



Event-Driven Architecture

Berlin, March 11, 2022

Zu mir 😊



Dr. Annegret Junker

- Lead Architect Allianz Technology SE
- habe in verschiedenen Rollen gearbeitet: Product Owner, Entwicklerin, Architektin

Veröffentlichungen

Integration Architecture, European Cloud Summit, München Oktober 2021

Einsatz von KI zur Unterstützung des Verkaufsprozesses in der Allianz-Gruppe, heise Akademie, München Oktober 2021

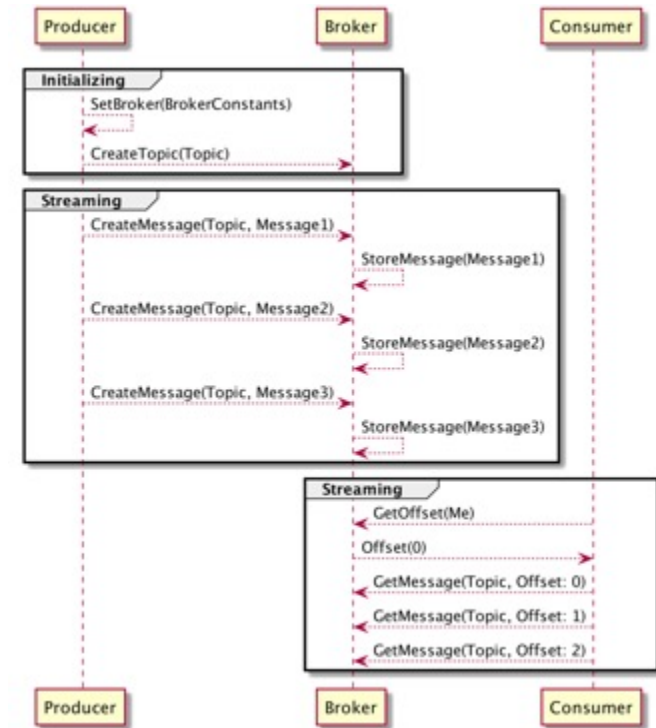
Workshop Domain Story Telling @ Allianz, Modern RE, Berlin Oktober 2021

Event-Driven Architecture

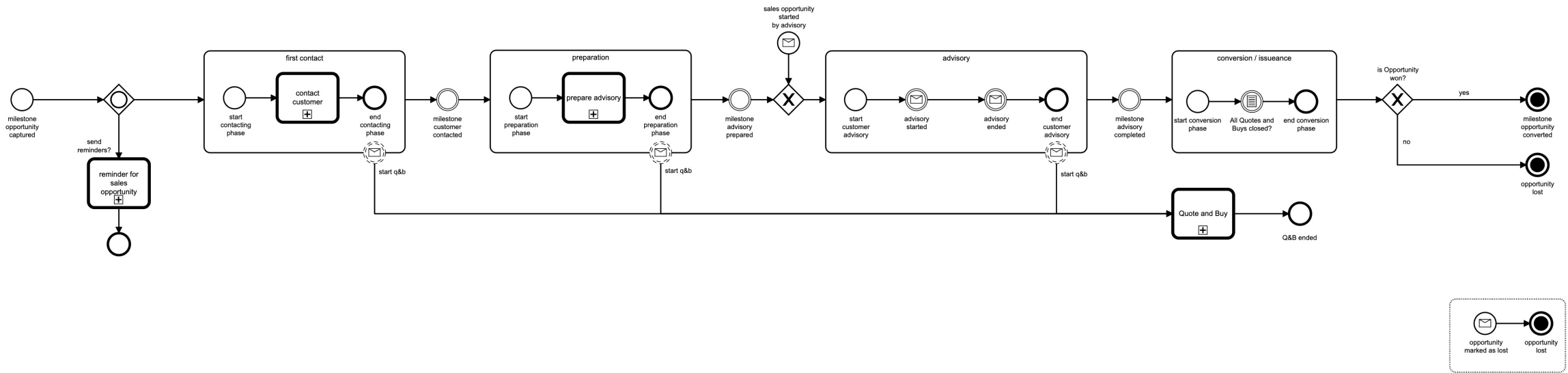
Zusammenspiel von Architektur-Modulen - (Micro-Services) wird über Ereignisse gesteuert.

Ereignisse (Events) sind Dinge, die passiert sind, z.B. ein Geschäftsobjekt wurde erzeugt. Da Events über entsprechende Broker verteilt werden, erhält man eine sehr flexible, entkoppelte Architektur.

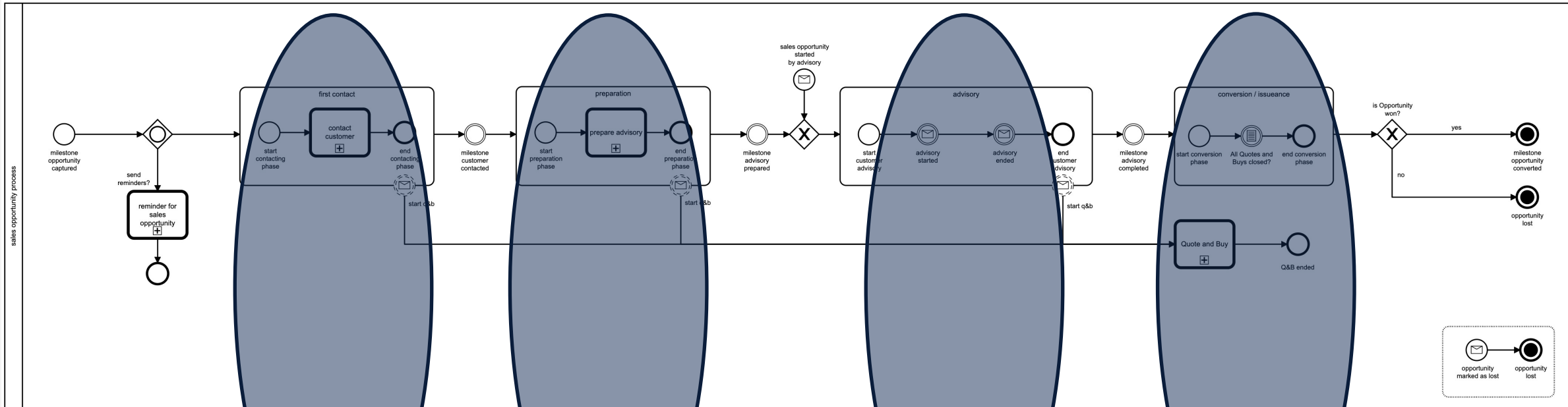
Kommandos (Commands) sind Dinge, die passieren sollen, z.B. ein Geschäftsobjekt soll erzeugt werden. Technologisch wird die gleiche Basis benutzt, aber Producer und Consumer des Commands sind enger gekoppelt. Daher gilt es diese Technik vorsichtig einzusetzen.



Die Aufgabe



Die Aufgabe



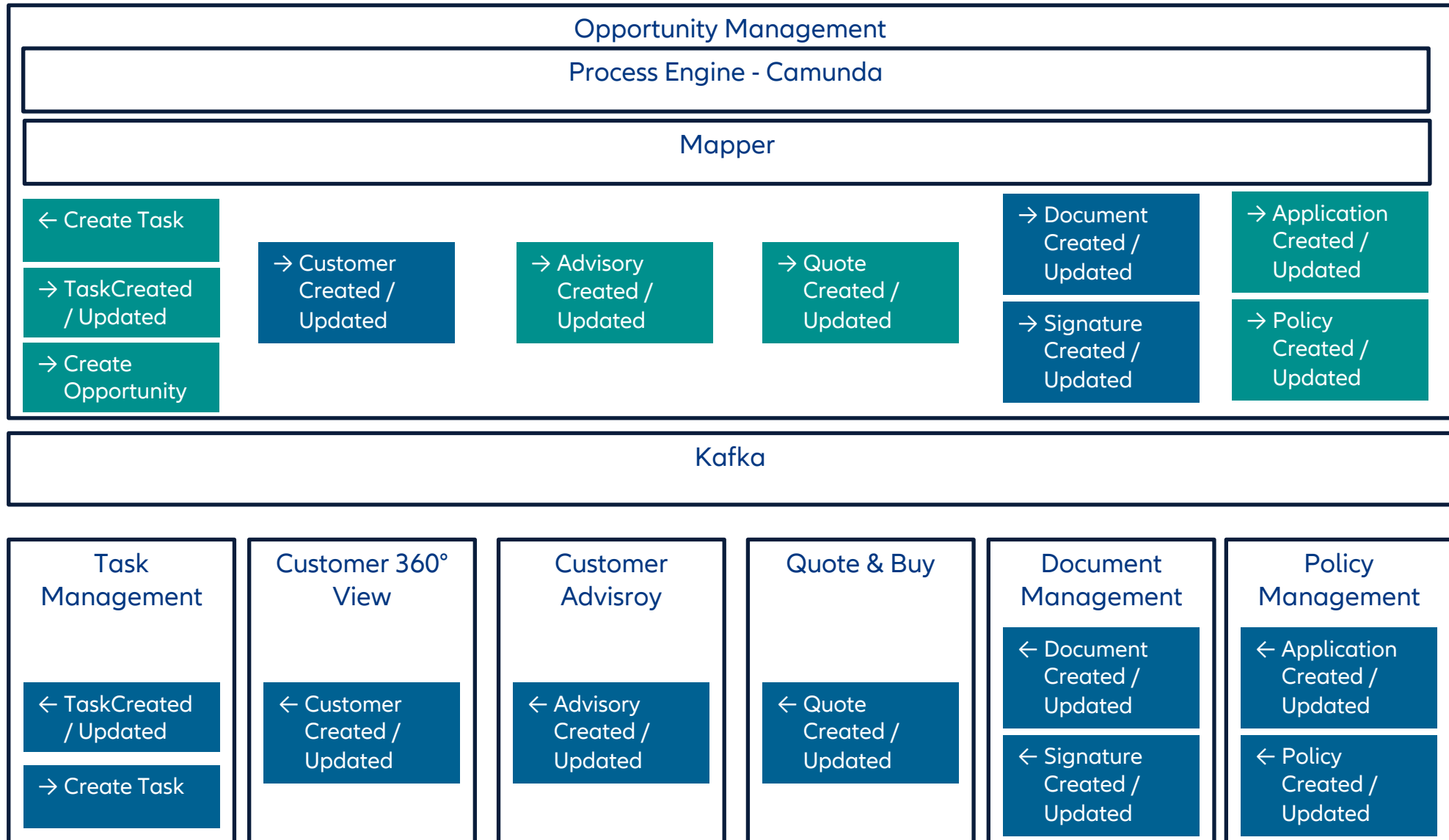
Appointment Management
Task Management

Customer 360° View
Task Management

Customer Advisory

Quote & Buy Journey
Document Management
Signature Service
Policy Management

EVENT-DRIVEN ARCHITECTURE @ ALLIANZ



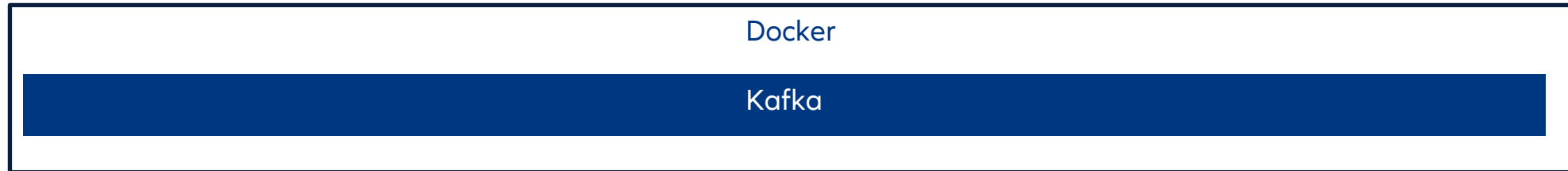
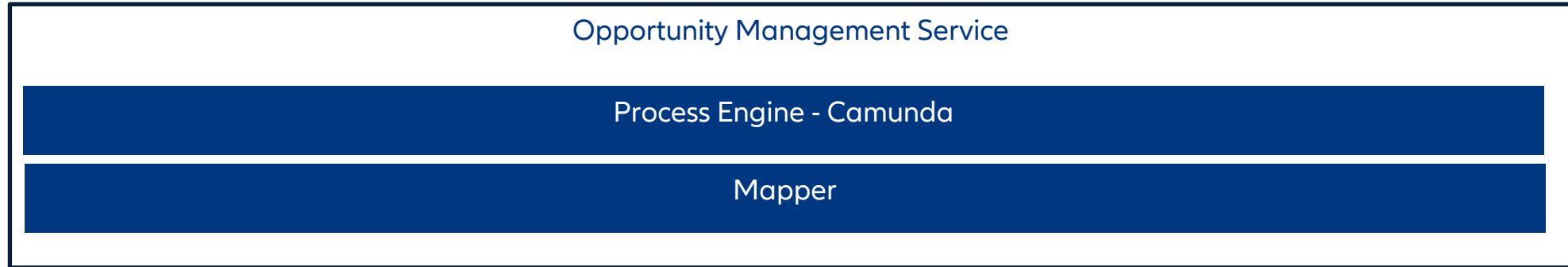
Events on Kafka as Topics

Hide internal topics

+ Add a topic

Topics	Availability			Throughput
Topic name	Under replicated partitions	Out of sync followers	Out of sync observers	Bytes/sec pro
<u>docker-connect-configs</u>	0 of 1	0 of 1	0 of 0	0
<u>docker-connect-offsets</u>	0 of 25	0 of 25	0 of 0	--
<u>docker-connect-status</u>	0 of 5	0 of 5	0 of 0	--
<u>local.sdw.customeradvisory.advisorymeeting.advisorycreated</u>	0 of 1	0 of 1	0 of 0	--
<u>local.sdw.customeradvisory.advisorymeeting.advisoryupdated</u>	0 of 1	0 of 1	0 of 0	--
<u>local.sdw.salesprocessmanagement.salesprocess.contractcreated</u>	0 of 1	0 of 1	0 of 0	--
<u>local.sdw.salesprocessmanagement.salesprocess.contractprocesstransitioncreated</u>	0 of 1	0 of 1	0 of 0	--
<u>local.sdw.salesprocessmanagement.salesprocess.contractremoved</u>	0 of 1	0 of 1	0 of 0	--
<u>local.sdw.salesprocessmanagement.salesprocess.createopportunity</u>	0 of 1	0 of 1	0 of 0	0
<u>local.sdw.salesprocessmanagement.salesprocess.opportunitycreated</u>	0 of 1	0 of 1	0 of 0	0
<u>local.sdw.salesprocessmanagement.salesprocess.opportunityupdated</u>	0 of 1	0 of 1	0 of 0	--
<u>local.sdw.salestaskmanagement.taskprocessing.createtask</u>	0 of 1	0 of 1	0 of 0	0
<u>local.sdw.salestaskmanagement.taskprocessing.taskcreated</u>	0 of 1	0 of 1	0 of 0	0
<u>local.sdw.salestaskmanagement.taskprocessing.taskupdated</u>	0 of 1	0 of 1	0 of 0	0

Local Sample



Take Away 01

Ein steuernder Prozess
führt zu starker
Kopplung

Stark steuernde Prozesse führen zu stark gekoppelten Systemen. Prozessindikatoren müssen in den jeweiligen Systemen bekannt sein und die Geschäftslogik liegt in den Abfragen und Verzweigungen des Prozesses. Unabhängige und verteilte Entwicklung ist kaum möglich.



Take Away 02

Prozesse können über Events von den produzierenden Systemen entkoppelt werden.

Event-getriebene Systeme schaffen entkoppelte Systeme. Die Geschäftslogik liegt in den entsprechenden Systemen. Event-produzierende Systeme und event-konsumierende Systeme können unabhängig voneinander entwickelt werden. Dieser Vorteil wird durch zusätzliche Komplexität in der Infrastruktur erkauft.

