|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Ziel der Gruppenarbeit „Handelsplattform Erweiterung I“ im Rahmen dieser Vorlesung ist es, eine moderne, wettbewerbsfähige Internet-Handelsplattform zu entwickeln, welche als Software-Architektur auf Microservices setzt. Jeder Microservice soll dabei genau eine einzige Aufgabe erfüllen und über genau definierte Schnittstelen erreichbar sein bzw. mit anderen Microservices über die angebotenen Schnittstellen kommunizieren.**  **Denkbar wären zum Beispiel folgende Microservices:**   * **Shopping-Microservice** * **Bezahl-Microservice** * **Rating-Microservice** * **BauernladenAProduktkatalog-Microservice** * **WeingutBProduktkatalgo-Microservice** * **Storno-Microservice** * **Währungsrechner-Microservice**   **Fokus: Integration von „elektronischen (Geschäfts)-Prozessen“ und Microservices. Nehmen Sie in Ihren Ausführung auch Bezug auf die im Artikel „Microservices a definition of this new architectural term“**  **(**[**http://martinfowler.com/articles/microservices.html**](http://martinfowler.com/articles/microservices.html)**) beschriebenen Konzepte.**  **Teambezeichnung & Teammitglieder:** | | | | | |  | **Vorname Nachname** | | | **Aufgabengebiet** | | **Name 1** | | **Obendrauf Stefan** |  | **1 + 2** | | **Name 2** | |  | | **Name 3** | |  | | **Name 4** | |  | | **Name 5** | |  | | **Name 6** | |  | | **Name 7** | |  |  |  | | **Name 8** | |  |  |  | |  | |  |  |  | |  | |  |  |  | | 1. **Analyse: Machen Sie sich mit dem Ausgangs-Source-Code „SolHandelsplattform V.01“ vertraut. Publizieren Sie die beiden Services „BlackFriday“ und „IEGEasyCreditCardService“ in die Microsoft Azure Cloud und Testen Sie die Funktionalität (0 Punkte)** | | | | | | **Both Microservices hosted. Links:**  <http://iwi17-blackfriday.azurewebsites.net/swagger/>  <http://iwi17-easycreditcardservice.azurewebsites.net/swagger/> | | | | | | 1. **2 weitere Microservice Produktkataloge: Erstellen Sie ein Microservice, welches eine Liste von Produkten anbietet. Der Inhalt der Liste soll dabei aus einem „microservice local datastore“ kommen – (Decentralized Data Management). Ersetzen Sie die hard codierten Werte im BlackFriday/ProductList-Controller durch den Aufruf des soeben erstellen Services. Ein weiterer Produktkatalog-Service soll Produkte aus einem Text File auf einem FTP-Server auslesen und zur Verfügung stellen. Der Produktkatalog soll ebenfalls lokal, angereichert um Verkaufsinformationen, im BlackFriday-Service zur Verfügung stehen(10 Punkte):** | | | | | | **URLs:**  <http://iwi17-productservicefile.azurewebsites.net/swagger/>  <http://iwi17-productservicelocal.azurewebsites.net/swagger/> | | | | | | 1. **Skalierung, Ausfallssicherheit und Logging (Design for failure) für CreditPaymentService. Detailsbeschreibung: Publizieren Sie das Service „IEGEasyCreditCardService“ mehrfach in die Cloud und rufen Sie die Services aus dem CashDesk-Controller im „Round Robin“ Stil auf. Falls es beim Aufruf eines Services zu einem (Kommunikations)fehler kommt, soll es eine Retry-Logik geben, außerdem soll der aufgetretene Fehler protokolliert werden. Nach n (config-Wert) erfolglosen Versuchen, soll das nächste Service aufgerufen werden. (10 Punkte)** | | | | | |  | | | | | | 1. **(theoretische) Überlegungen zum Einsatz von Asynchronen Kommunikationsstilen in der Handelsplattform (10 Punkte)** <http://soapatterns.org/design_patterns/reliable_messaging><http://soapatterns.org/design_patterns/event_driven_messaging> **http://soapatterns.org/design\_patterns/asynchronous\_queuing** | | | | | |  | | | | | |  | | | | | | 1. **Schreiben Sie ein zusätzliches „Paymentservice“. Dieses Payment-Service soll sowohl JSON, XML-Nachrichten als auch Nachrichten im Format CSV verarbeiten und erzeugen können. Orientieren Sie sich an dem Pattern - http://soapatterns.org/design\_patterns/content\_negotiation  (10 Punkte)** | | | | | |  | | | | | |  | | | | |  |  | | --- | | 1. **(theoretische) Überlegungen zu einem PaymentService-Broker. Dieses Service soll zwischen Shops und Payment-Services „vermitteln“.** <http://serviceorientation.com/serviceorientation/service_loose_coupling><http://serviceorientation.com/serviceorientation/service_discoverability> **http://soapatterns.org/design\_patterns/data\_format\_transformation**   **http://www.enterpriseintegrationpatterns.com/patterns/messaging/ CanonicalDataModel.html**  **http://soapatterns.org/design\_patterns/canonical\_schema**  **http://soapatterns.org/design\_patterns/canonical\_protocol  (10 Punkte)** | |  | |  |  |  | | --- | | 1. **Webhook-Subscriber mit Worfklow-Tracking - (Dezember)** | |  | |  | |