Test Makroarchitektur, Teil 2

==========

(Schwerpunkt der Überprüfung sind Aufgaben 6-10 des Übungszettels, also eigenständiges Entwerfen/Codieren)

Aufgabenstellung "Buchhandlung"

Es soll der Use Case einer Buchbestellung umgesetzt werden:

1. es wird eine Buchbestellung angenommen
2. danach wird der Buchdruck passend zur Bestellung durchgeführt
3. sobald der Buchdruck abgeschlossen ist, kann das Buch "abgeholt" werden.

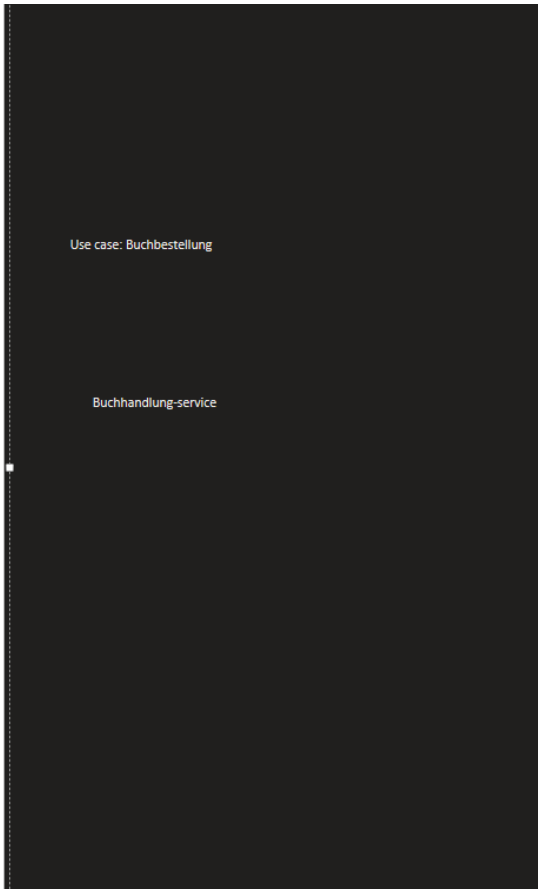
Bitte in der folgenden Detailbeschreibung die Namen in Anführungszeichen auch genau so im Programm benennen!

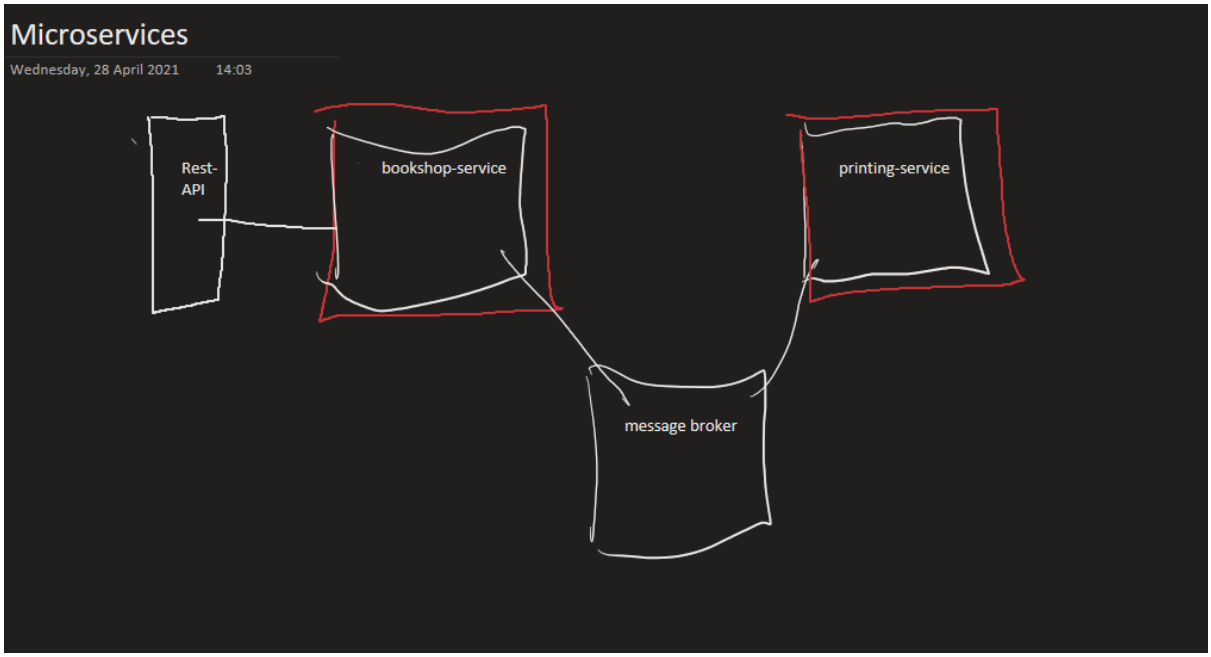
Details:

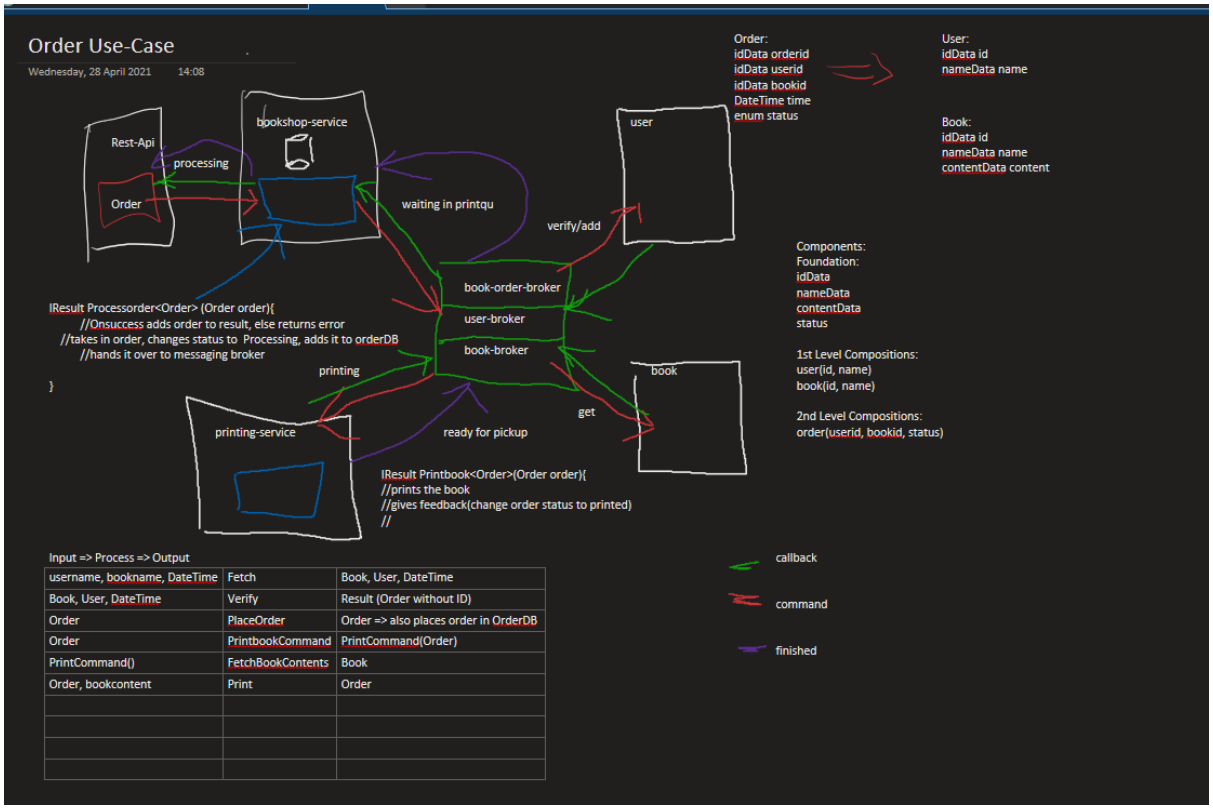
- Über eine REST-API kann beim "buchhandlung-service" die Buchbestellung initiiert werden. Dazu wird ein entsprechendes "Bestellung"-Objekt angelegt und in der Datenbank gespeichert.
- Das "buchhandlung-service" übermittelt in weiterer Folge ein "BuchBestelltEvent" (dieses Event hat genau diesen Namen) an den Message Broker (idealerweise RabbitMQ Message Broker, wie bereits in Aufgabe 6 "Cargo" verwendet -> gern dort nachsehen; zur Erinnerung: auf Moodle findet ihr auch das Video zur Inbetriebnahme von Cargo mit RabbitMQ). Nach erfolgreichem Versand des Event wird am zugehörigen "Bestellung"-Objekt ein Status "BESTELLT" gesetzt.
- Das "druckerei-service" reagiert auf das "BuchBestelltEvent" (erstellt und gesendet vom "buchhandlung-service") und führt den Buchdruck durch.
- Sobald der Buchdruck im "druckerei-service" durchgeführt wurde, übermittelt das "druckerei-service" das "BuchGedrucktEvent" über den Message Broker. Über den Message Broker wird dieser Event an entsprechende Konsumenten (in diesem Fall lediglich das "buchhandlung-service") weiter geleitet.
- Bei Erhalt des "BuchGedrucktEvent" setzt der "buchhandlung-service" den Status am zugehörigen "Bestellung"-Objekt auf "ABHOLBEREIT"

Zusätzliche Rahmenbedingungen:

- zuerst ist die Architektur zu entwerfen; dazu sind geeignete Diagramme bzw. Handskizzen zu erstellen







Input => Process => Output

username, bookname, DateTime	Fetch	Book, User, DateTime
Book, User, DateTime	Verify	Result (Order without ID)
Order	PlaceOrder	Order => also places order in OrderDB
Order	PrintbookCommand	PrintCommand(Order)
PrintCommand()	FetchBookContents	Book
Order, bookcontent	Print	Order