LEUPHANA
UNIVERSITÄT LÜNEBURG

Prüfungsleistung der Veranstaltung "Verteilte Anwendungssysteme"

Hintergrund

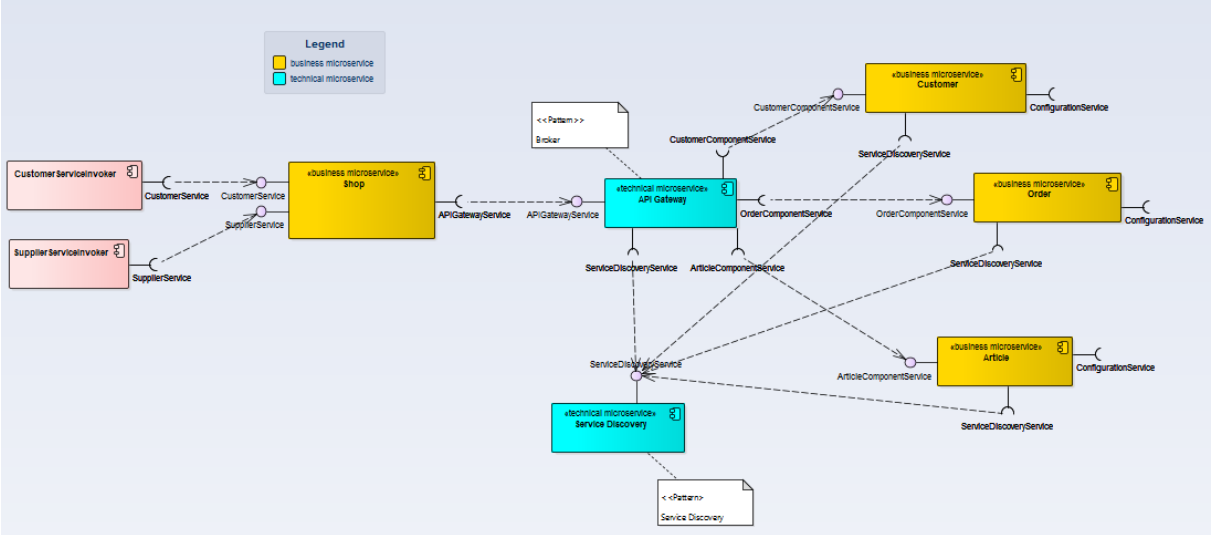
- Idee
  - Aufteilung eines fachlichen/technischen Monolithen in mehrere fachliche MicroServices mit Infrastruktur

    ➔ wg. Skalierung !!!

- Ein MicroService besteht aus
  - Einer Component
  - Mehrere Connector (Stub / Skeleton)
  - Einer Configuration

Entwicklungsergebnisse

- MicroServices
  - 4 fachliche MicroServices (Customer, Order, Article bzw. Shop)
  - 3 Infrastrukturelemente (Discovery Service, API Gateway)

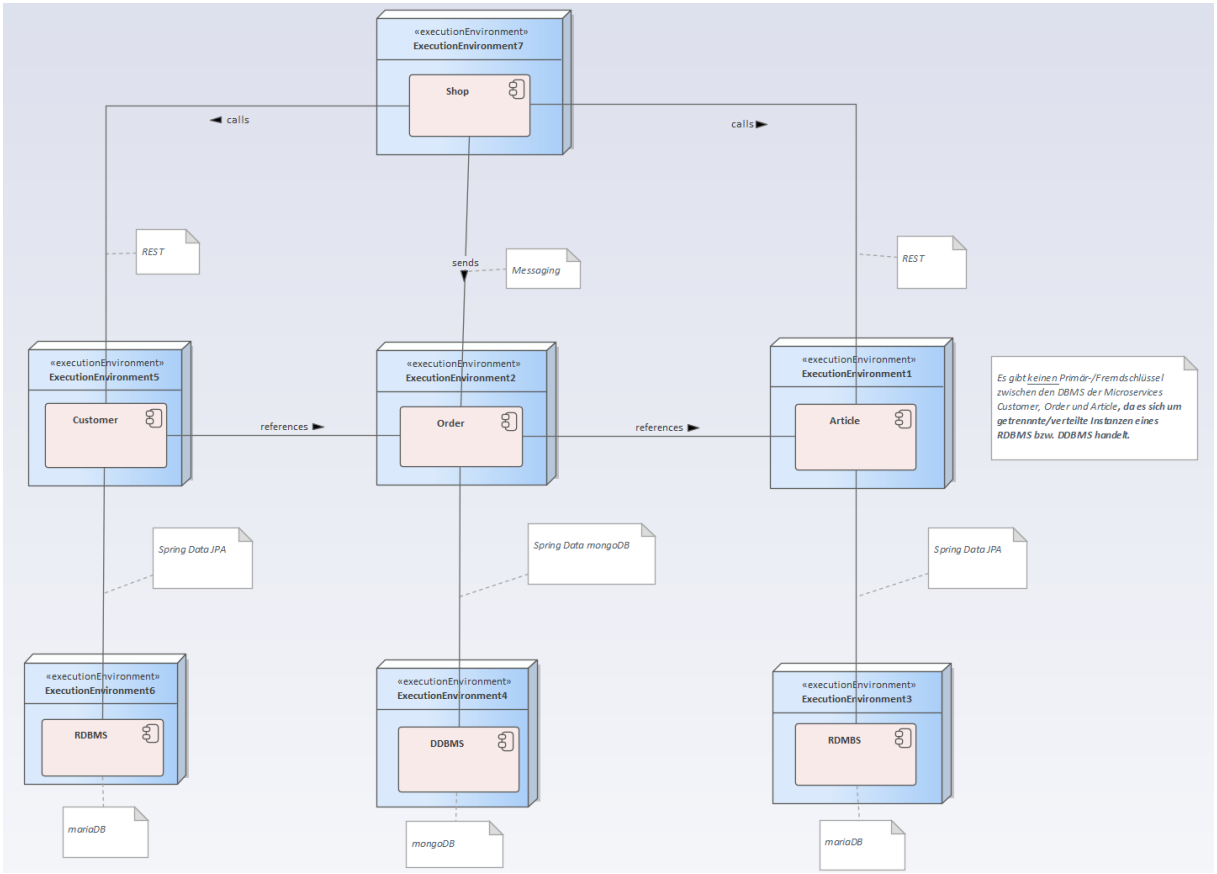


Überblicksdiagramm (siehe Enterprise Architect-Projekt)

- Pro fachlichen MicroService ein separates Eclipse-Projekt mit folgendem Projektlayout:
  - src

        - component
            - structure
            - behaviour
                - *ComponentName*Service
        - connector
            - *ForeignComponentName*RestConnectorRequester
              or
            - *ForeignComponentName*JMSConnectorSender
            - *ComponentName*RestConnectorProvider
              or
            - *ComponentName*JMSConnectorSender
            - *ComponentName*SpringDataConnectorRequester
        - configuration
            - *ComponentName*Configuration
    - test
        - component
            - behaviour
                - *ComponentName*ServiceTest
        - connector
            - *ForeignComponentName*RestConnectorRequesterTest
            - *ComponentName*RestConnectorProviderTest
              or
            - *ForeignComponentName*JMSConnectorSenderTest
            - *ComponentName*JMSConnectorReceiverTest
            - *ComponentName*SpringDataConnectorRequesterTest

Die nachfolgende Abbildung zeigt das Deploymentdiagramm in dem die MicroServices und DBMS (Component) sowie die jeweiligen Verbindungen (Connector) dargestellt sind. Aus Übersichtsgründen fehlt das Deploymentdiagramm das API-Gateway und die Service-Discovery (siehe Komponentendiagramm oben).



- Zu verwendende Technologien
  - verpflichtend
    - Für MicroServices
      - Spring Boot
      - Spring Data JPA
      - Spring Data mongoDB
      - Spring Cloud
        - REST Client (z.B. Feign)
        - Spring REST Service
        - Spring JMS Service
        - Discovery Service (z.B. Eureka oder Consul)
        - API Gateway Service (z.B. Eureka)
    - Für Tooling
      - Maven
      - JUnit 5
  - Optional
    - Für Tooling
      - z.B. Postman zum Testen des REST-Schnittstelle
      - Statt Eclipse nun Spring Tool Suite (STS)
      - Spring Initializr

- Empfohlene Vorgehensweise
    - Aufteilen der Eclipse Shop-Komponente (Monolith) in 4 lose gekoppelte fachliche Komponenten (MicroService)
    - Umstellung aller fachlichen MicroServices auf Spring Data JPA bzw. mongoDB
    - Einfache Spring Boot Application erstellen und ausprobieren
    - Einen fachlichen MicroService erstellen und ausprobieren
    - Alle fachlichen MicroServices integrieren und ausprobieren
    - Getrennt jeweils die technischen MicroService erstellen und ausprobieren (Erst DiscoveryService, dann API Gateway, dann Logging&Tracing)

➔ für jeden Schritt in der eine Technologie verwendet wird, empfiehlt es sich ein lauffähiges Beispiel aus dem Internet herauszusuchen!

Projektteam

- 3 Studierende

Dokumentation

- Die MicroServices sind als gezipte Eclipse-Projekte in myStudy pro Gruppe hochzuladen.

Abgabetermin

- Abgabe der Entwicklungsergebnisse bis 24.03.2024 einschließlich. (Geänderter Termin gegenüber myStudy !!!)

Präsentation

- Präsentation der lauffähigen Entwicklungsergebnisse am 25.03. bzw. 26.03.2024, in der Fragen zu den verwendeten Technologien gestellt werden.