



JOHANNES SEITZ – 14.11.2019

Mehr Flow: essenzielle Techniken für Continuous Delivery

INNOQ



JOHANNES SEITZ – 14.11.2019

Mehr Flow: essenzielle Techniken für Continuous Delivery

INNOQ

**“Qualität muss
man von Anfang
an einbauen.“**



Johannes Seitz

**Senior Consultant
bei innoQ Deutschland GmbH**

Johannes Seitz arbeitet seit über 10 Jahren in der Softwarebranche, zunächst als Entwickler, später als Architekt und Coach. Als Entwickler, Berater, Coach und Trainer hilft er Teams dabei Software nachhaltig zu bauen oder sie zu sanieren.

Was ich euch erzählen möchte

- Mehr "Flow" durch Continuous Integration
- Wie weit man mit C.I. gehen kann
 - Große Refactorings
 - Unvollständige Features
 - Datenbankschemaänderungen
 - Library-Änderungen
 - Service-Änderungen

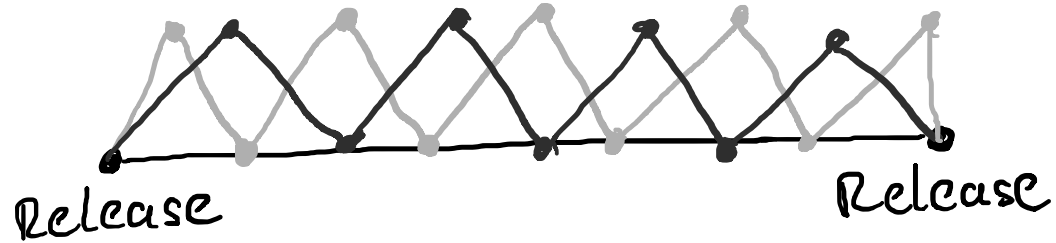


Continuous =
Ohne Unterbrechung

Jenkins reicht nicht!

Wie Teams zusammenarbeiten

Continuous
Integration



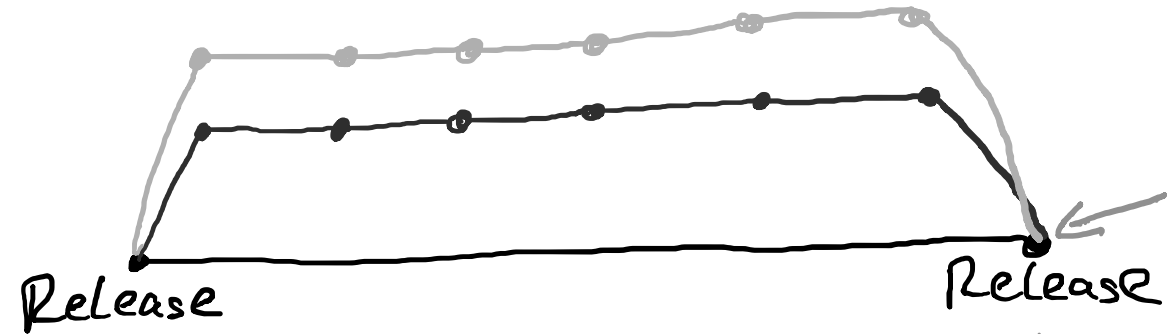
„Big Bang“
Integration



• • • Commits
— — — Branches

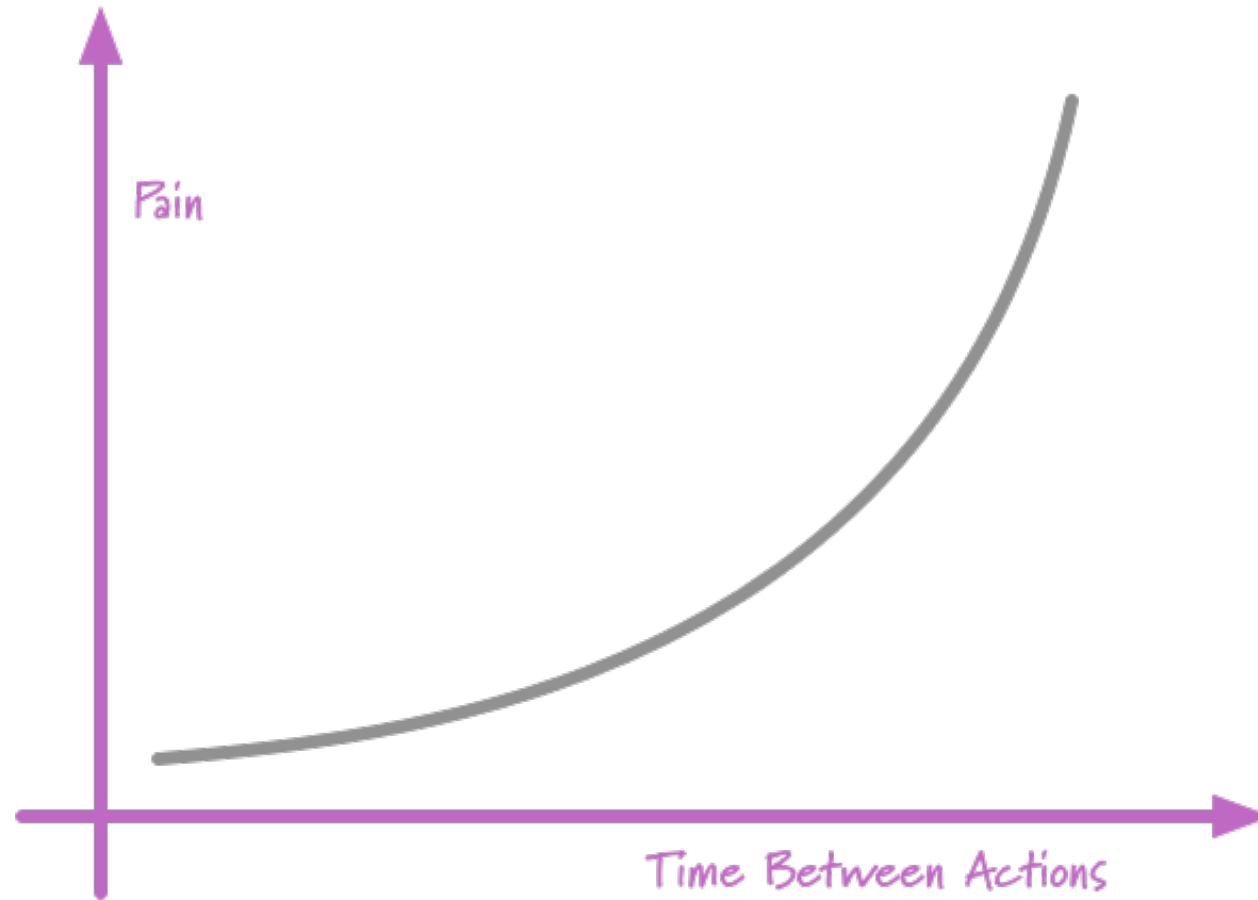
„Big Bang“ Integration

Z.B. Git Flow...



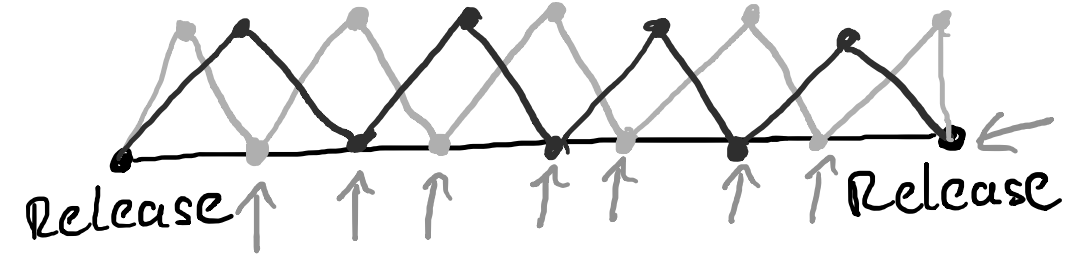
- ✓ Feature cherry-picking
- ✓ Isoliertes Arbeiten

„If it hurts, do it more often!“



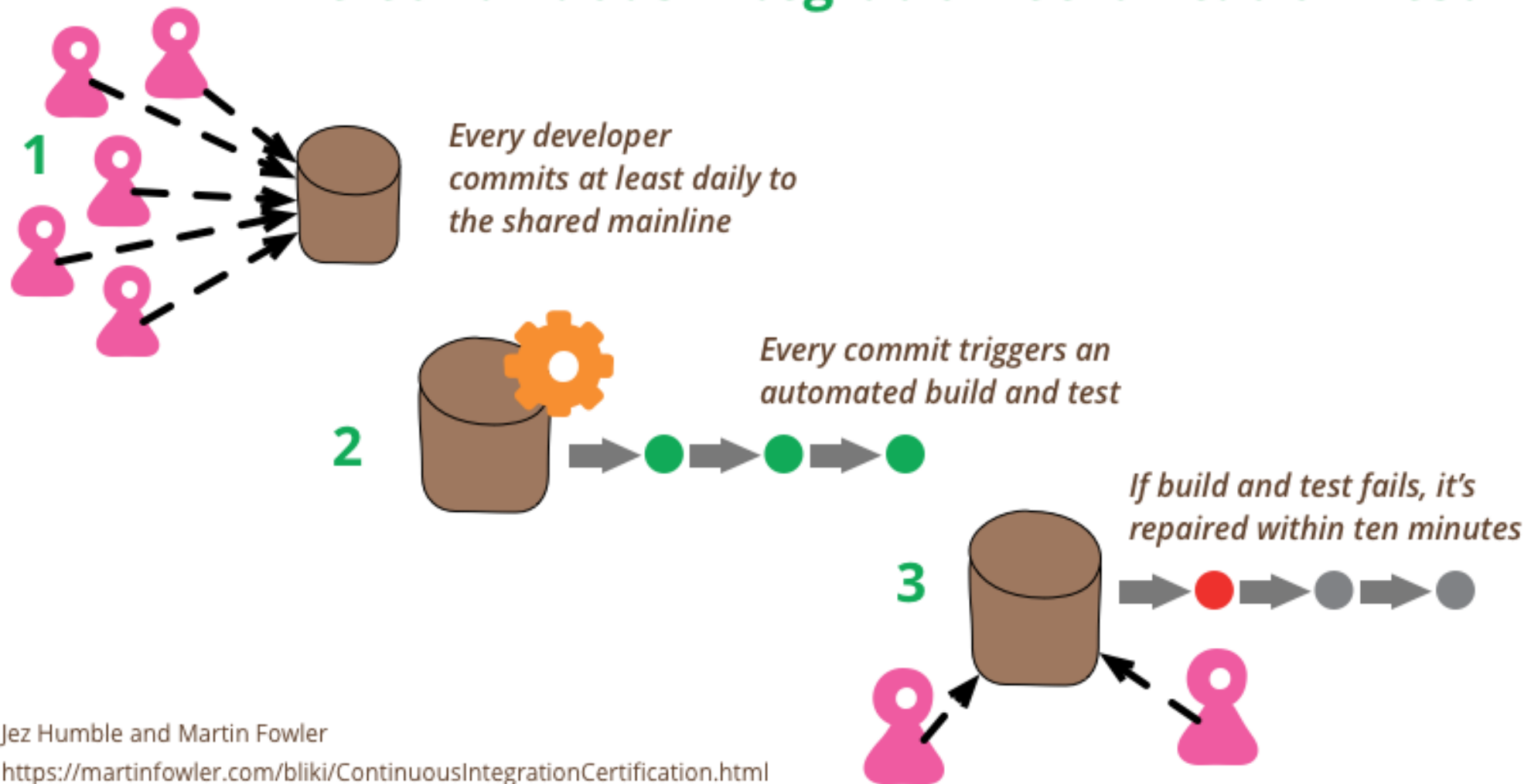
Continuous Integration

Z.B. Trunk-based Development



- ✓ Mergekonflikte sind kleiner
- ✓ Refactorings sind einfacher
- ✓ Wenig Aufwand durch branching
- ✓ Keine Integrationsphasen
- ✓ Keine "verrottenden" Features in branches

The Continuous Integration Certification Test



Jez Humble and Martin Fowler

<https://martinfowler.com/bliki/ContinuousIntegrationCertification.html>

CI ist mehr als ein Buildserver.

feedback

CI = schnelles Feedback

Wie weit
kann man
mit CI gehen?

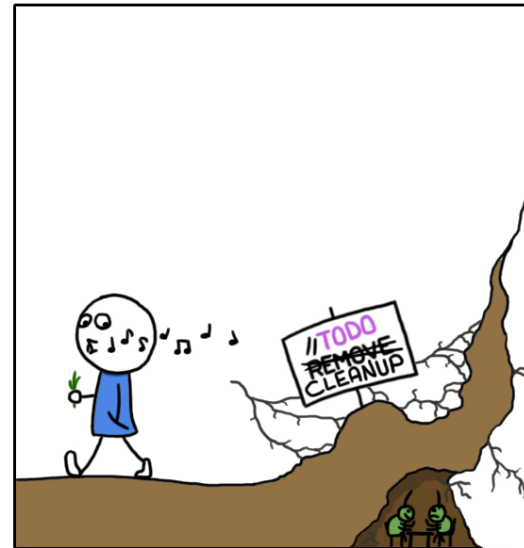
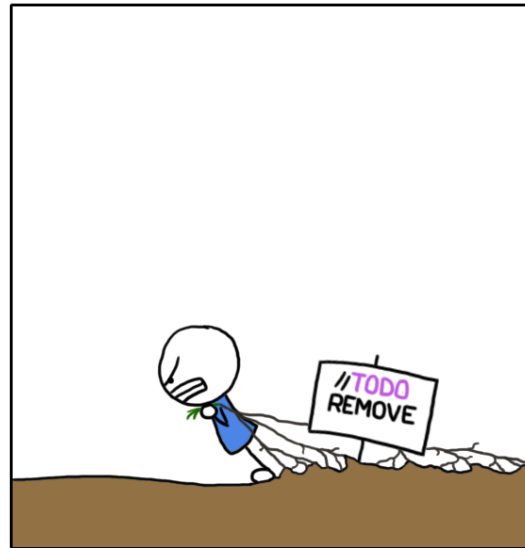
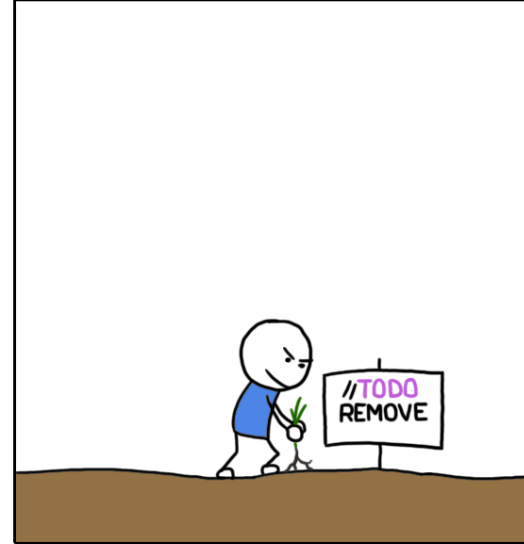
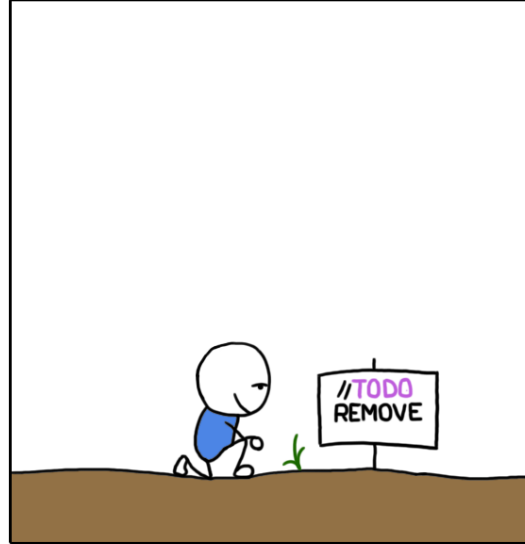


Großes Refactoring –

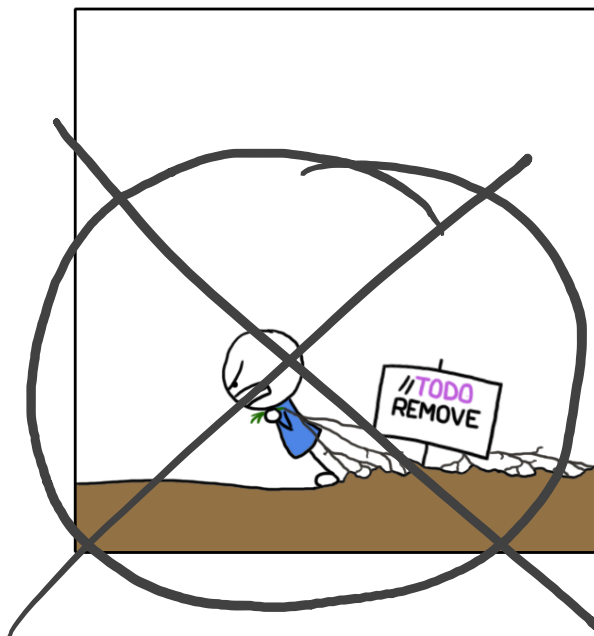
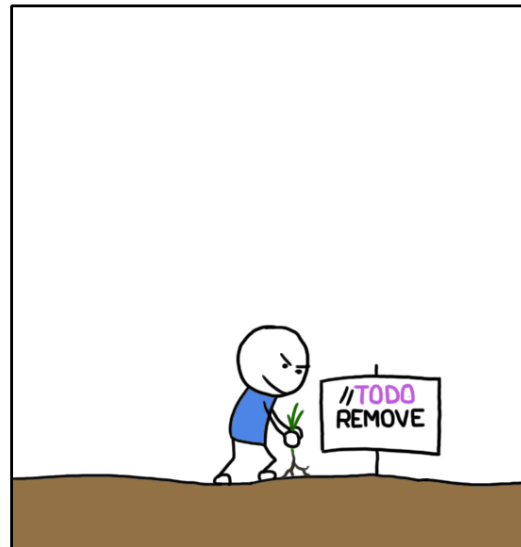
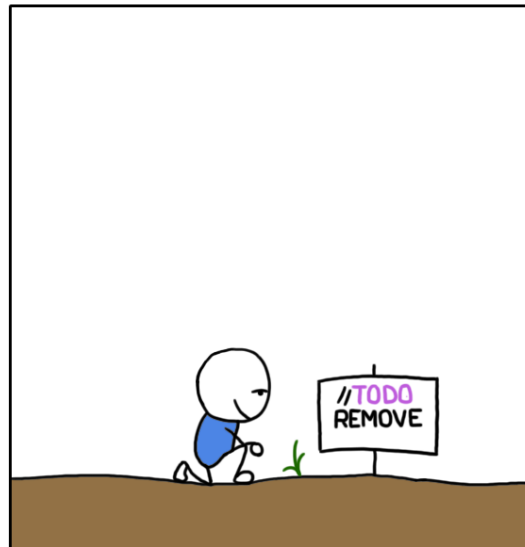
Kleine Schritte

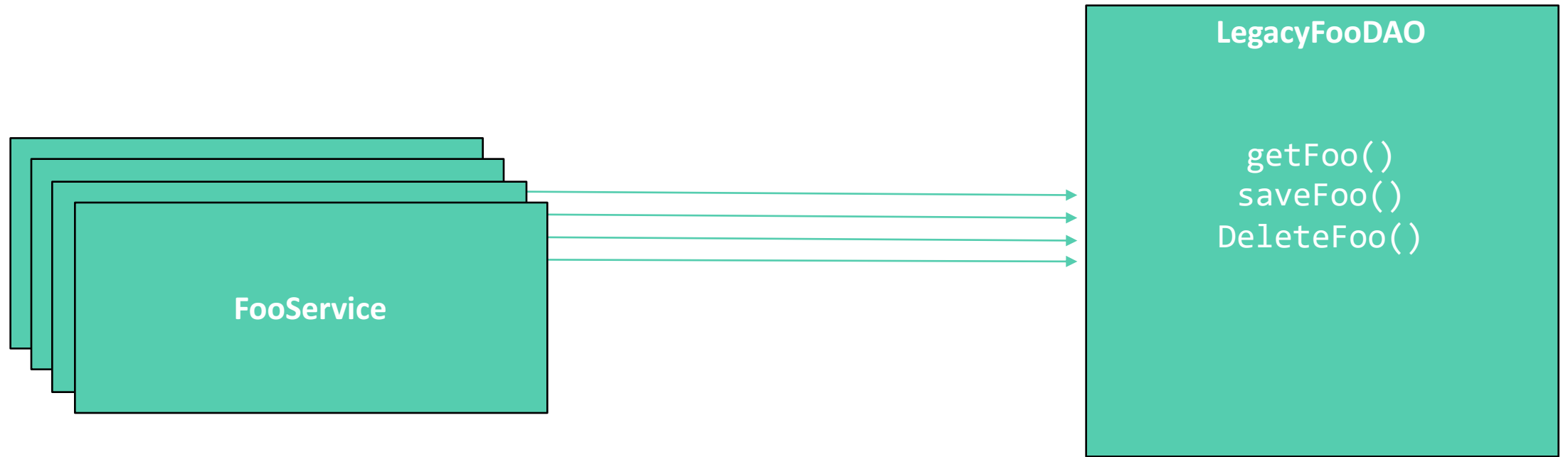


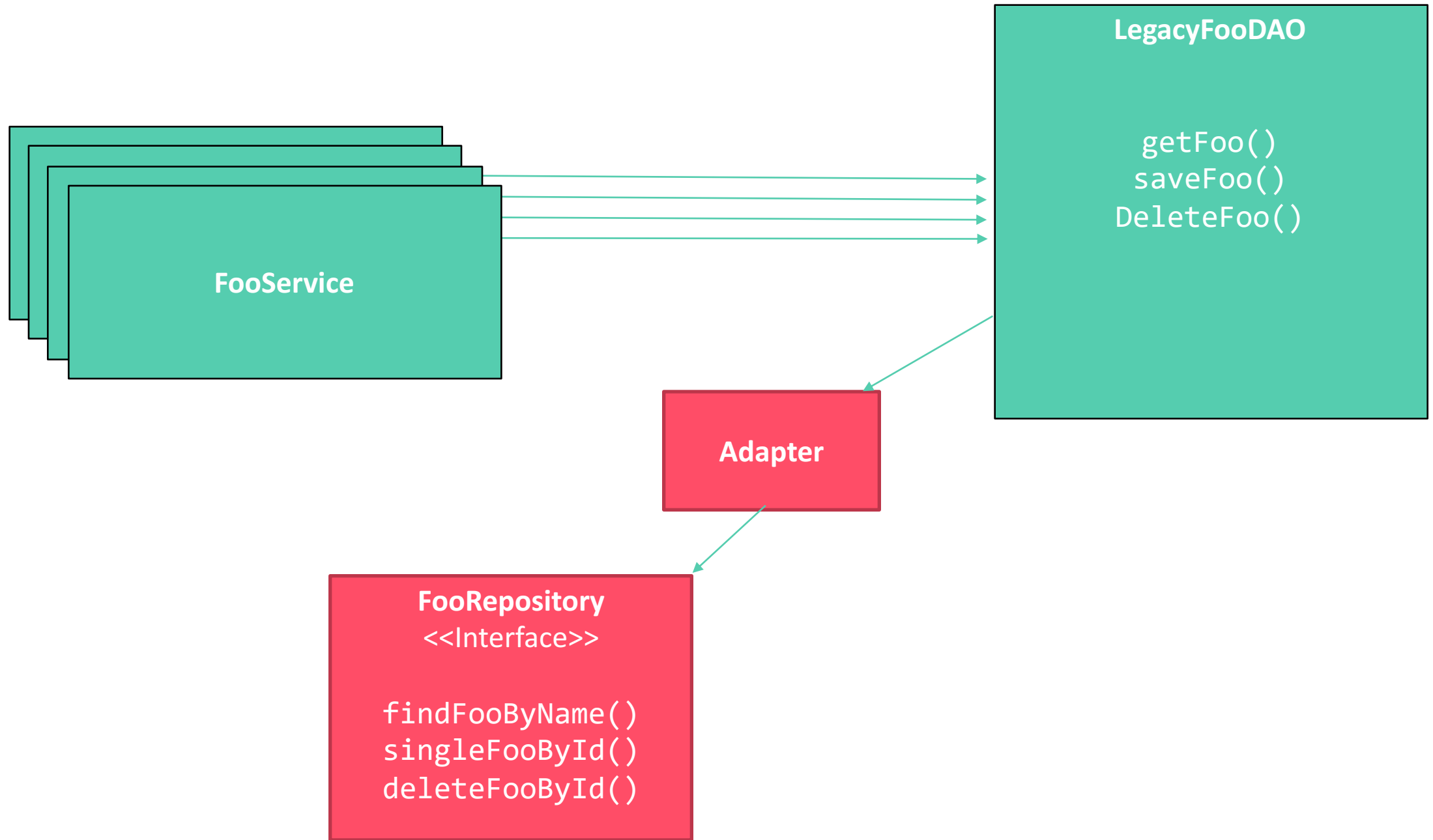
// TODO

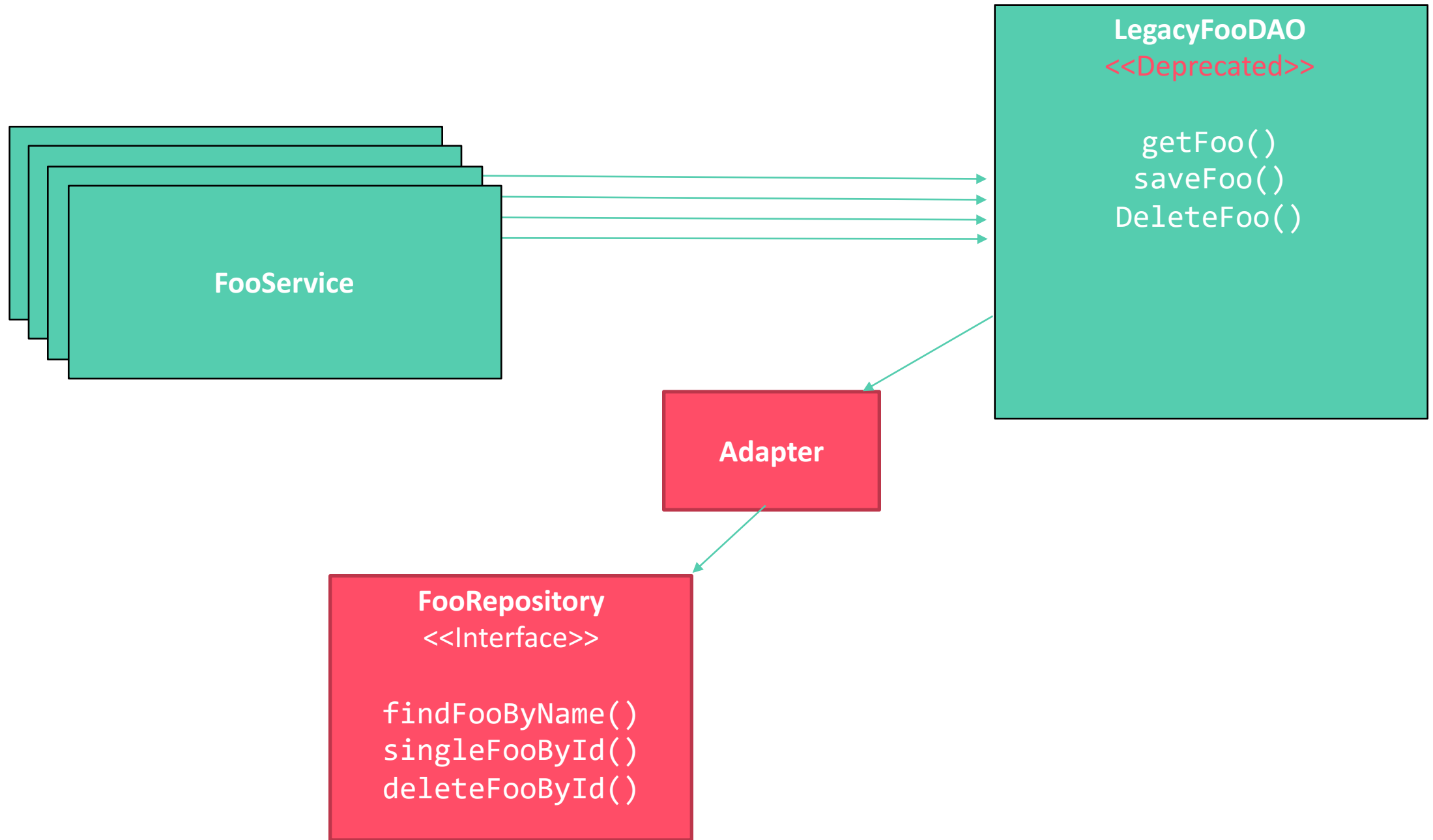


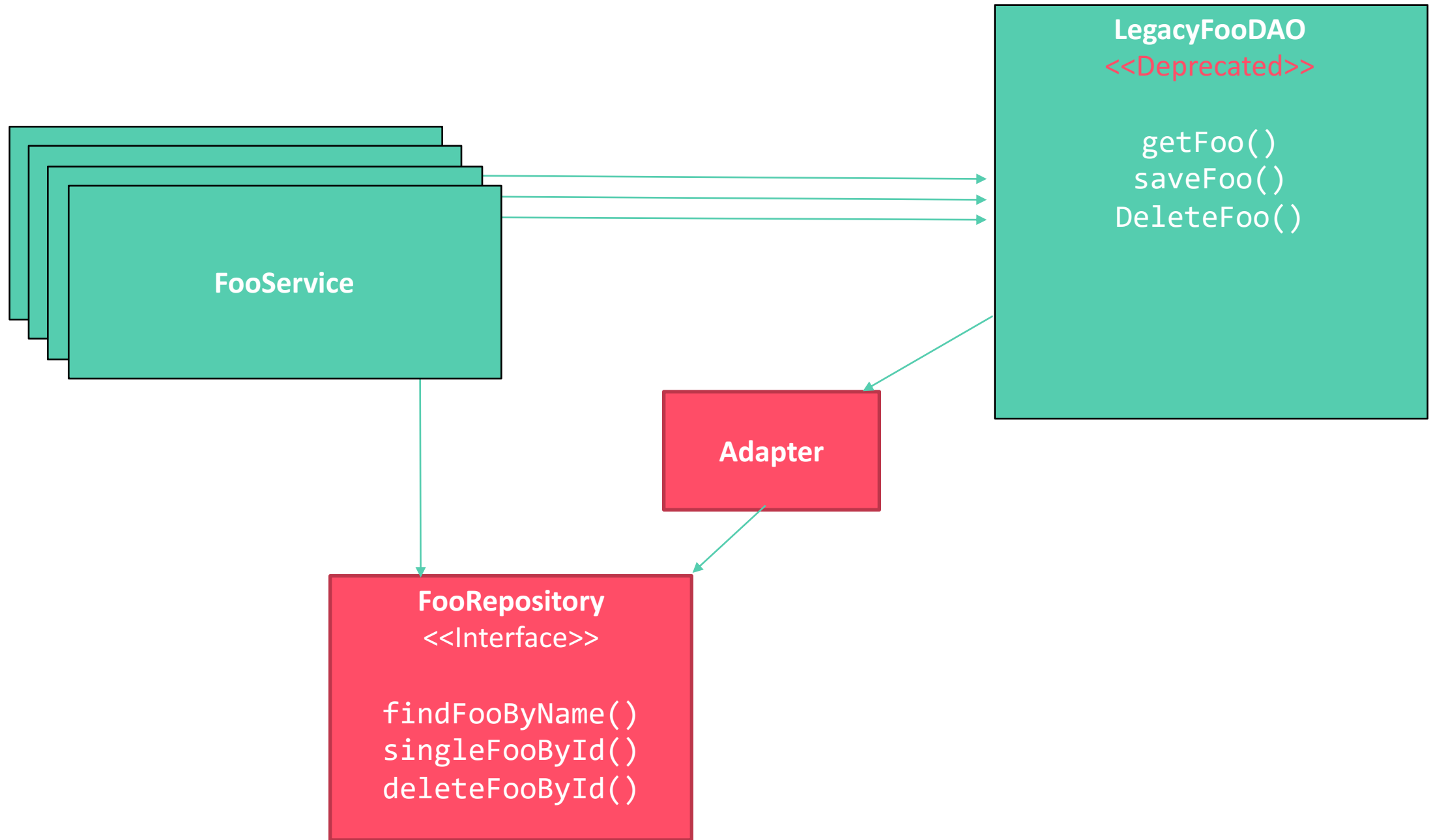
// TODO

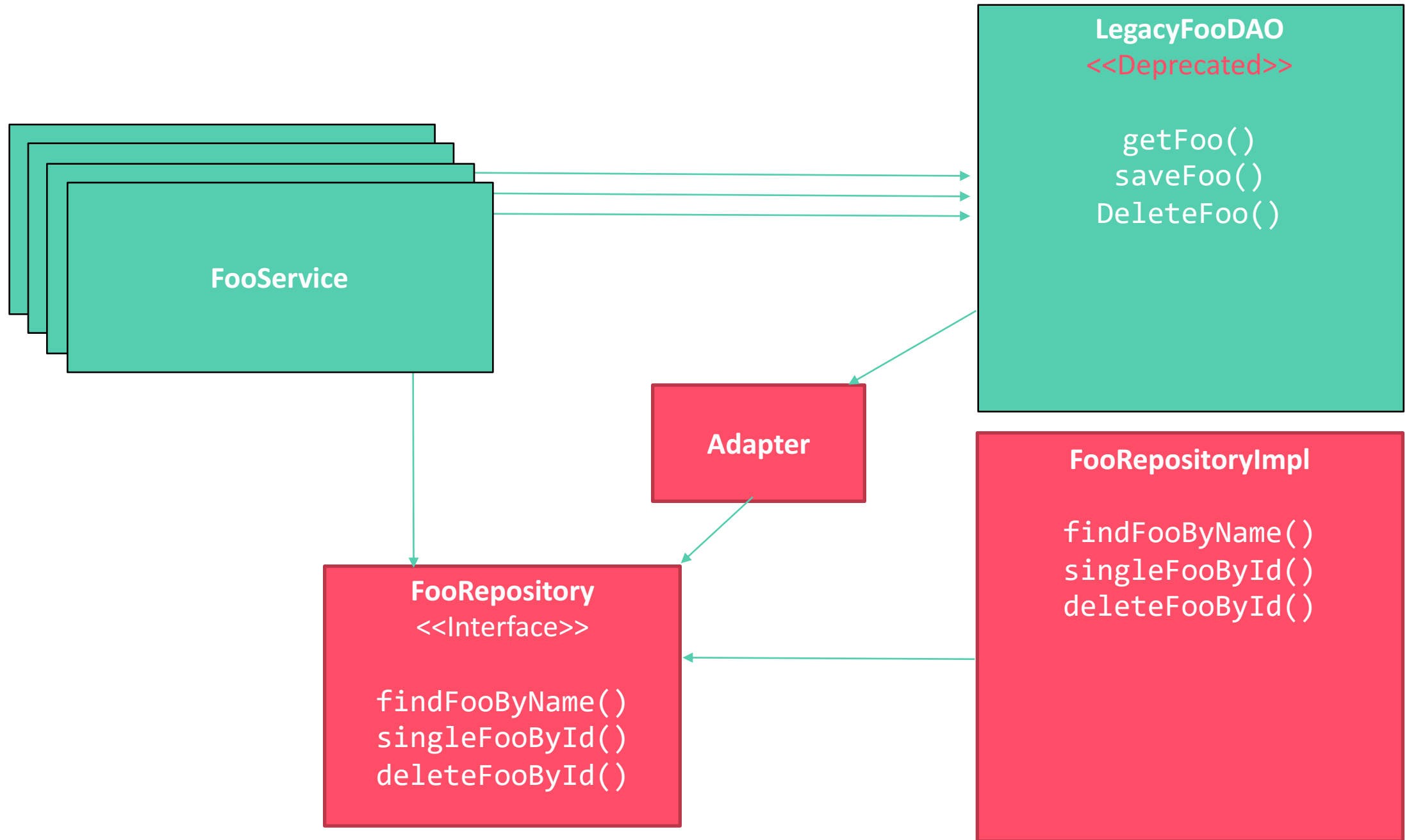


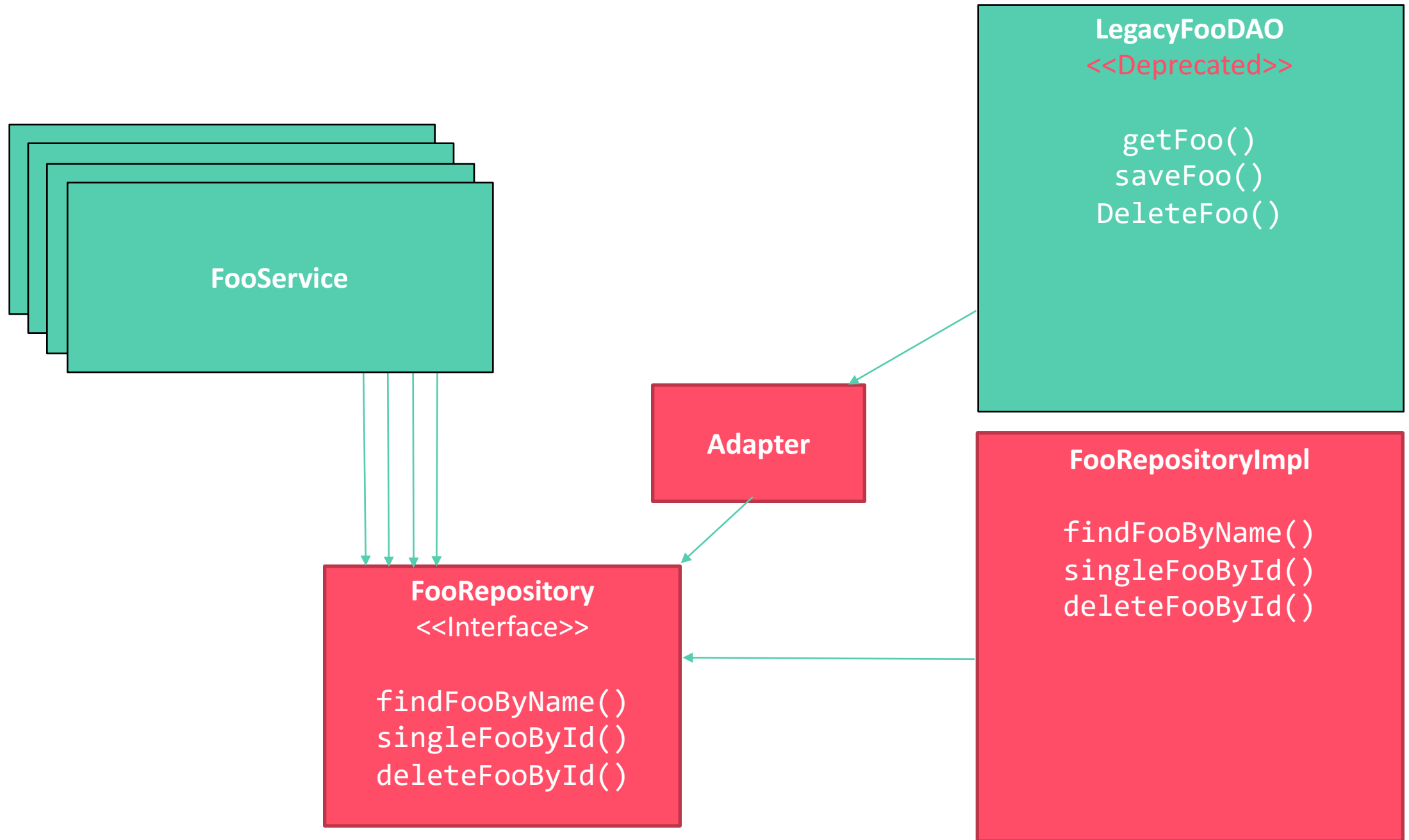


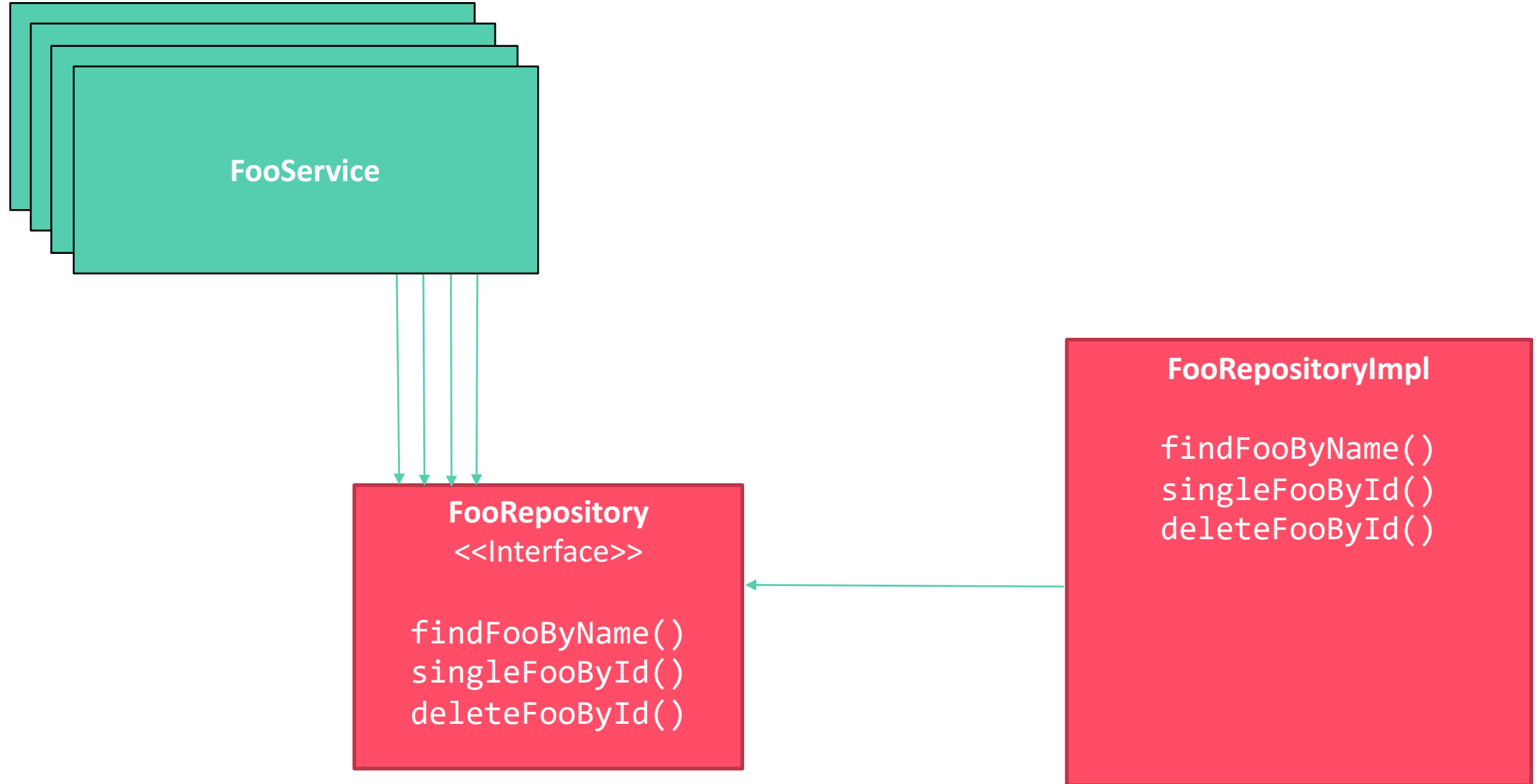












Branch by Abstraction

- ✓ **Lauffähiger Zustand während Refactoring**
- ✓ **@Deprecated + Boy Scout Rule**
=> Refactoring mit dem ganzen Team
- ✓ **Test der neuen Impl. mit Hilfe der alten**

Mehr zum Thema:

<https://printhelloworld.de/posts/refactoring-in-baby-steps/>

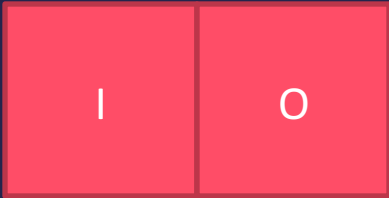
Zu viele Baustellen
vermeiden



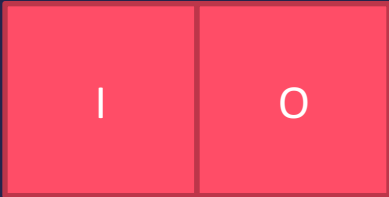
Unfertige Features



Feature Toggles



User Interface

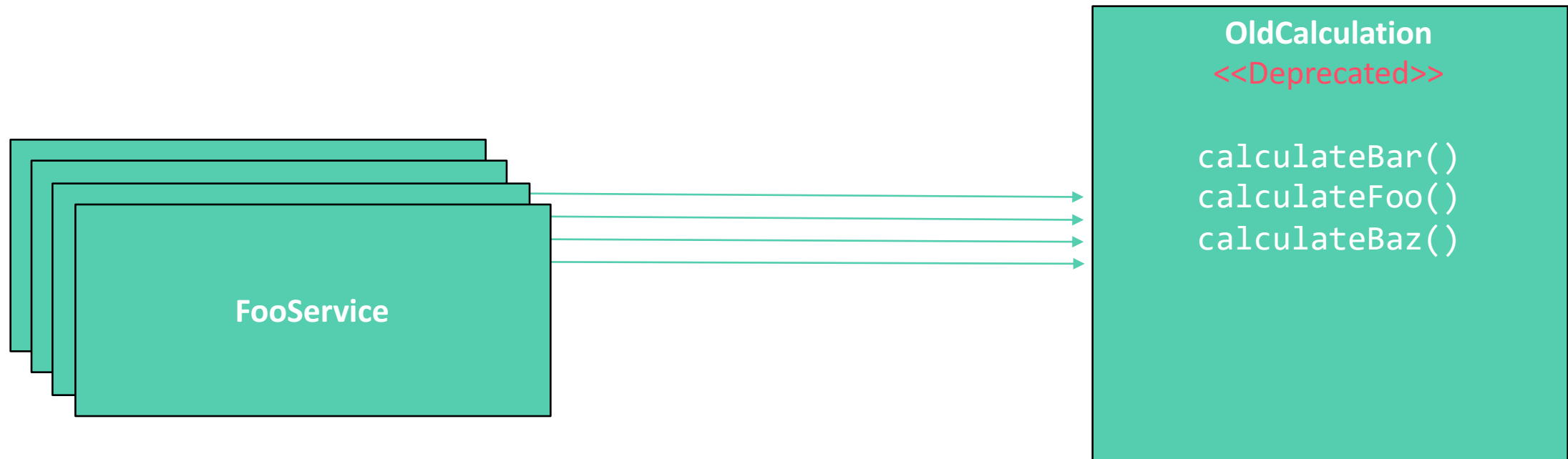


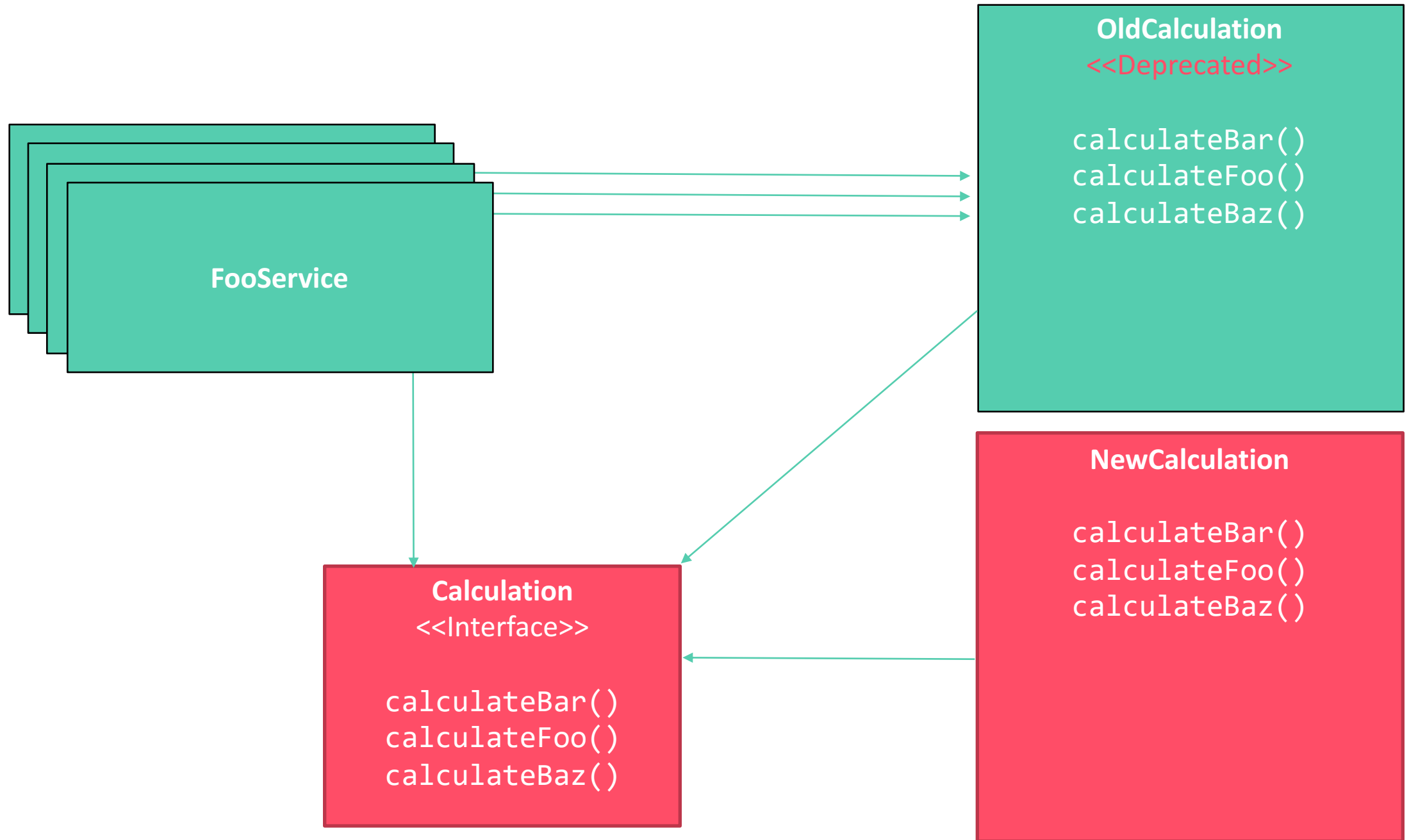
Business Logik

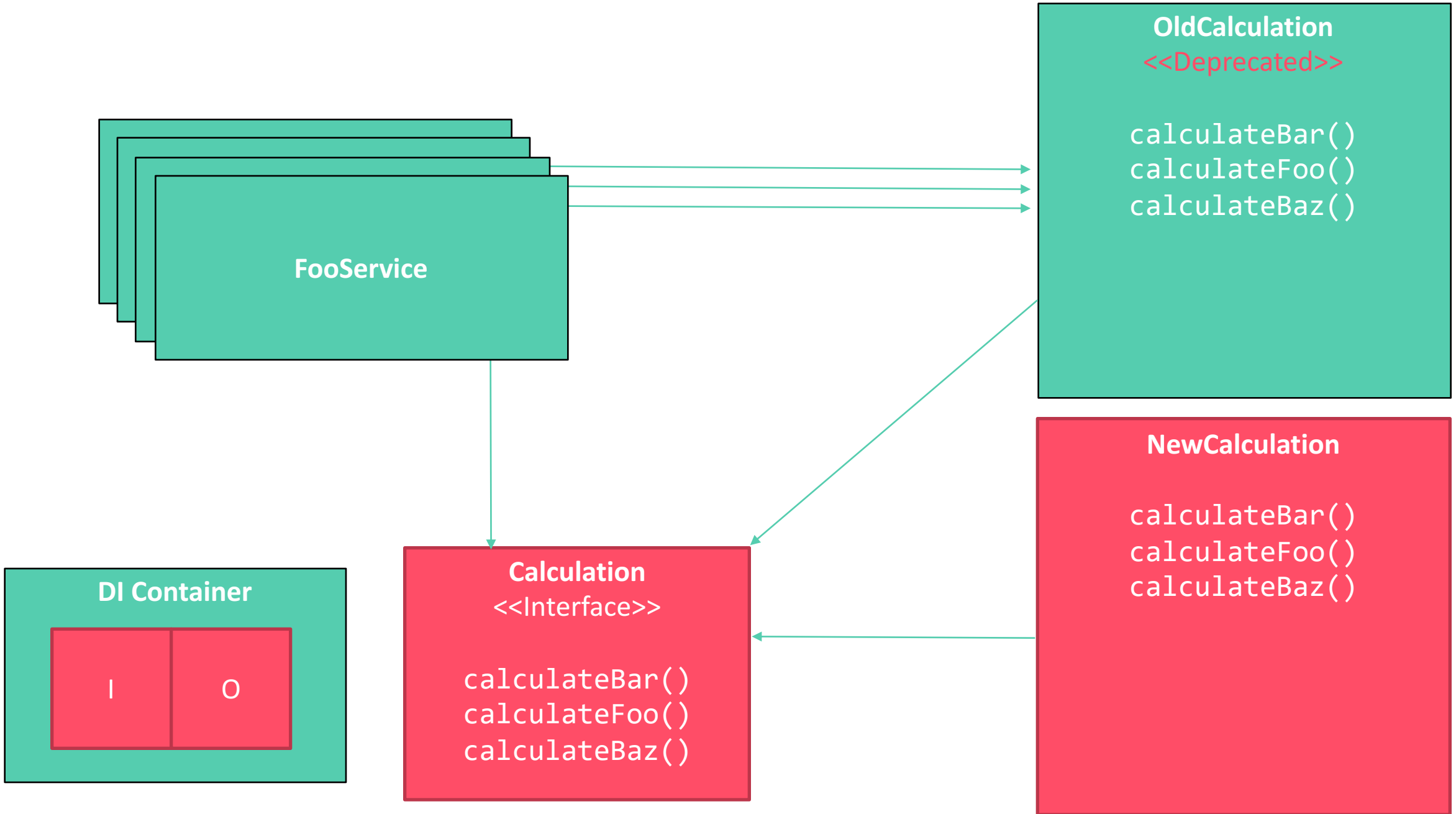
Persistenz

UI Toggles

```
<toggle name="petSurvey">  
  <p>  
    Take our new  
    <a href='petSurvey'>  
      pet survey  
    </a>  
  </p>  
</toggle>
```





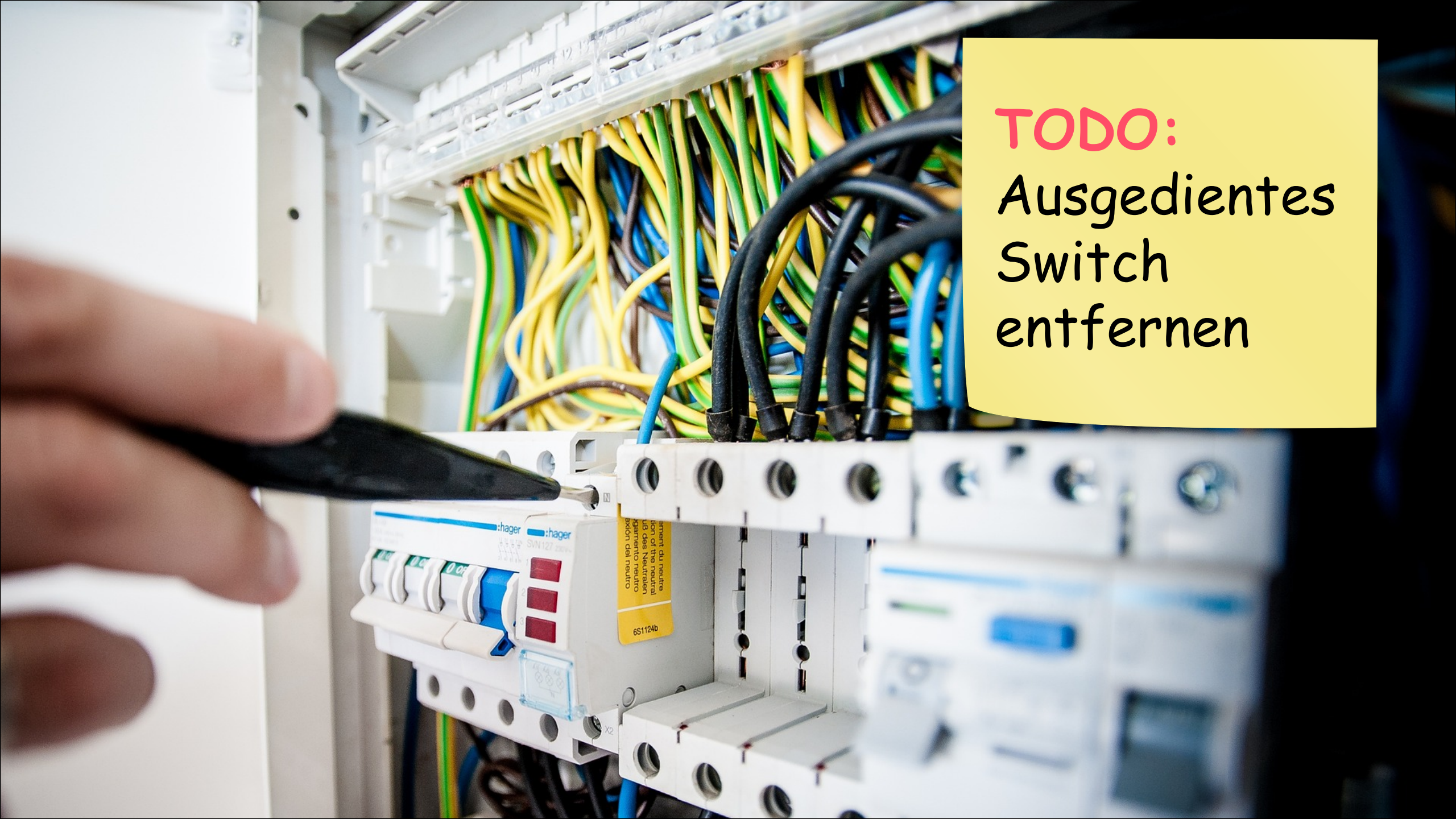


Feature toggles

- ✓ **Lauffähiger, deploybarer Stand im Repository**
- ✓ **Unterstützt in-sprint testing**
- ✓ **Canary releasing**
- ✓ **Release ohne Deployment möglich**

Mehr zum Thema:

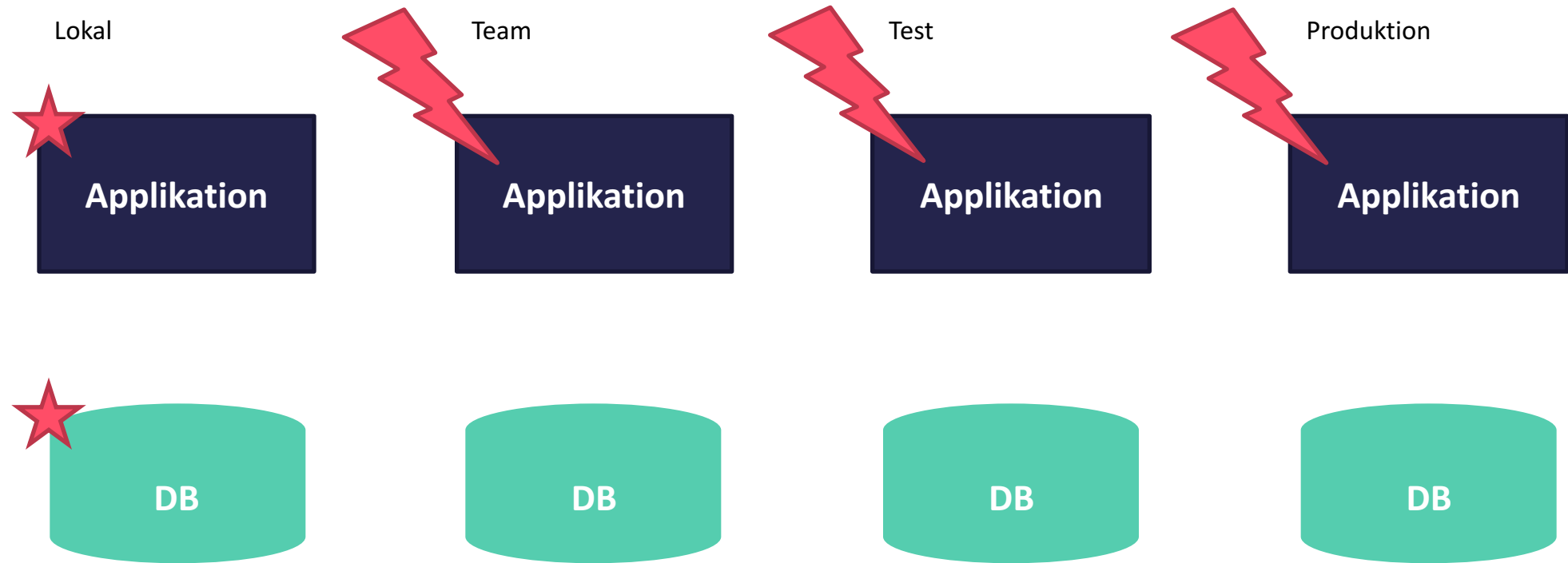
<https://www.martinfowler.com/articles/feature-toggles.html>

A close-up photograph of an electrical control panel. A hand is using a black-handled screwdriver to remove a white switch from a rack. The switch is labeled 'hager' and 'SW 127 200V'. Above the switch, there are several rows of terminal blocks with many colorful wires (yellow, green, blue, brown) connected to them. The background is slightly blurred, showing more of the panel and other components.

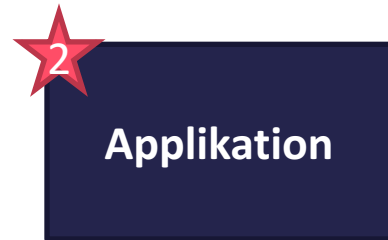
TODO:
Ausgedientes
Switch
entfernen



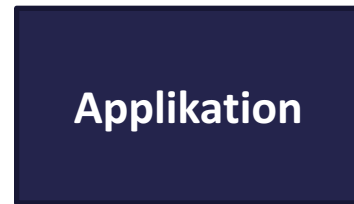
DB Schema-
änderungen



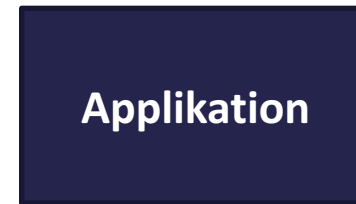
Lokal



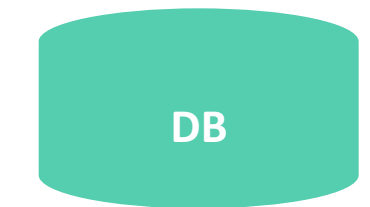
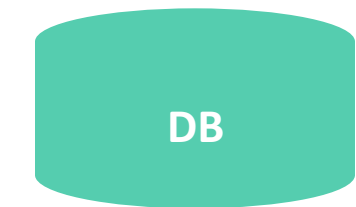
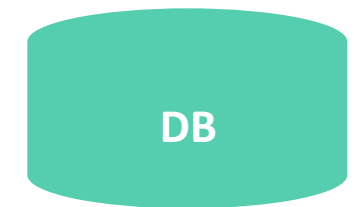
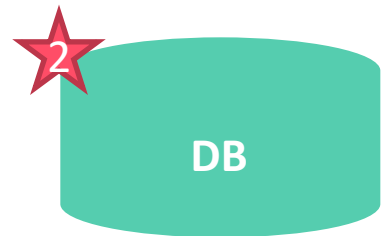
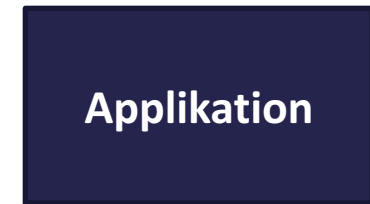
Team



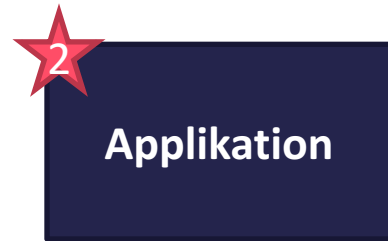
Test



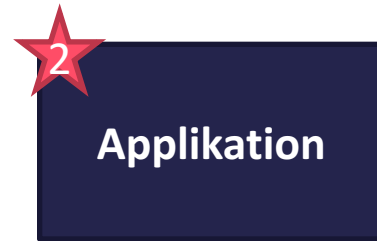
Produktion



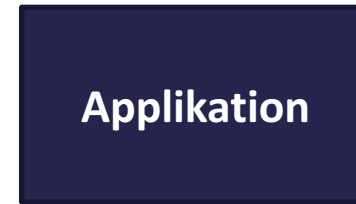
Lokal



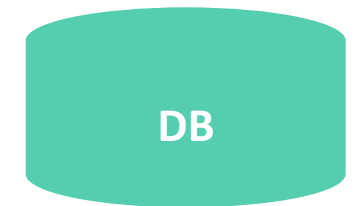
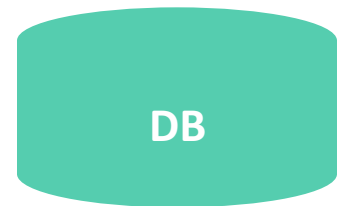
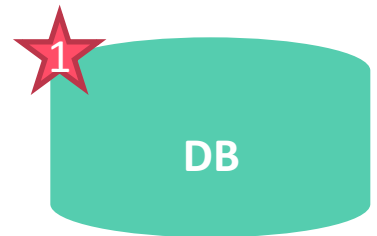
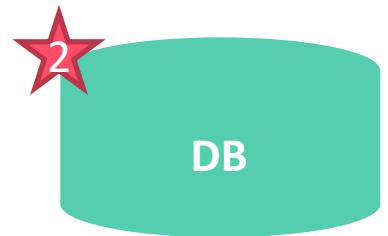
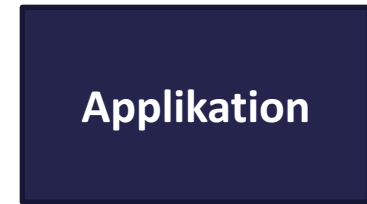
Team



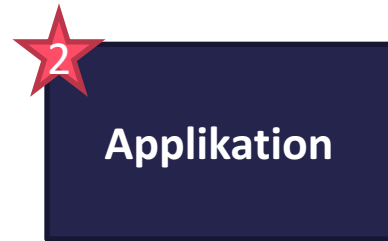
Test



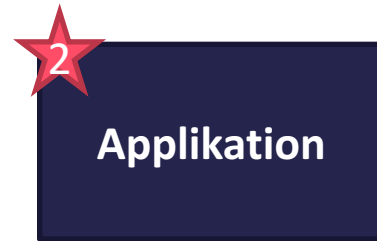
Produktion



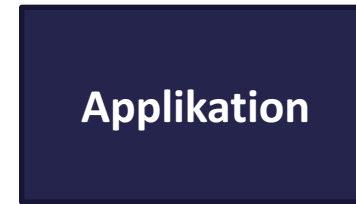
Lokal



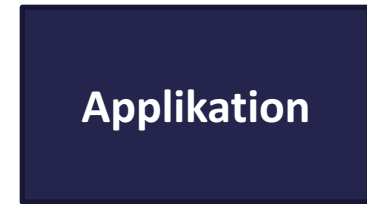
Team

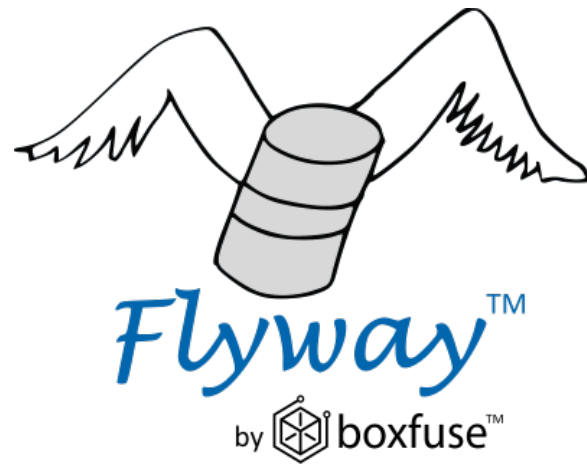


Test



Produktion





www.flyway.com



<https://www.liquibase.org/>

Database Schema Migrationen

- ✓ **Schema passt immer zum Code**
- ✓ **Inkrementelles, kollaboratives entwickeln**
- ✓ **Schema in Versionskontrolle**

Mehr zum Thema:

<https://www.sandra-parsick.de/publication/cdbi-flyway/>



Deployment absprechen



Warum nicht "latest"?

14.09.2017 13:29 Uhr | Security

Equifax-Hack: Angreifer über Apache-Struts-Lücke eingestiegen

Untersuchungen zeigen, dass Equifax es offensichtlich versäumt hat, Sicherheitsupdates für eine kritische Lücke zu installieren. Darüber hinaus ist es zu einem weiteren Datenleck gekommen.

Von Dennis Schirmmacher

🔊 🖨️ 💬 34



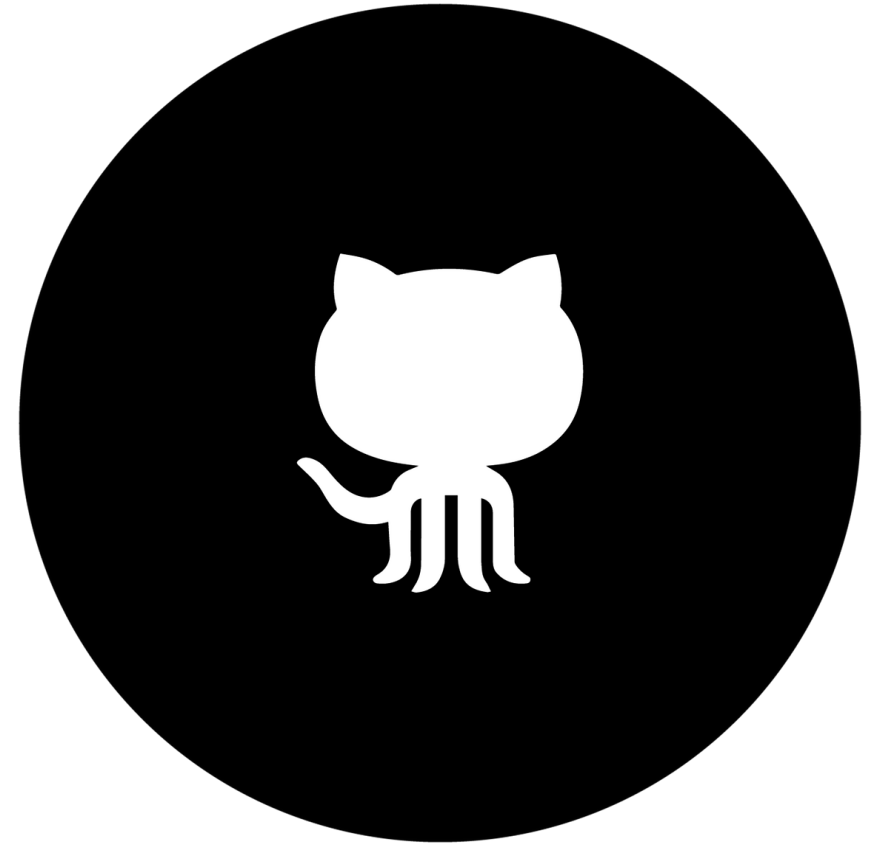
(Bild: medithIT, CC BY 2.0)

"... hatte das Unternehmen
rund eineinhalb Monate Zeit,
das abgesicherte Update zu
installieren..."

Libraryversionsnummern
erhöhungsangst

„In total the project took **a year and a half** to upgrade from Rails 3.2 to Rails 5.2“

"Lessons learned:
**Upgrade early and
upgrade often."**





Canary Pipelines

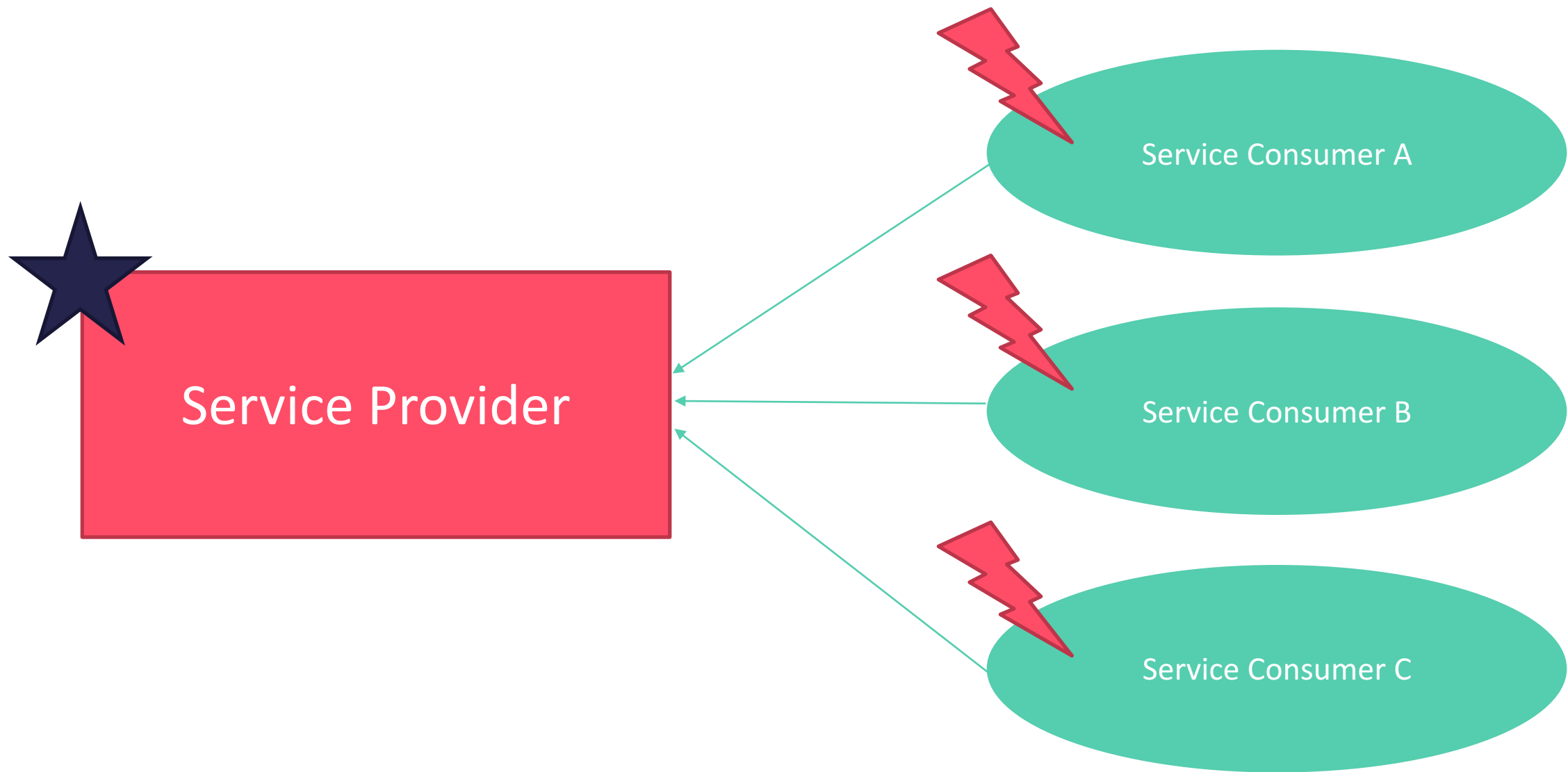
- ✓ Feedback über "breaking changes"
- ✓ Notification über Patches möglich
- ✓ Deployment Build bleibt stabil

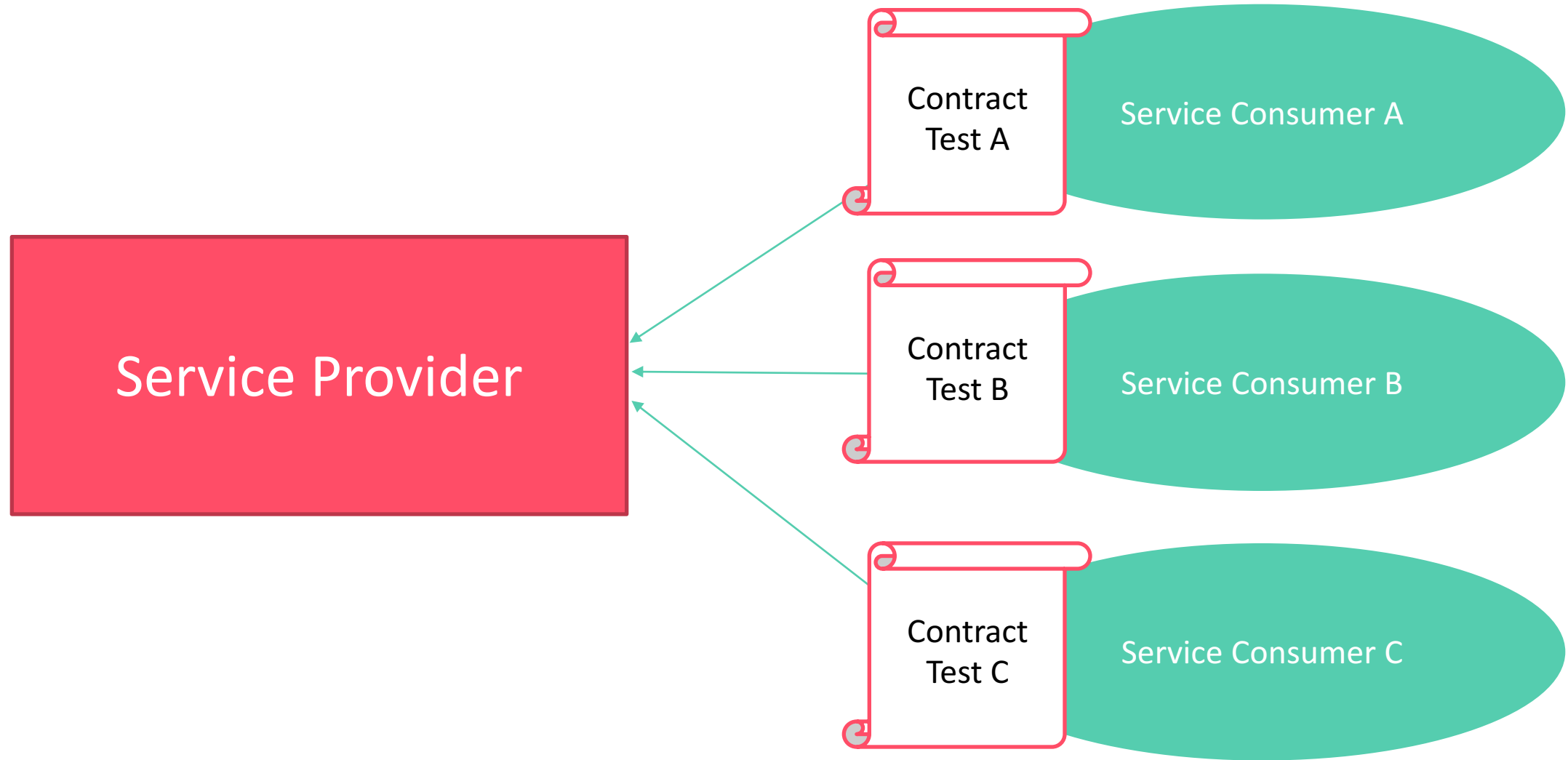
Mehr zum Thema:

<https://printhelloworld.de/posts/continuous-integration-with-dependencies/>

Wenn sich
Services ändern

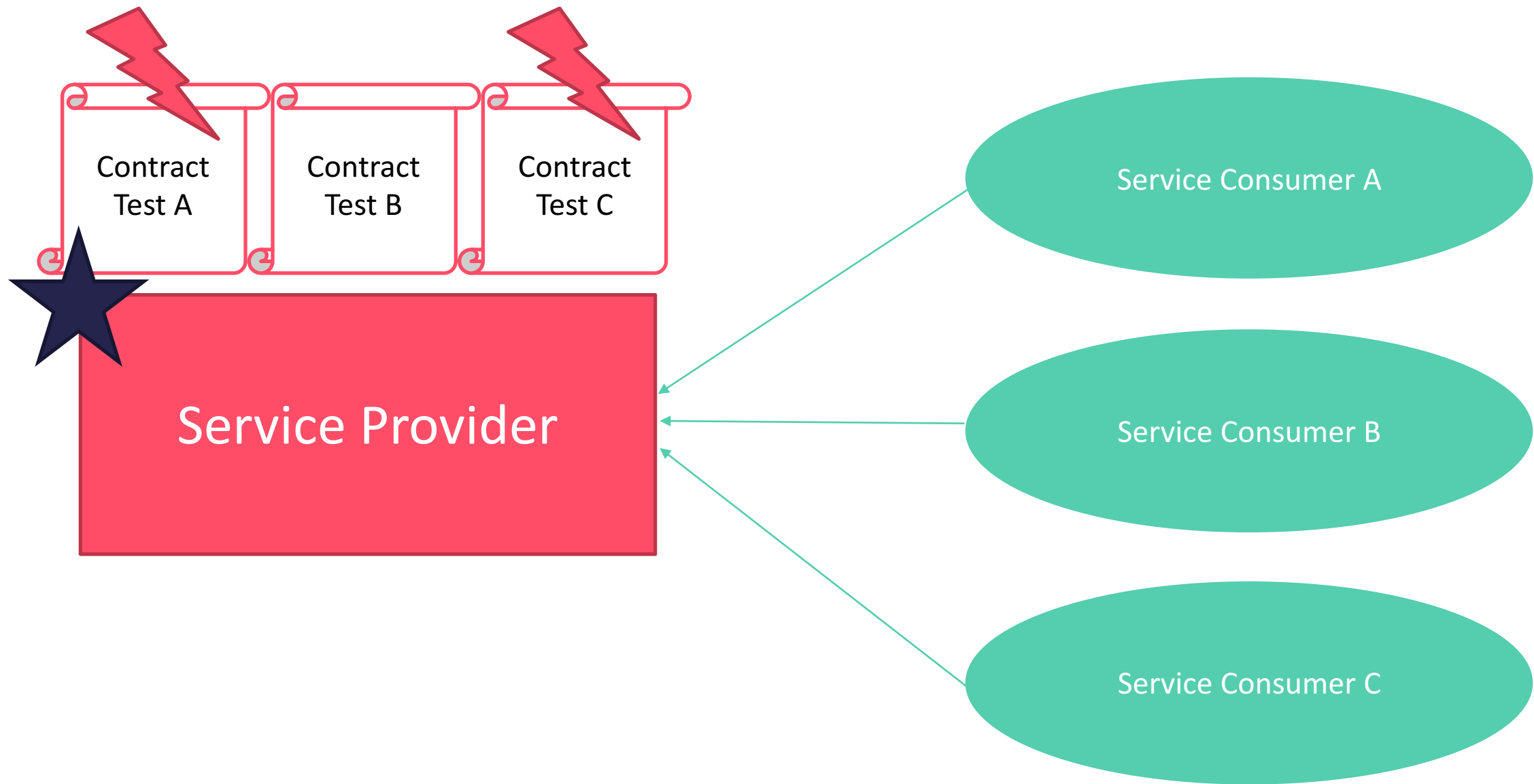






Contract Test

```
16 class AnimalServiceSpec extends PlaySpecification {
17
18     "Animal Service" should {
19         val port = 9000
20         def url = s"http://localhost:$port"
21         def animalService = new AnimalService(url)
22         val executionContext = play.api.libs.concurrent.Execution.Implicits.defaultContext
23
24         val alligatorPact = MakePact()
25             .withProvider("Animal_Service")
26             .withConsumer("Zoo_App")
27
28         "List All Animals" in {
29             def alligatorJson = """[{"name": "Bob"}]"""
30             val state = "there are alligators"
31             val pact: Pact = alligatorPact.withInteractions(
32                 given(state)
33                     .uponReceiving(
34                         description = "a request for animals",
35                         method = "GET",
36                         path = "/animals")
37                     .willRespondWith(
38                         200,
39                         Map("Content-Type" -> "application/json; charset=UTF-8"),
40                         alligatorJson)
41             ).build
42
43             pact.runConsumer(PactServerConfig(port), state) {
44                 Await.result(animalService.animals(executionContext), Duration(10, "s")) must beEqualTo(Seq(Alligator("Bob")))
45             } must beEqualTo(PactVerification.PactVerified).await
46         }
47     }
```



Consumer-Driven Contracts

- Feedback dort, wo die changes passieren
- Feedback für wen Changes breaking sind

Mehr zum Thema:

<https://www.youtube.com/watch?v=n5QFe6uwWHI>

feedback

CI = schnelles Feedback

Danke! Fragen?



Johannes Seitz
johannes.seitz@innoq.com

 @ookami86

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin
Germany

Ludwigstr. 180E
63067 Offenbach
Germany

Kreuzstr. 16
80331 München
Germany

c/o WeWork
Hermannstrasse 13
20095 Hamburg
Germany

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
+41 41 743 01 11

Albulastr. 55
8048 Zürich
Switzerland