Matrix-Organisationen, Conway's Law und Microservices

Was kann uns das sagen?

Daniel Lübke
daniel.luebke@digital-solution-architecture.com

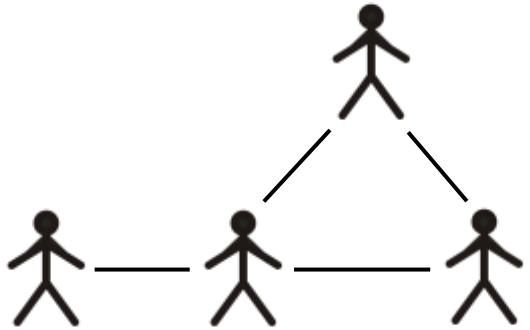
Über mich

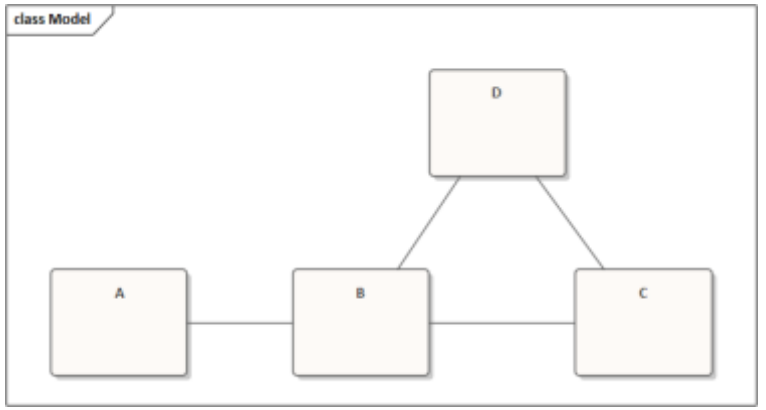
Dr.-Ing. Daniel Lübke

Consultant für Softwarearchitektur und Geschäftsprozessautomatisierung

“Organizations which design systems […] are constrained to produce designs which are copies of the communication structures of these organizations.”

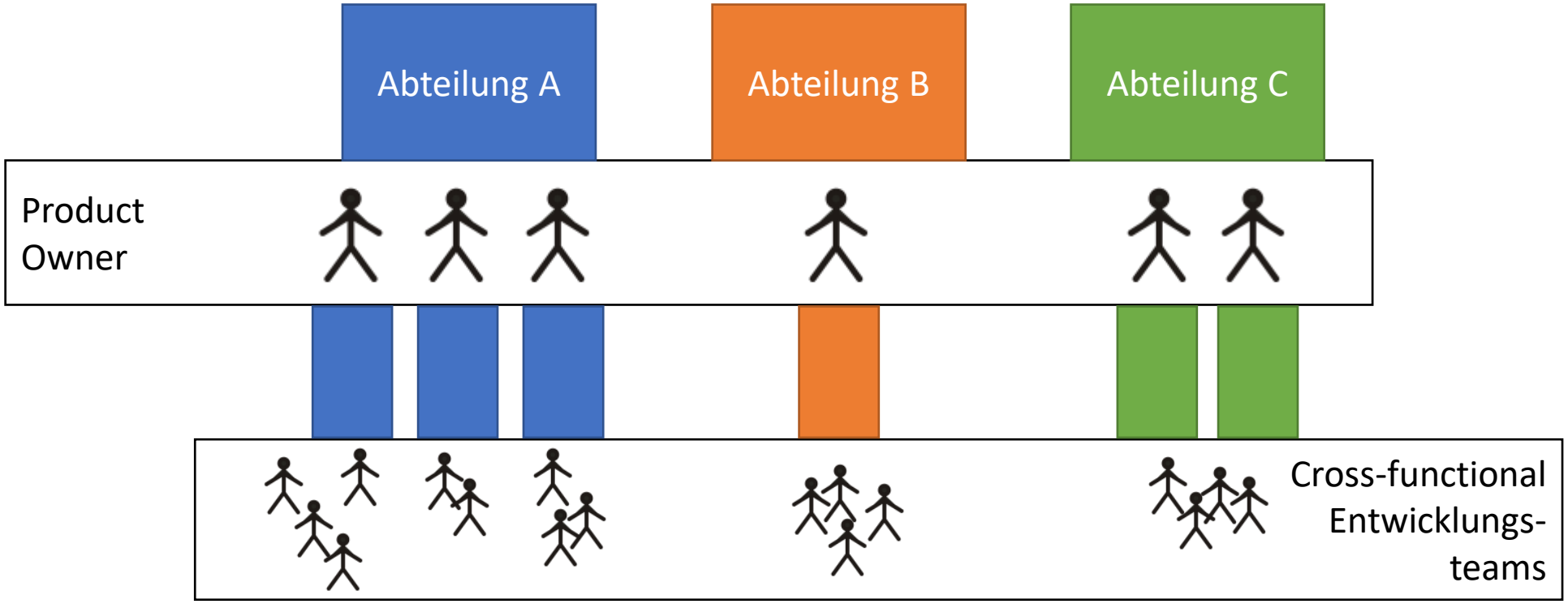
--- Melvin E. Conway





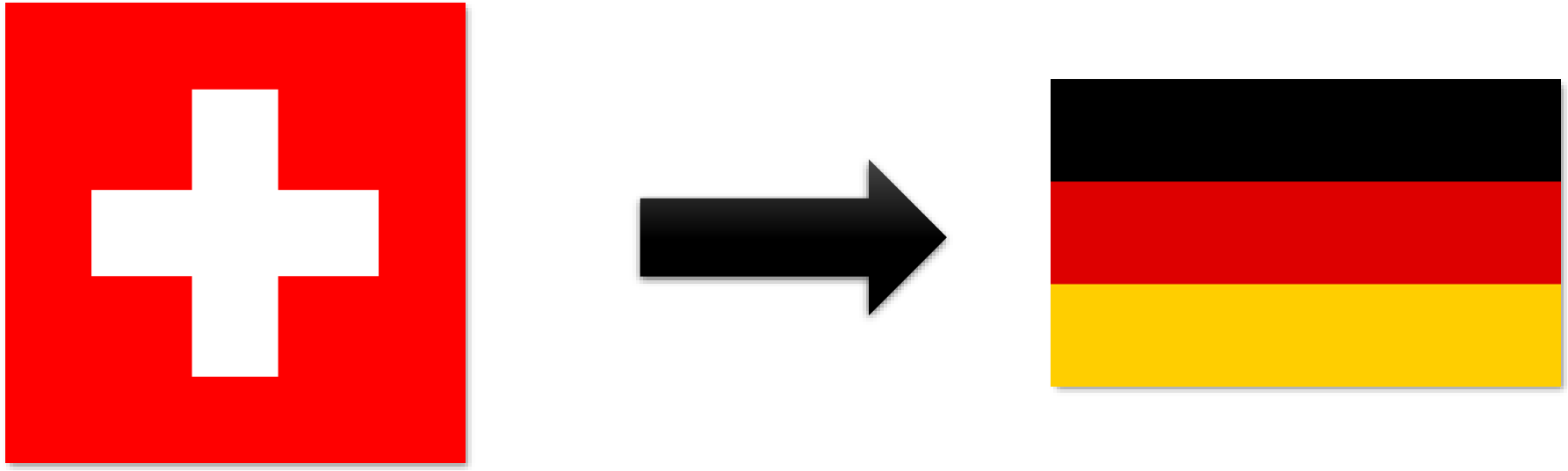
Funktionale Organisationsstruktur

Microservice Design / Bounded Contexts / Vertikalen



Problem 1

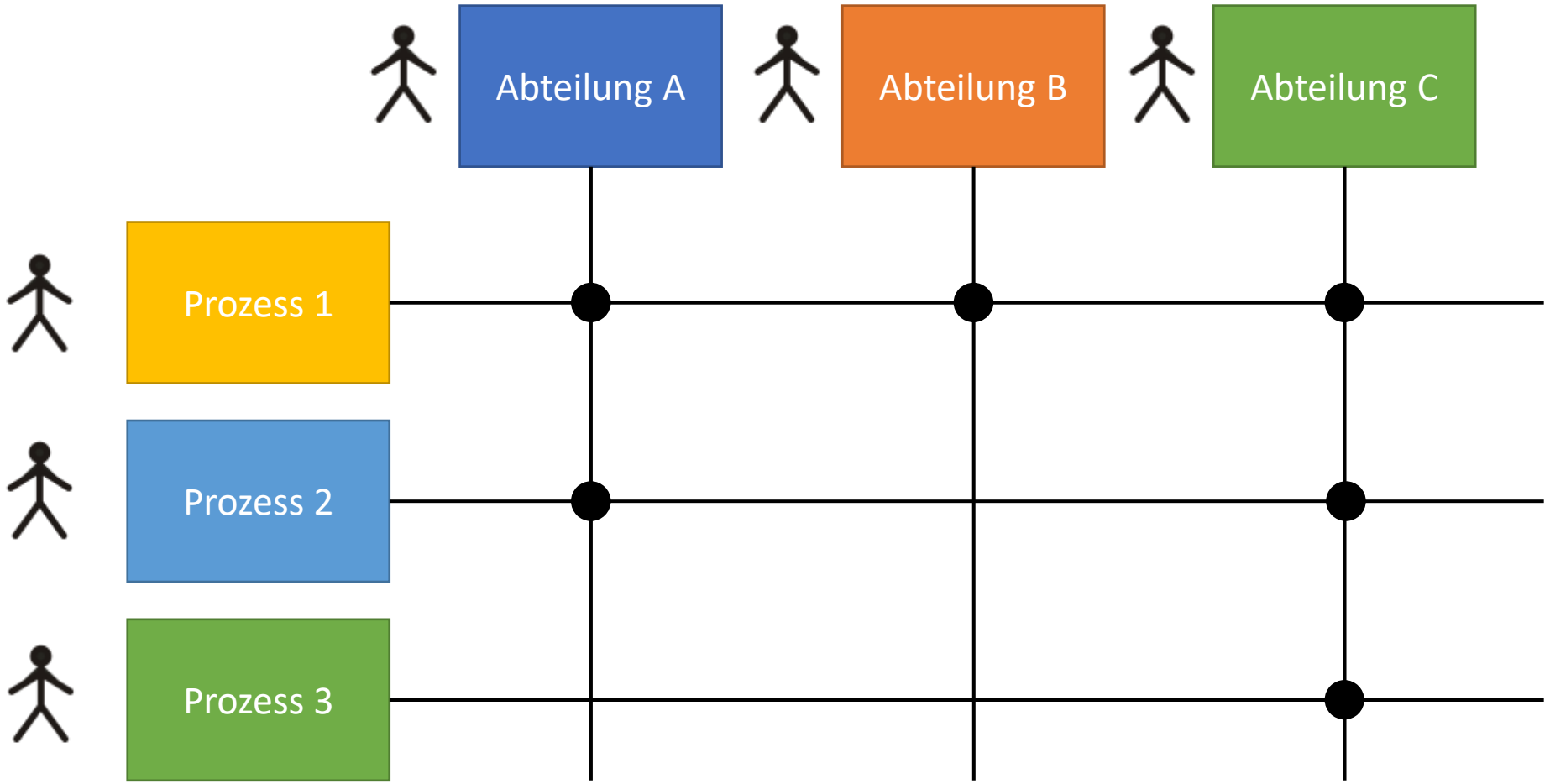
Problem 2



Problem 3

Schlechte Software-Lösung

Prozessorientierte Organisationen



Conway‘s Law rückwärts angewandt

Wir benötigen einen Platz für die Process Owner!!!

Wo implementieren wir Geschäftsprozesse in der Software?

Verschiedene Arten von Prozessen

- Laufzeit
  - Kurzlebig („Use Case“)
  - Langlebig (mehrere Jahre)
- Beteiligte
  - Eine Person
  - Eine Abteilung
  - Mehrere Abteilungen
- Organisationen
  - Eine Organisation
  - Organisationsübergreifend

Wo implementieren wir Geschäftsprozesse in der Software?

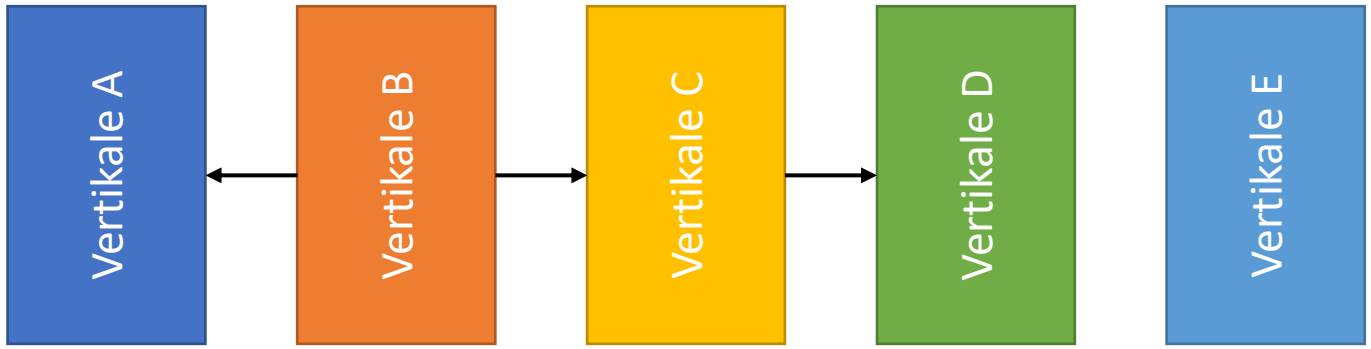
Option 1


Vertikale A
Vertikale B
Vertikale C
Vertikale D
Vertikale E

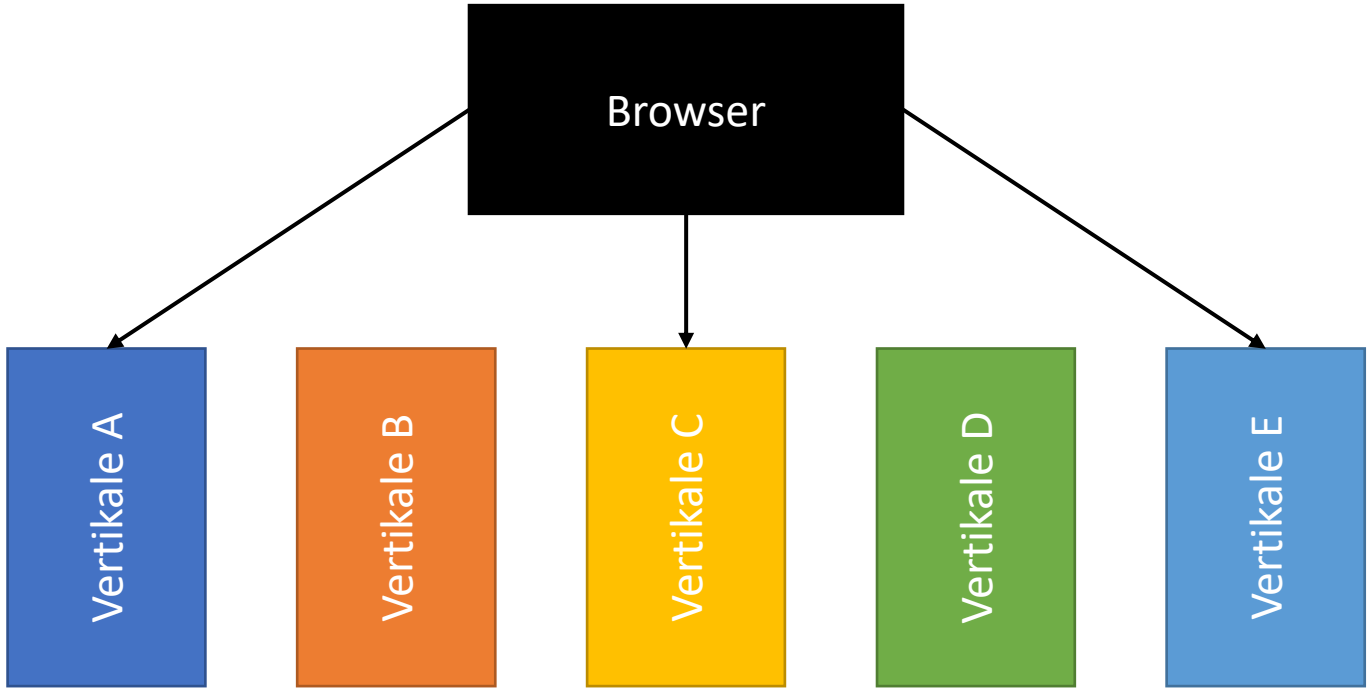
„The microservice community favours an alternative approach: *smart endpoints and dumb pipes*. Applications built from microservices aim to be as decoupled and as cohesive as possible - they own their own domain logic and act more as filters in the classical Unix sense - receiving a request, applying logic as appropriate and producing a response. **These are choreographed using simple RESTish protocols rather than complex protocols such as WS-Choreography or BPEL or orchestration by a central tool.**”

https://www.martinfowler.com/articles/microservices.html

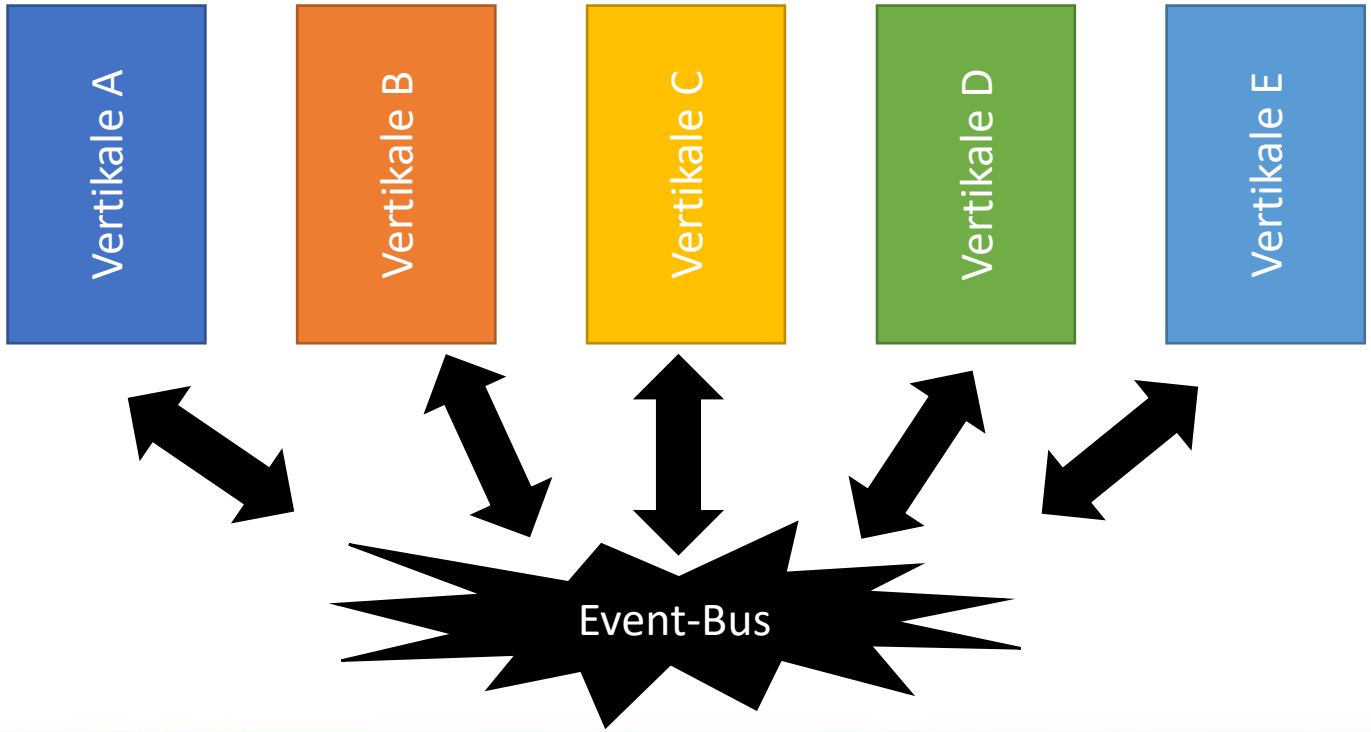
Option 2a


Vertikale A
Vertikale B
Vertikale C
Vertikale D
Vertikale E

Option 2b


Browser
Vertikale A
Vertikale B
Vertikale C
Vertikale D
Vertikale E

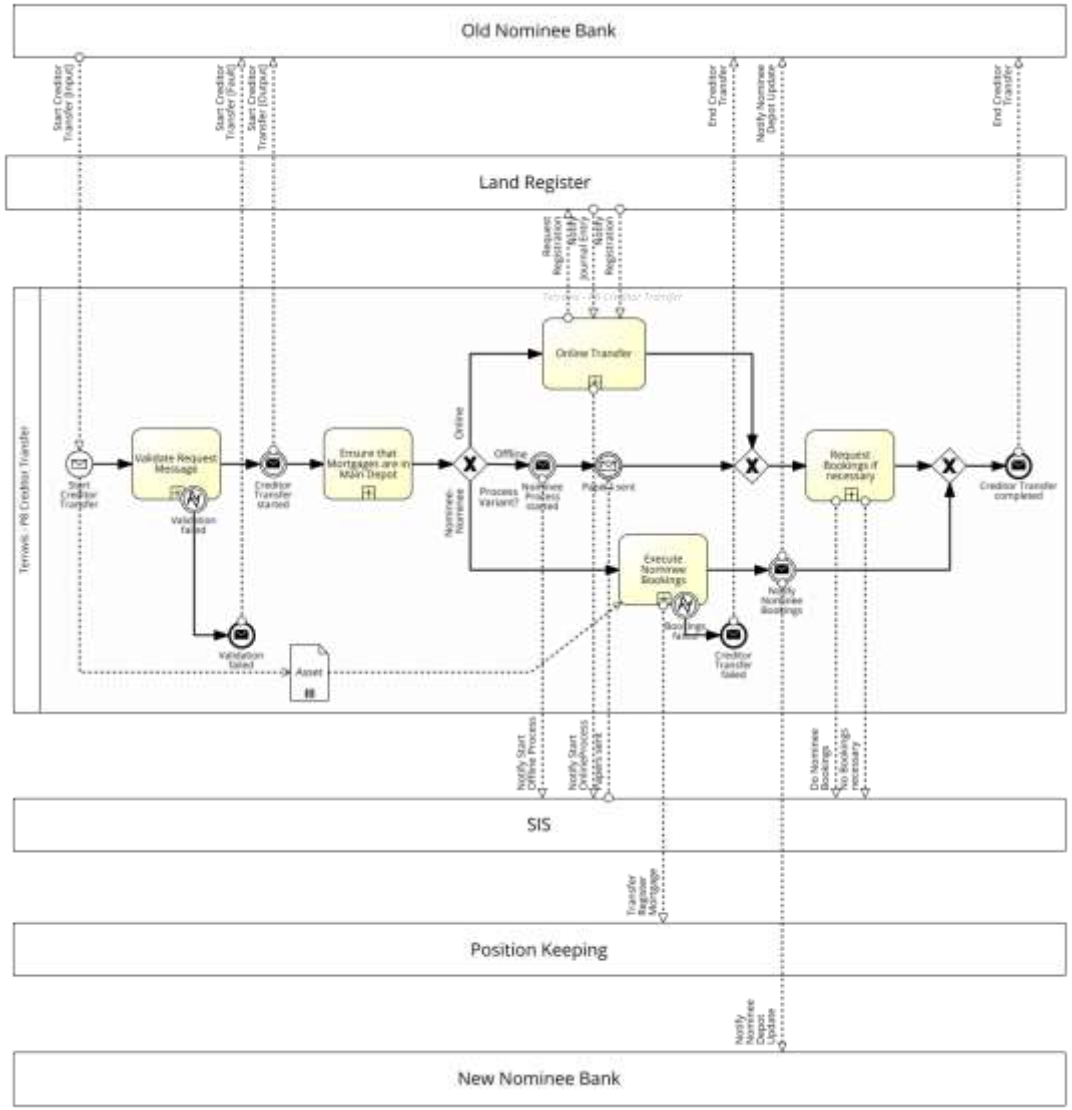
Option 3


Vertikale A
Vertikale B
Vertikale C
Vertikale D
Vertikale E
Event-Bus

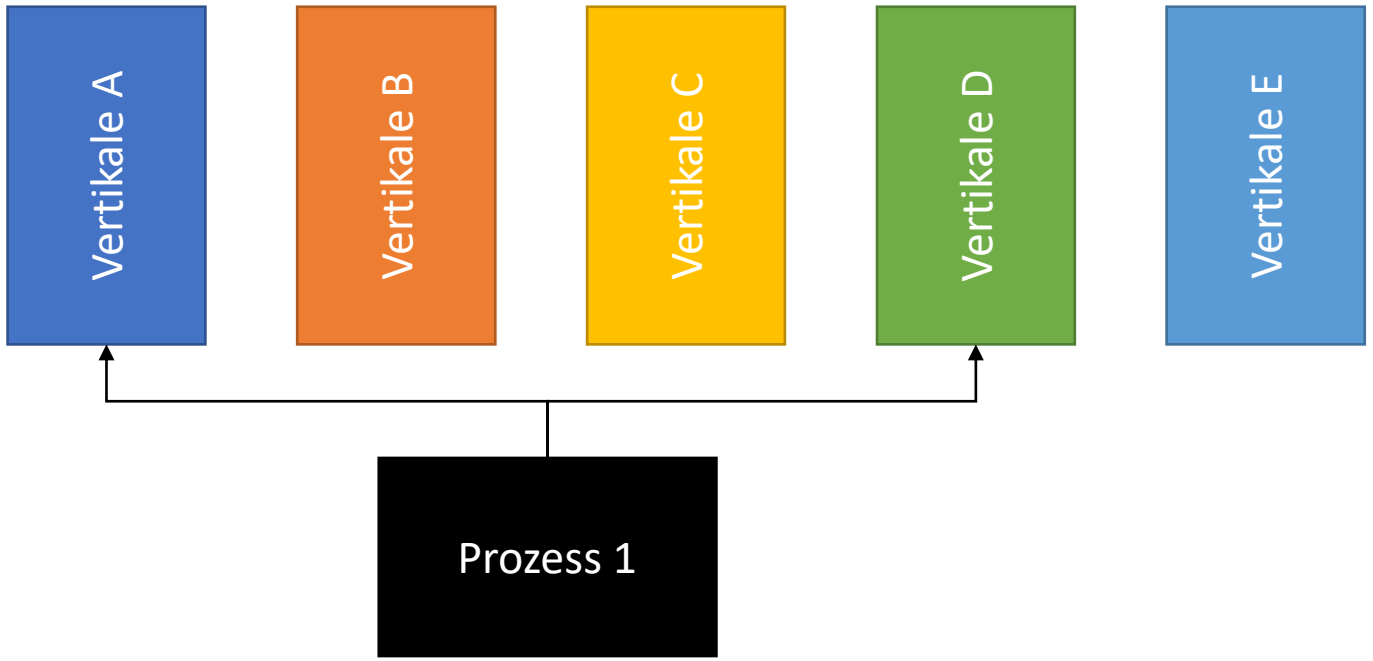
„Event notification is nice because it implies a low level of coupling, and is pretty simple to set up**. It can become pro run** can pro fro**g and** ma without realizing that you're losing sight of that larger-scale flow, and thus set yourself up for trouble in future years.”

“These are **choreographed** using simple RESTish protocols rather than complex protocols such as WS-Choreography or BPEL or orchestration by a central tool.”

https://martinfowler.com/articles/201701-event-driven.html


Old Nominee Bank
Land Register
Terrows - P8 Creditor Transfer
Validate Request Message
Validation failed
Creditor Transfer started
Ensure that Mortgages are in Main Depot
Online
Offline
Online Transfer
Process Variant?
Nominee-Nominee
Nominee Process started
Payment sent
Execute Nominee Bookings
Booking failed
Notify Nominee Bookings
Request Bookings if necessary
Creditor Transfer completed
Validation failed
Creditor Transfer failed
Asset
Notify Start Offline Process
Notify Start OnlineProcess
Payment sent
Do Nominee Bookings
No Bookings necessary
SIS
Transfer Register Mortgage
Position Keeping
New Nominee Bank

Option 4

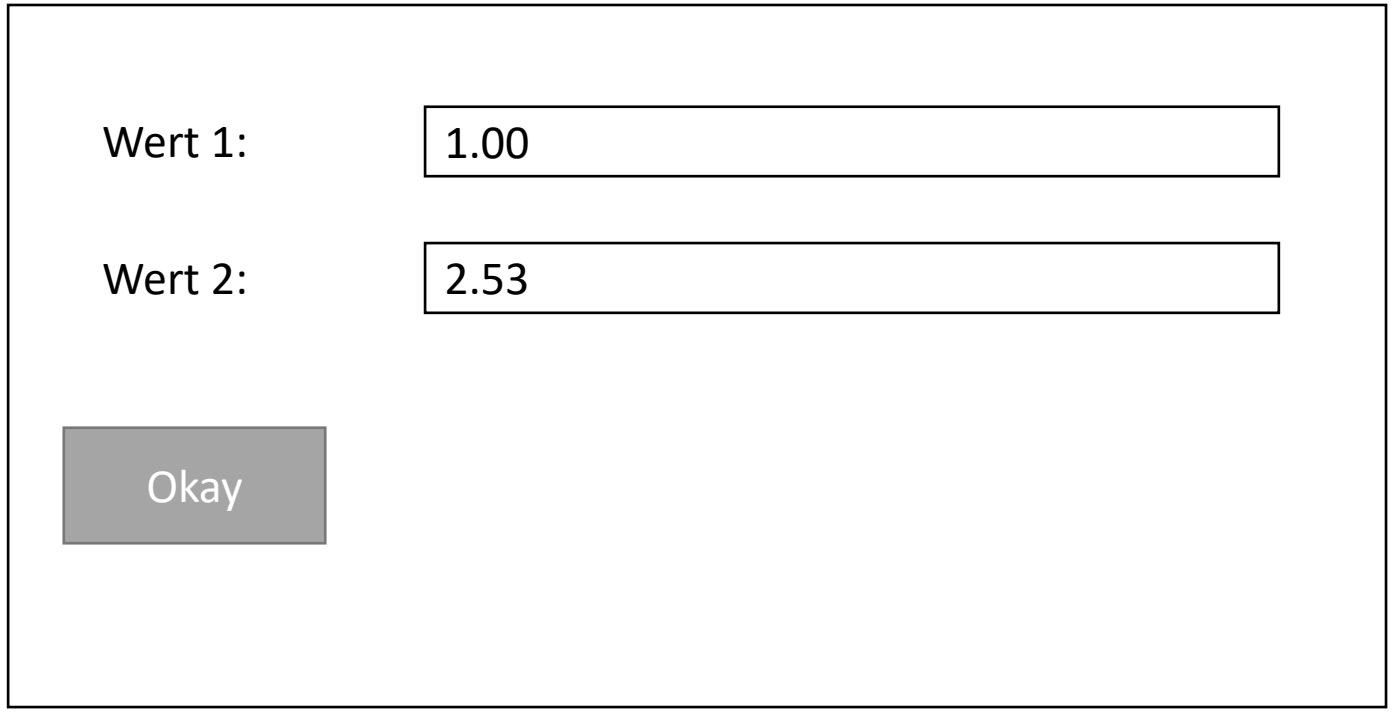

Vertikale A
Vertikale B
Vertikale C
Vertikale D
Vertikale E
Prozess 1

Vorteile von Prozess-Microservices

- Klare Architektur-Zuständigkeit für
  - den Prozessfluss
  - prozessspezifische Artefakte
- Klare Zuständigkeit in der Organisation (Process Owner)
- Sind keine (reinen) Orchestrierungskomponenten
  - Enthalten eigene Logik und die Prozesslogik
  - Können/dürfen/sollen andere Services/APIs aufrufen (entkoppelt)
  - Events können konsumiert und geworfen werden
  - Kann selber eine Vertikale sein

Prozessspezifische Artefakte

- UIs
- Reports
- Generierte Dokumente
- Einfache Verfügbarkeit von Kennzahlen zur Prozessausführung


Wert 1:
1.00
Wert 2:
2.53
Okay

„Polyglot“-Entwicklung

- Unterschiedliche Technologien können bei Bedarf und zielgerichtet zum Einsatz gebracht werden

- Plain Old Programming Languages
- Persistent Actors
- BPMN Engines
- Task-Management
- Datenanalyse-Tools und -Frameworks
- Audit-Trails

Verbesserte Testbarkeit

- Prozess-Abhängigkeiten auf andere Vertikalen können gemockt werden
- End-to-End Prozesslogik kann isoliert getestet werden

Zusammenfassung

- Beachten Sie die Prozessdimension in einer Organisation
  - Bessere Software (Usability, Architektur)
  - Beachtung von Conway‘s Law
- Entscheiden Sie sich bewusst, wo Sie Geschäftsprozesse in der Software implementieren
  - Unterschiedliche Prozesscharakteristiken
- Dedizierte Prozess-Microservices sind eine valide Architekturentscheidung!

Vielen Dank für Ihre Aufmerksamkeit!

Daniel Lübke <daniel.luebke@digital-solution-architecture.com>