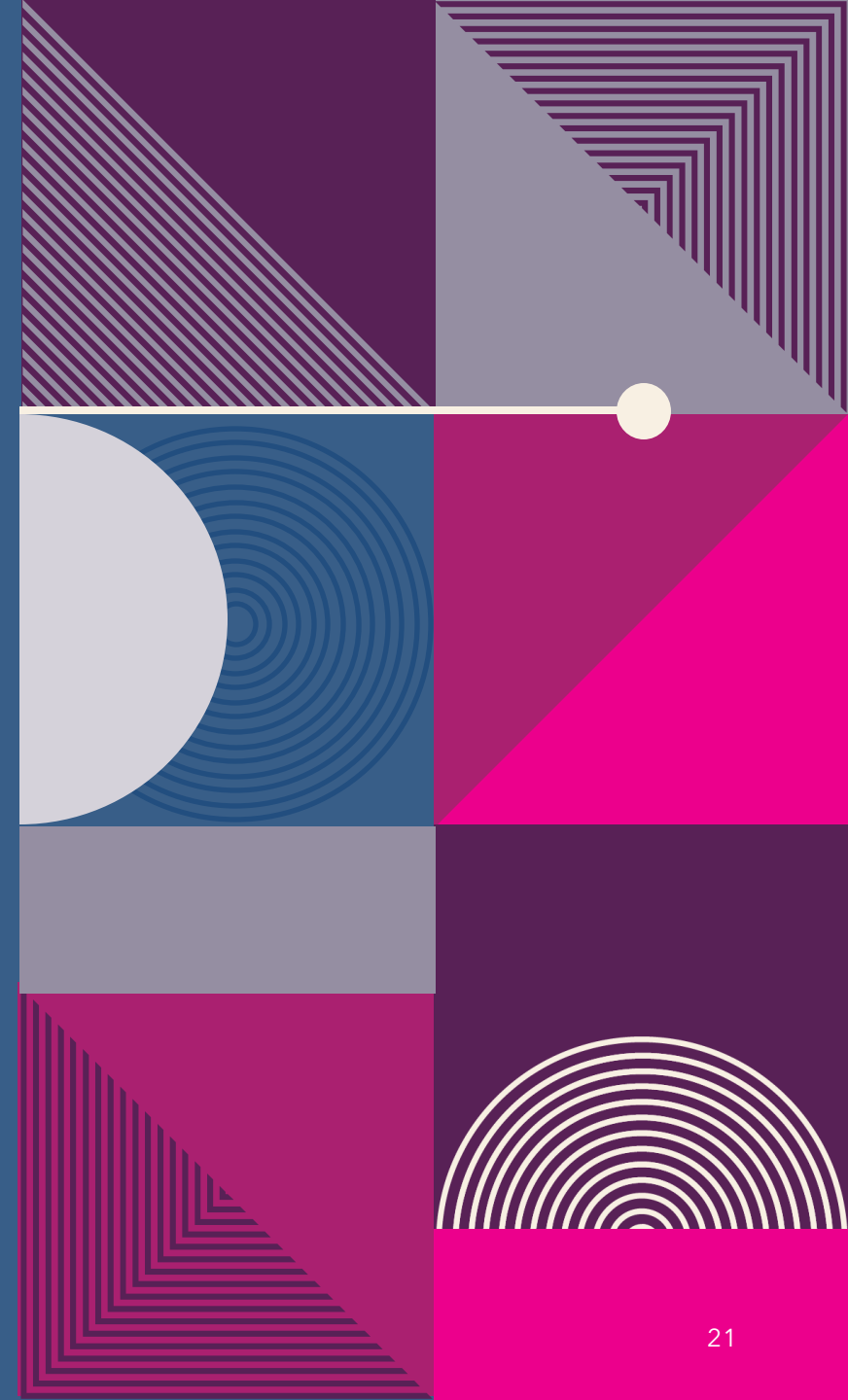
- Aktualisiert bestehende Ausgaben
- Benutzer-ID und Ausgabe-ID erforderlich
- Einfache Fehlerkorrektur ohne Löschen



# API-GATEWAY UND DEMONSTRATION

# API-GATEWAY

- Kernstück unsere Anwendung
- Schnittstelle zur Außenwelt (REST)
- Über Konsole, Postman und Swagger UI zugänglich
- Teilt und übersetzt Anfragen in REST und GraphQL requests





**VIELEN DANK FÜR IHRE  
AUFMERKSAMKEIT**