

BRYTER

No-code Service Automation

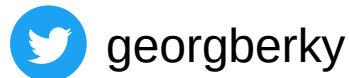
Branch by Abstraction

Georg Berky



über mich

- **Entwickler + DevOps Enthusiast**
 - Java, Groovy, Kotlin, Clojure
 - Linux, Ansible, Liquibase, Maven
- **Co-Organisator Software Crafting Meetup:**
 - Ruhrgebiet
 - Coderetreats, Coding Dojos



georgberky

BRYTER berky@bryter.io



mein Problem

- **mal eben die ganze Persistenz tauschen**
 - alt: JBoss Distributed Cache
 - neu: Hibernate
- **aber:**
 - ohne Downtime
 - so etwa 100k Requests/sec.
 - andere Teams arbeiten auch am Modul

Coding Time!

so (in etwa) hat unser Code ausgesehen

Branches?

- **Integrative Distanz**
 - passt Branch noch zu Main?
 - passt Branch_i noch zu Branch_{i+1}
- **Branches skalieren nicht:**
 - n Branches \Rightarrow n! Integrationen
 - merge hell is real



Branch by Abstraction

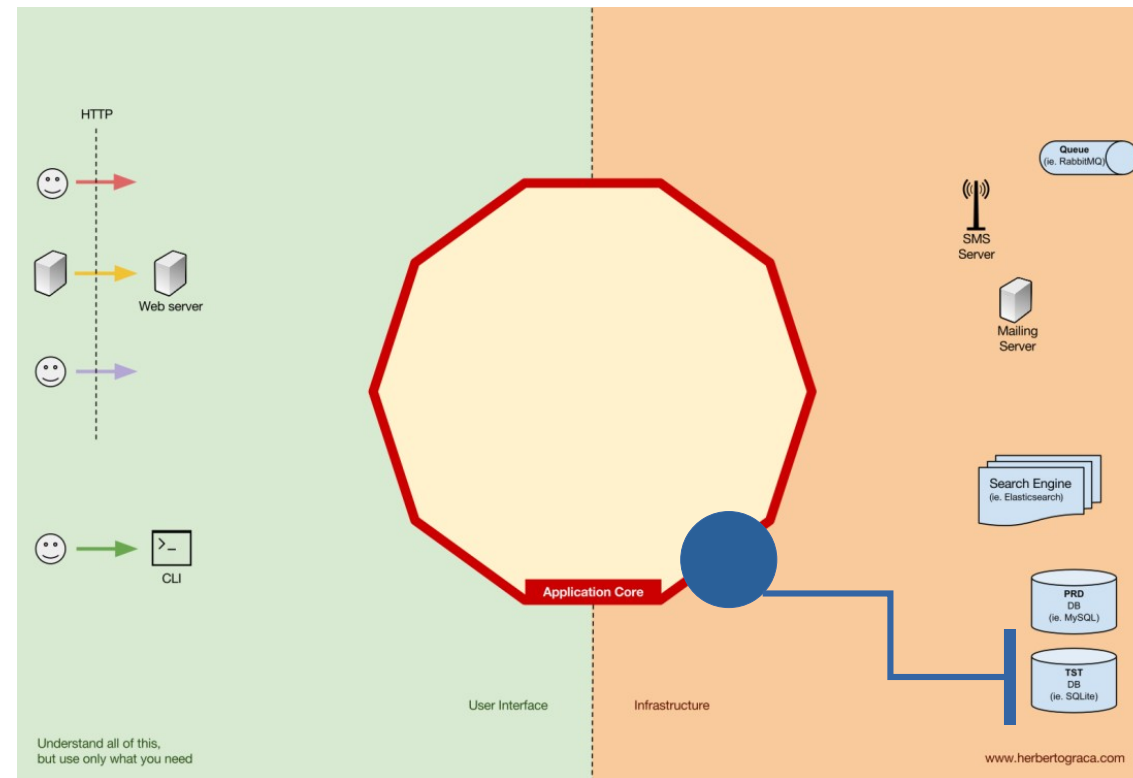
- **OOP statt Branching**
 - Abstraktion einführen \Rightarrow "DataStore"
 - inklusive Datenklassen
 - Store liefert Business Objects
 - Datenklassen/Entities \Rightarrow intern
 - Abstraktionskonformität herstellen
 - Tests + Produktivcode!
 - Tests: Store-Logik vs. spezifische Logik

Branch by Abstraction

- **OOP statt Branching**
 - Feature Toggle einführen
 - neue Implementierung daneben legen
 - alle Implementierungen durch CI absichern
 - Dark Deployment + Staging
 - Cleanup

Zielarchitektur

- **hexagonale Architektur**



Branch by Abstraction

- **mit anderen Worten**
 - Open-Closed Principle
 - zuerst: close, dann: open
 - Liskov Substitution Principle
 - Stores austauschbar
 - Single Responsibility Principle
 - keine DB in der Business-Schicht

Coding Time!

Danke!



<https://georg.berky.dev>

 georgberky

BRYTER berky@bryter.io