MicroServices in .NET

Disclaimer

Der Vortrag beruht auf den persönlichen Erfahrungen und Kenntnissen des Autors und erhebt keinerlei Anspruch auf Korrektheit und Vollständigkeit.

Vorstellung

- Seit 2006 im .NET Umfeld tätig
- Seit 2011 fokussiert auf .NET Client-Server Anwendungen mit WCF
- Zwischendurch immer mal wieder Web-Anwendungen
- Seit 2014 verstärkt im Bereich Software Architektur unterwegs

HSBC INKA

Aareon

Agenda

- Was sind Microservices?
- Warum Microservices verwenden?
- Ein Microservice im Überblick
- Beispiel in .NET mit ASP.NET WebApi 2

Was sind Microservices?

“Definition”:

“Microservices sind ein Architekturmuster der Informationstechnik, bei dem komplexe Anwendungssoftware aus **kleinen**, **unabhängigen** Prozessen **komponiert** wird, die untereinander mit **sprachunabhängig**en Programmierschnittstellen **kommunizieren**. Die Dienste sind **klein**, weitgehend entkoppelt und erledigen **eine** kleine Aufgabe. So ermöglichen sie einen modularen Aufbau von Anwendungssoftware.”

https://de.wikipedia.org/wiki/Microservices

Was sind Microservices?

Eigenschaften eines Microservices:

- Verantwortlich für eine einzige “Funktion” (capability / Fähigkeit)
- Kann unabhängig von anderen Microservices veröffentlicht werden
- Besteht aus einem oder mehreren Prozessen
- Besitzt seine eigene Datenhaltung
- Ist leicht / schnell ersetzbar
- Ein kleines Team kann eine handvoll Microservices pflegen

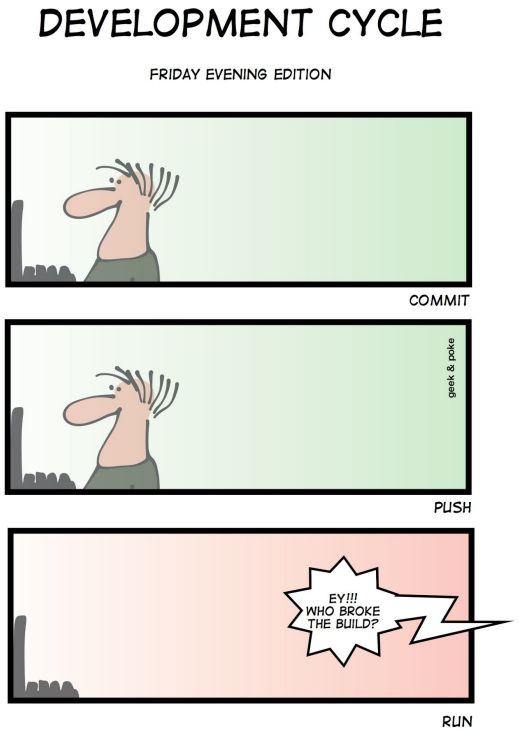
Eine “Funktion”/Fähigkeit (capability)

- Basiert auf dem SRP (Uncle Bob)
- Microservice ändert sich nur, wenn sich die Fähigkeit ändert
- Unterschiedliche Fähigkeiten
  - Fachlich > Funktion
  - Technisch > Integration eines externen Systems

@Instagram

Individuelles Deployment

- Muss ohne Anpassung anderer Services veröffentlicht werden können
- Deployments besser handhabbar
- Geringe Downtime
- Schnittstellen müssen abwärtskompatibel sein



Eigene Prozesse

- Unabhängigkeit von anderen Microservices
- Vorbedingung für individuelles Deployment
- Warum mehrer Prozesse?
  - Beispiel: Microservice und Datenbank

Eigene Datenhaltung

- Verantwortung der Daten liegt im entsprechenden Microservice
- Einsatz verschiedene Technologien möglich
  - Beste Lösung für ein Problem
- Nachteile:
  - Administration, Entwicklung und Wartung der unterschiedlichen Systeme

Kleines Team - Mehrere Microservices

- Größe des Microservices nicht definiert
- 4-5 Leute sollten eine handvoll Microservices handhaben können:
  - Weiterentwicklung
  - Wartung
  - Bugfixing
  - Monitoring
  - Testing
  - etc.

Leicht / Schnell ersetzbar

- Anforderungen ändern sich
- Schlechte Performance im Betrieb
- Neuere, bessere Technologien

Warum Microservices verwenden?

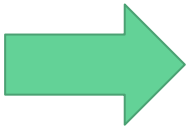
Warum Microservices verwenden?

- Wird von den “Global Playern” eingesetzt (Amazon, Netflix, Otto-Versand, AutoScout24)
- Können schnell Legacy-Systeme mit Funktion erweitern
- Effizienteres Arbeiten im Entwicklungs-/Wartungsprozess
- Robuste Architektur
- Skalierung
- Continuous Delivery wird vereinfacht
- “State-of-the-Art-Implementation” / Beste Lösung für das Problem

Warum Microservices (nicht) verwenden?

Bitte beachtet jedoch:

- Verteilte Architektur => höhere Komplexität (Netzwerk, Last, Fehler)
- Komplexere Testszenarien
- Abhängigkeiten zwischen Funktionen weiterhin vorhanden
- Beeinflusst die gesamte Unternehmenskultur/-organisation



Nur einsetzen wenn es sich lohnt!

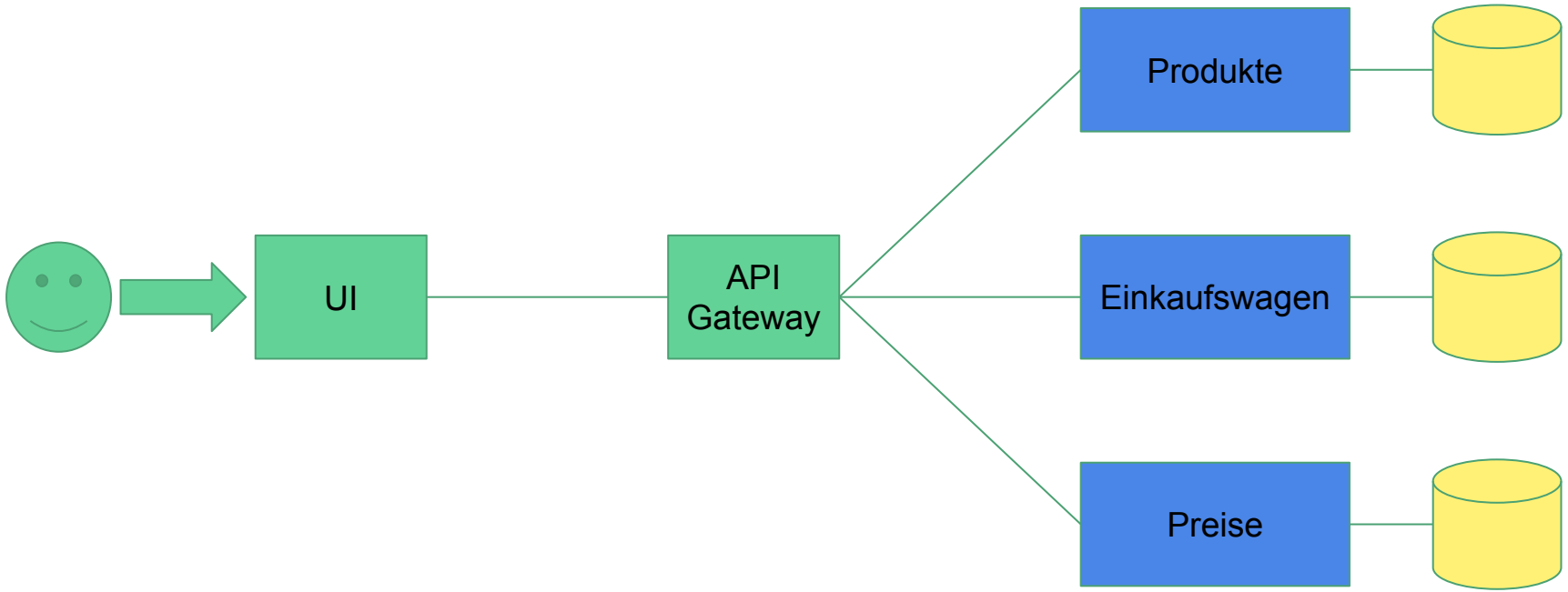
Ein Microservice im Überblick

Der Einkaufswagen

Ein “einfacher” Einkaufswagen in einem eShop soll implementiert werden.

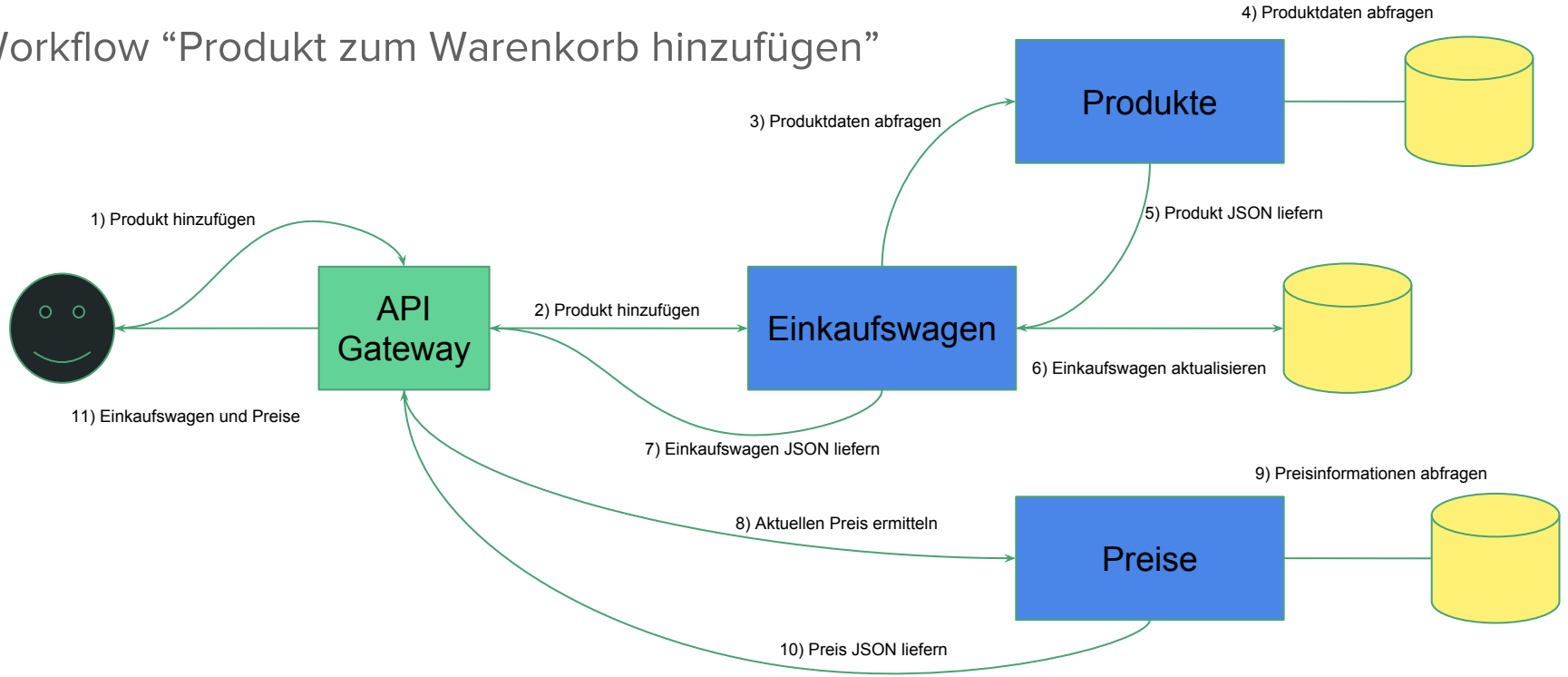
- Abhängigkeiten:
  - Produktinformationen
  - Preise inkl. von Rabatten, Angeboten, o.a. Aktionen
- Aufgaben:
  - Speichern des Warenkorb eines Benutzers
  - Aktualisieren der Preise innerhalb des Warenkorbes

Der Einkaufswagen



Der Einkaufswagen - Ablauf

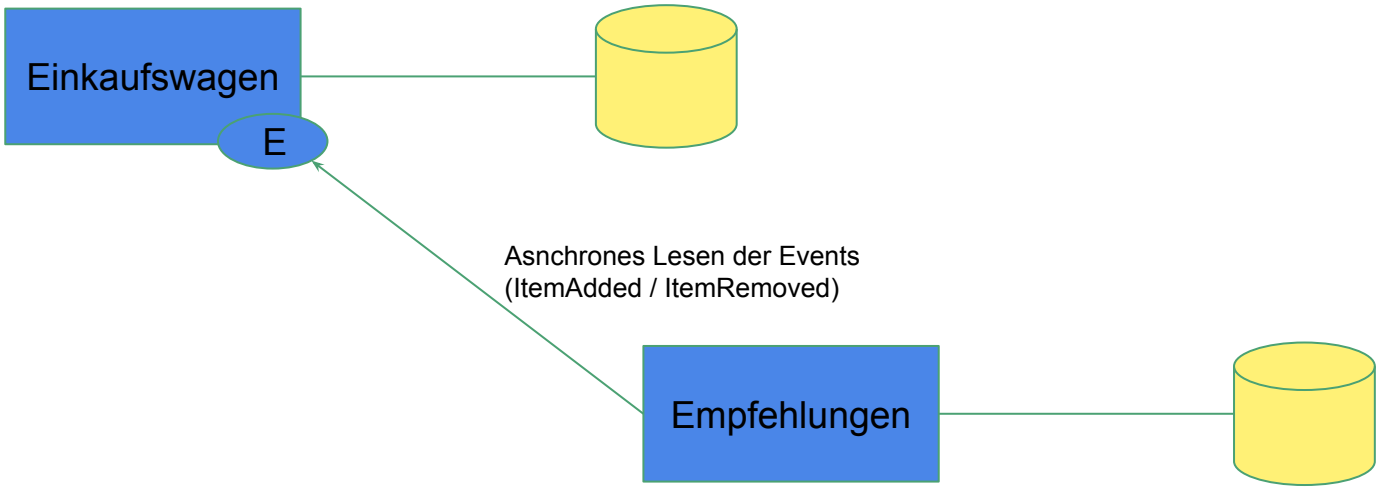
Workflow “Produkt zum Warenkorb hinzufügen”



Der Einkaufswagen - Kommunikation

- Verteilte Microservices interagieren miteinander
- Synchrone Kommunikation
  - Commands - HTTP POST / PUT
  - Queries - HTTP GET
- Asynchrone Kommunikation
  - Events - Polling des Subscribers HTTP GET

Der Einkaufswagen - Events Beispiel



Und im Code?

Weitere Lektüre

- Microservices in .NET - Horsdal - Manning (ISBN 9781617293375)
- Building Microservices - Newman - O'Reilly (ISBN 9781491950357)
- http://martinfowler.com/articles/microservices.html
- http://www.informatik-aktuell.de/entwicklung/methoden/microservices.html

Folien und Source

https://github.com/m4cx/microservices.sample

Noch Fragen?