RabbitMQ

Broker-Style

Was wir bis jetzt gesehen haben

- REST Architecture Style: Ressourcenorientiert denken
  - JSON-over-HTTP
- GRPC
  - Remote-Procedure-Call, Streaming
- Beide haben zeitliche Abhängigkeiten → andere Komponenten muss da sein
- Übung 4: Broker Style
- Vollständige Entkoppelung

Übung 4

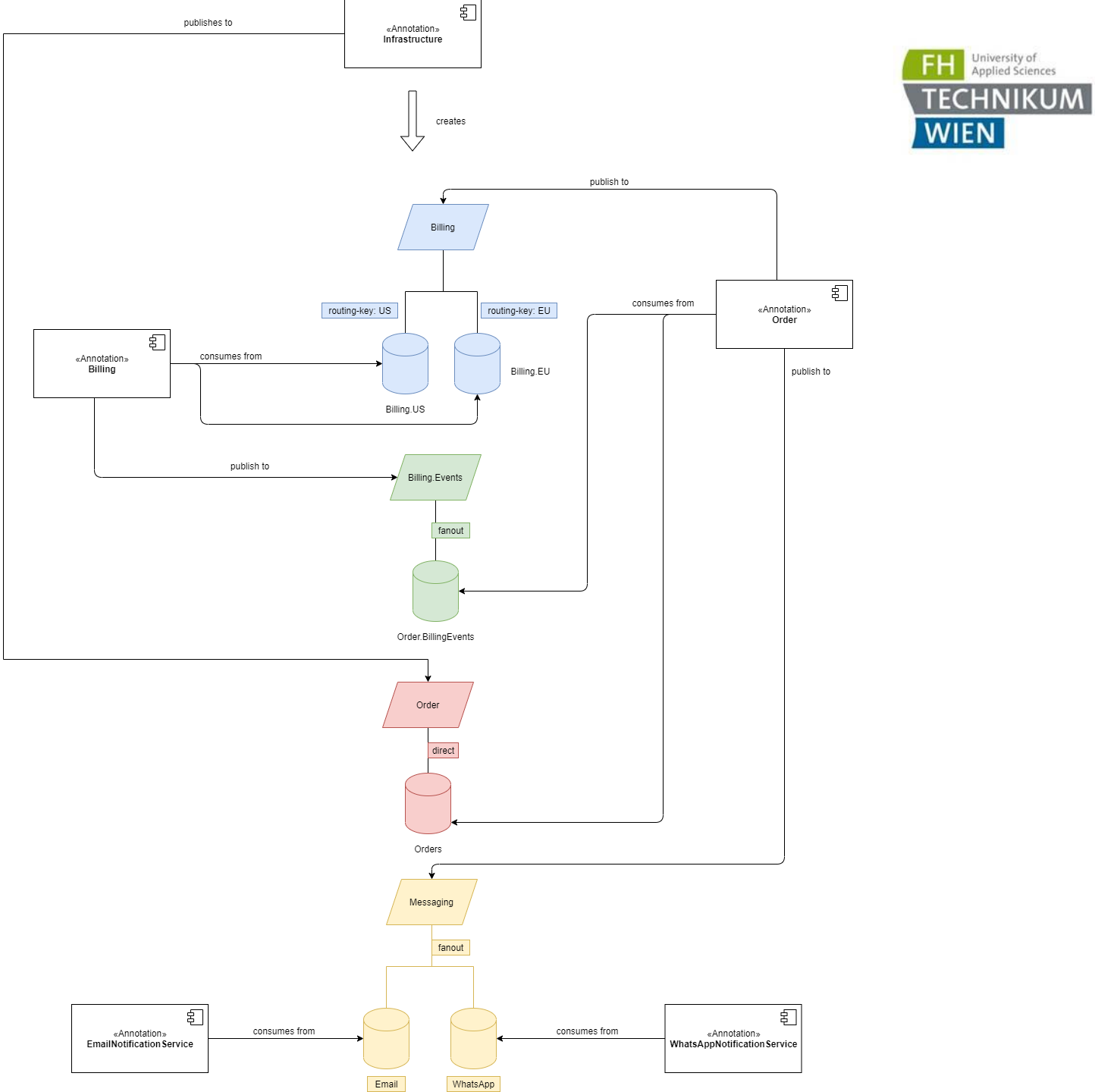
- Vorlage im Ordner `OrderManagement`
- Beispiel zeigt
  - Beispielverbindung
  - Nutzung der Library „EasyNetQ“ mit dem AdvancedBus
- Wichtig
  - Es wird die lokale Docker RabbitMQ Instanz genutzt
  - Admin Interface: http://localhost:15672

Entitäten

- Nutzen Sie die Library `Entities` – diese soll alle in den folgenden Folien beschriebenen Entitäten enthalten
- Nutzen sie https://github.com/protobuf-net/protobuf-net

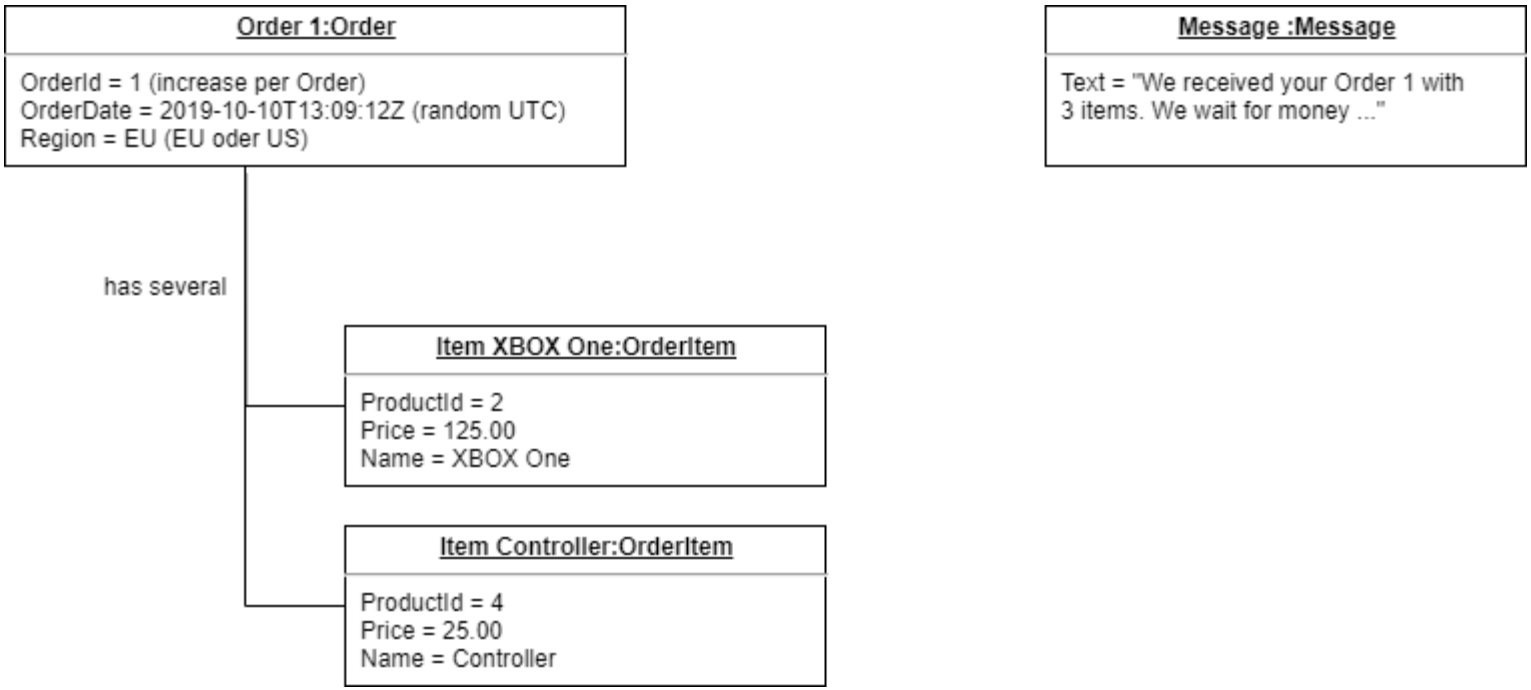
Übung 4

- Komponente `Infrastructure`
  - Erstellt Exchanges, Queues und Bindings
  - Erstellt alle X-Sekunden (randomisiert 2-5) eine `Order` und published sie in den `Order` Exchange (Beschreibung `Order` siehe folgende Folien)
  - Erzeugen Sie initial ein Array aus 10 Produkten und selektieren sie daraus randomisiert (inkl. Anzahl 1-3)
- Alle Komponenten werden als Console-Application .NET Core 3.1 in einer Solution umgesetzt (es sollten dann 6 Einträge in der Solution sein)
  - Einfach alle manuell starten
- Achten Sie auf genaue Namen von (sonst keine Wertung)
  - Komponenten
  - Exchanges
  - Queues


publishes to
«Annotation»
Infrastructure
creates
publish to
Billing
routing-key: US
routing-key: EU
consumes from
«Annotation»
Order
«Annotation»
Billing
consumes from
Billing.EU
Billing.US
publish to
publish to
Billing.Events
fanout
Order.BillingEvents
Order
direct
Orders
Messaging
fanout
«Annotation»
EmailNotification Service
consumes from
consumes from
«Annotation»
WhatsAppNotification Service
Email
WhatsApp

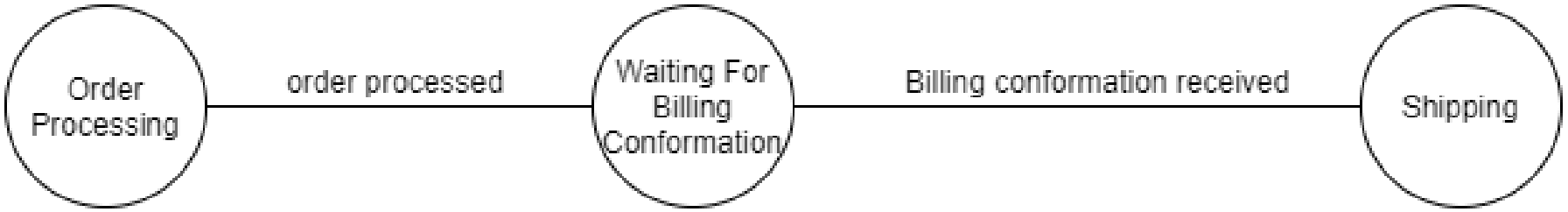
Komponente `Order`

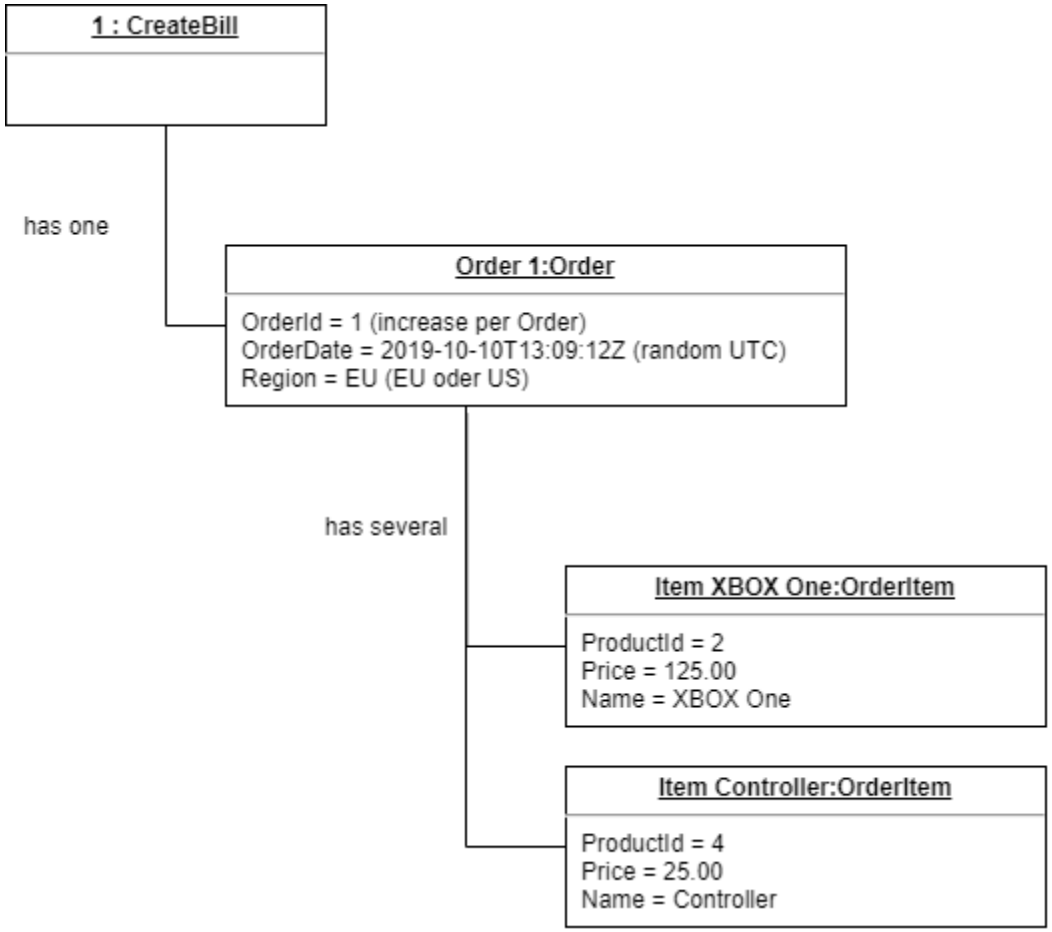
- liest aus der `Orders` Queue neue Orders und speichert sie In-Memory in einer Liste



Komponente `Order`

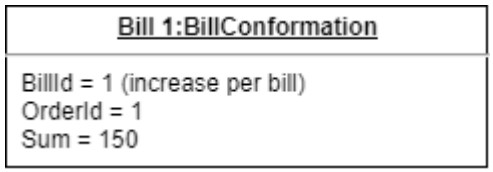
- Die Order hat folgende States (Optional: bauen einer State-Machine)
  - **OrderProcessing**:
    - in diesem Status wird ein Command `CreateBill` an den `Billing` Exchange geschickt. Ebenfalls Nachricht an Exchange `Messaging`, dass Bestellung erhalten wurde
    - Wechsel in den Status **WaitingForBillingConformation**
  - **WaitingForBillingConformation** : Es wird auf eine `BillConformation` (siehe nächste Folie) aus `Order.BillingEvents` gerwartet. Wenn empfangen: **Shipping**
  - **Shipping**: Nachricht an Exchange `Messaging`, dass Zahlung erhalten wurde und die Bestellung versandt wird




1 : CreateBill
has one
Order 1:Order
OrderId = 1 (increase per Order)
OrderDate = 2019-10-10T13:09:12Z (random UTC)
Region = EU (EU oder US)
has several
Item XBOX One:OrderItem
ProductId = 2
Price = 125.00
Name = XBOX One
Item Controller:OrderItem
ProductId = 4
Price = 25.00
Name = Controller

Komponente `Billing` „EU“ und „US“

- Beim Hochfahren der Applikation Auswahl „EU (1)“ oder „US (2)“ mit der Eingabe einer Zahl „1“ oder „2“
- Lesen sie aus `Billing.US` bzw. `Billing.EU` Queue
- Addieren der Item Preise
- Es wird eine Bill-Event `BillConformation` in den `Billing.Events` Exchange geschrieben
- Sinnvolle Ausgabe auf der Console (Nachvollziehbarkeit)



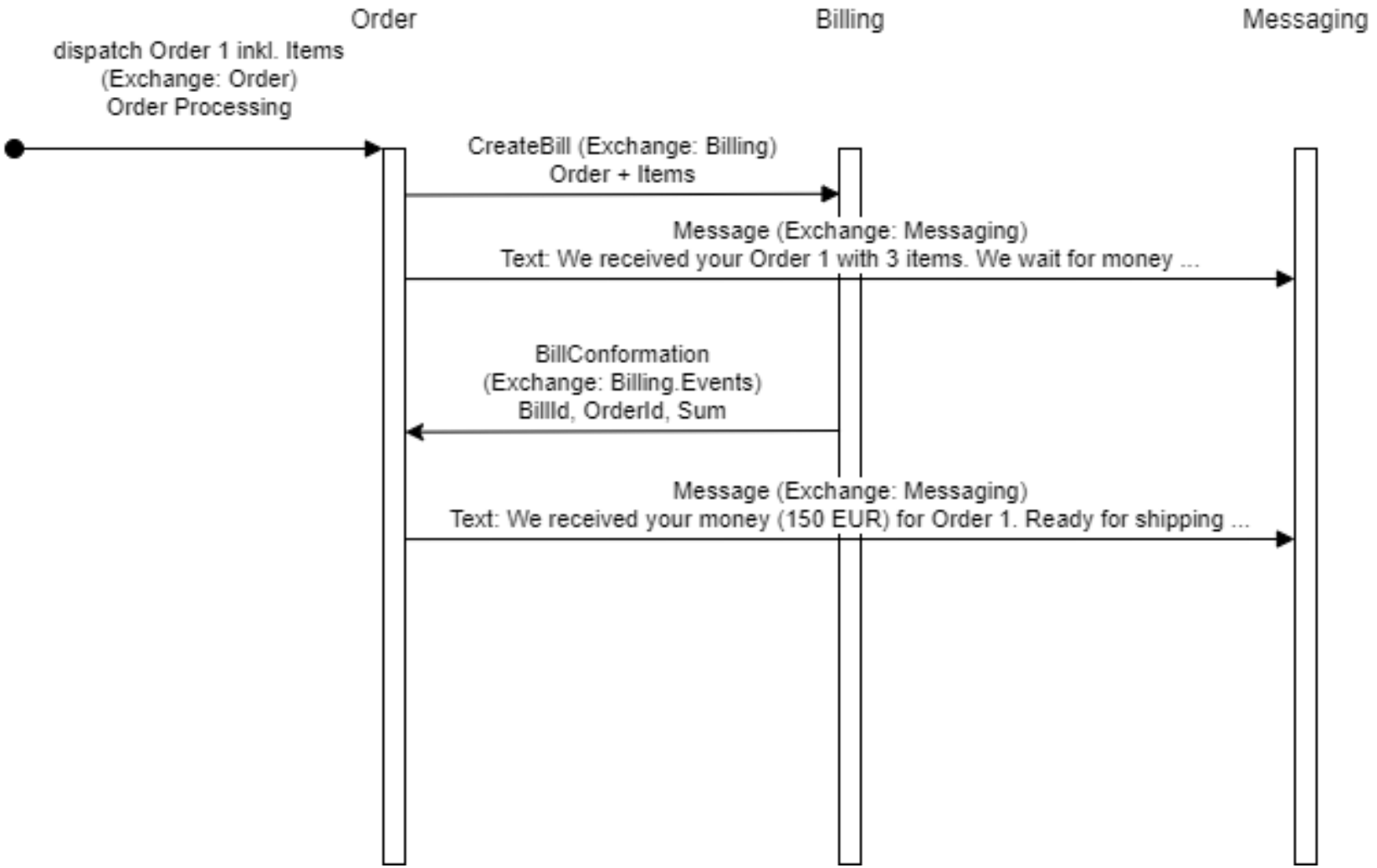
Komponente `WhatsAppNotificationService`

- Lesen aus der Queue `WhatsApp`
- Ausgabe auf der Console

Komponente `EmailNotificationService`

- Lesen aus der Queue `Email`
- Ausgabe auf der Console

Zusammenfassung



Anmerkung

- Wie Sie bemerkt haben → es wurden mehrere Nachrichten Typen verwendet: https://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageConstructionIntro.html
- Command, Event
- Ziel: Unterschiede aufzeigen – keine Empfehlung für die Praxis per se
- Anmerkung zu ProtoBuf:
  - Keine .proto File erforderlich
  - https://github.com/protobuf-net/protobuf-net

Fragen

- Erstellen sie im selben Ordner wie die *.SLN Datei eine „Antwort.txt“
- Fragen:
  1. Was passiert, wenn Sie e.g. Infrastructure und Order laufen lassen und die anderen Komponenten erst 60 Sekunden später starten? Gehen Bestellungen verloren?