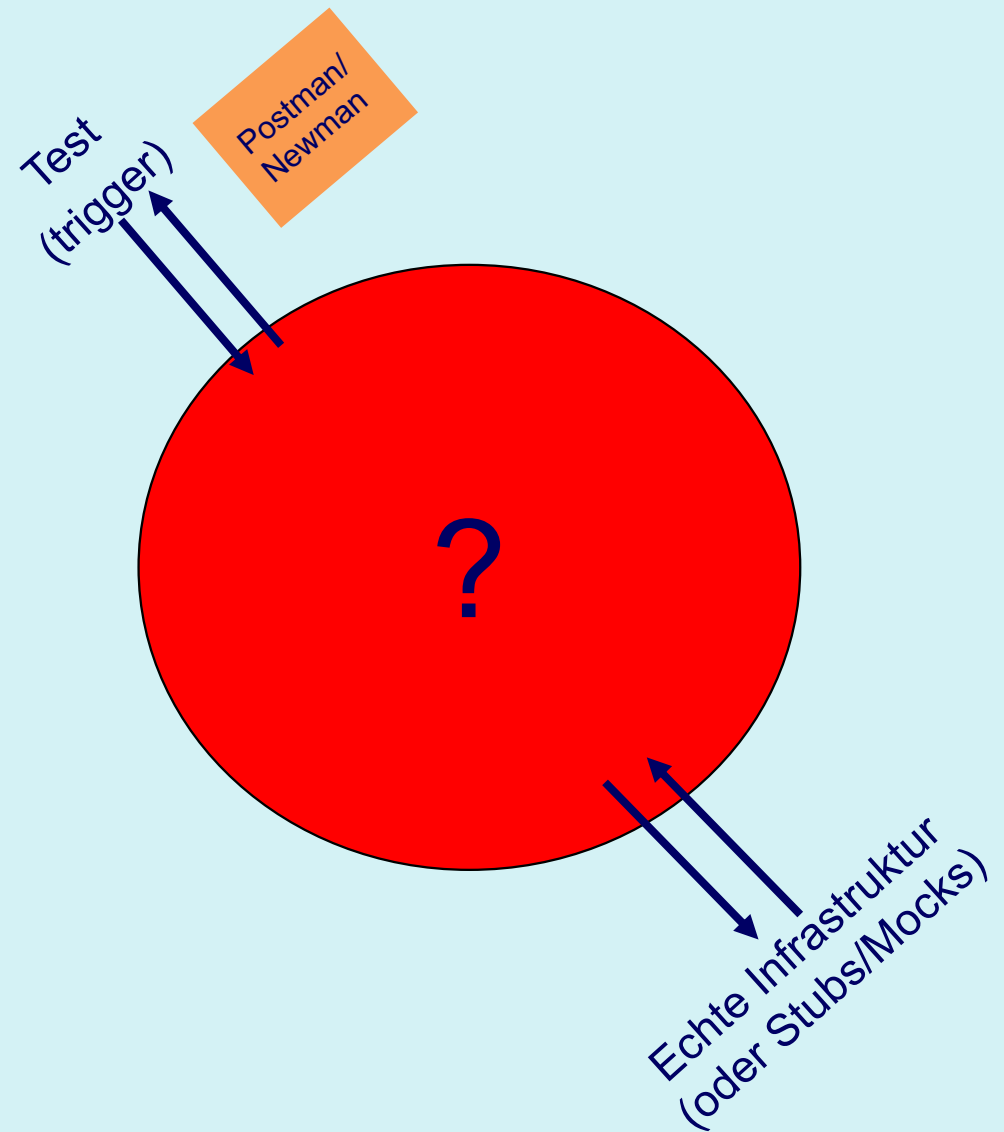
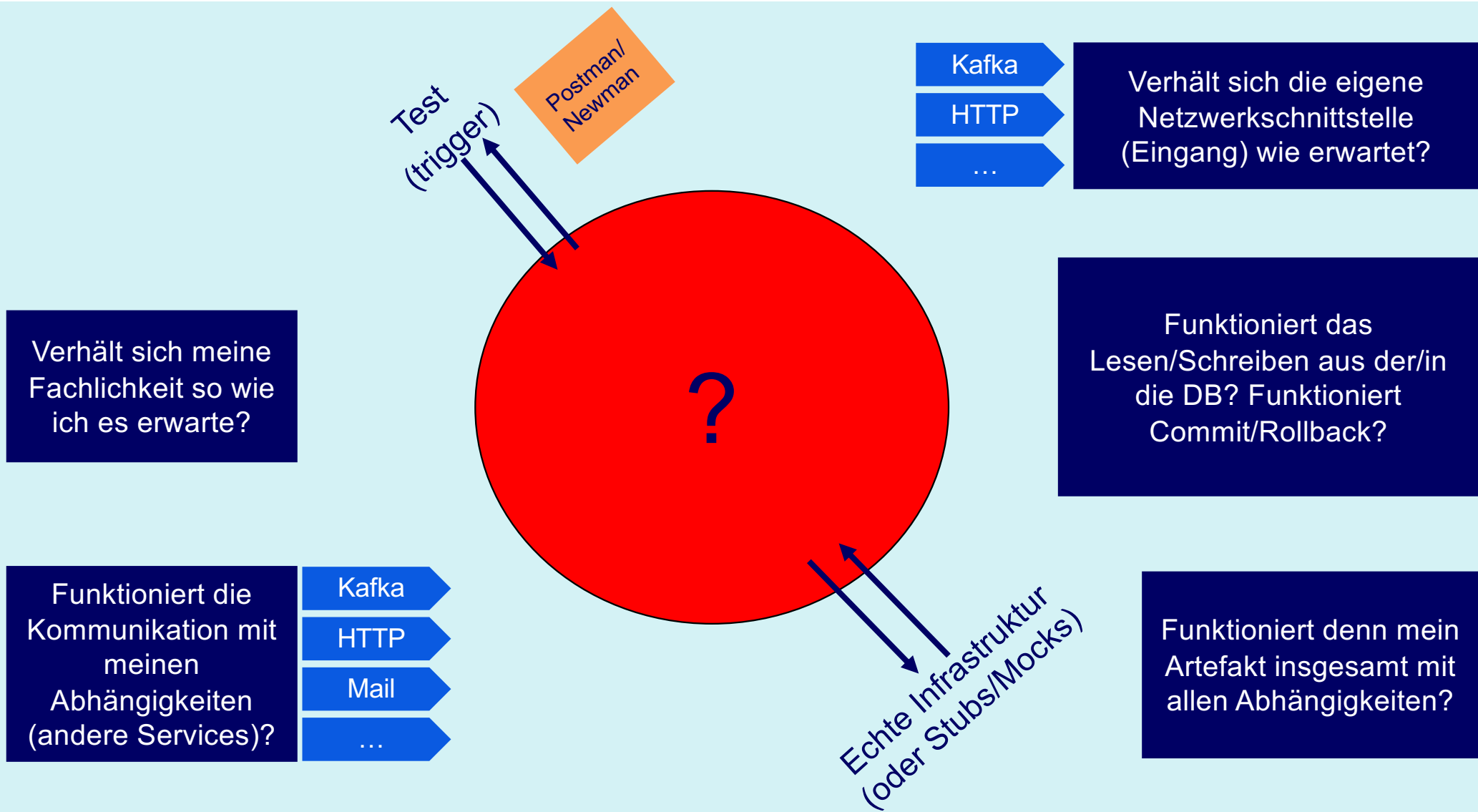


Tests ohne Grenzen: Wenn Modulgrenzen sich in Luft auflösen

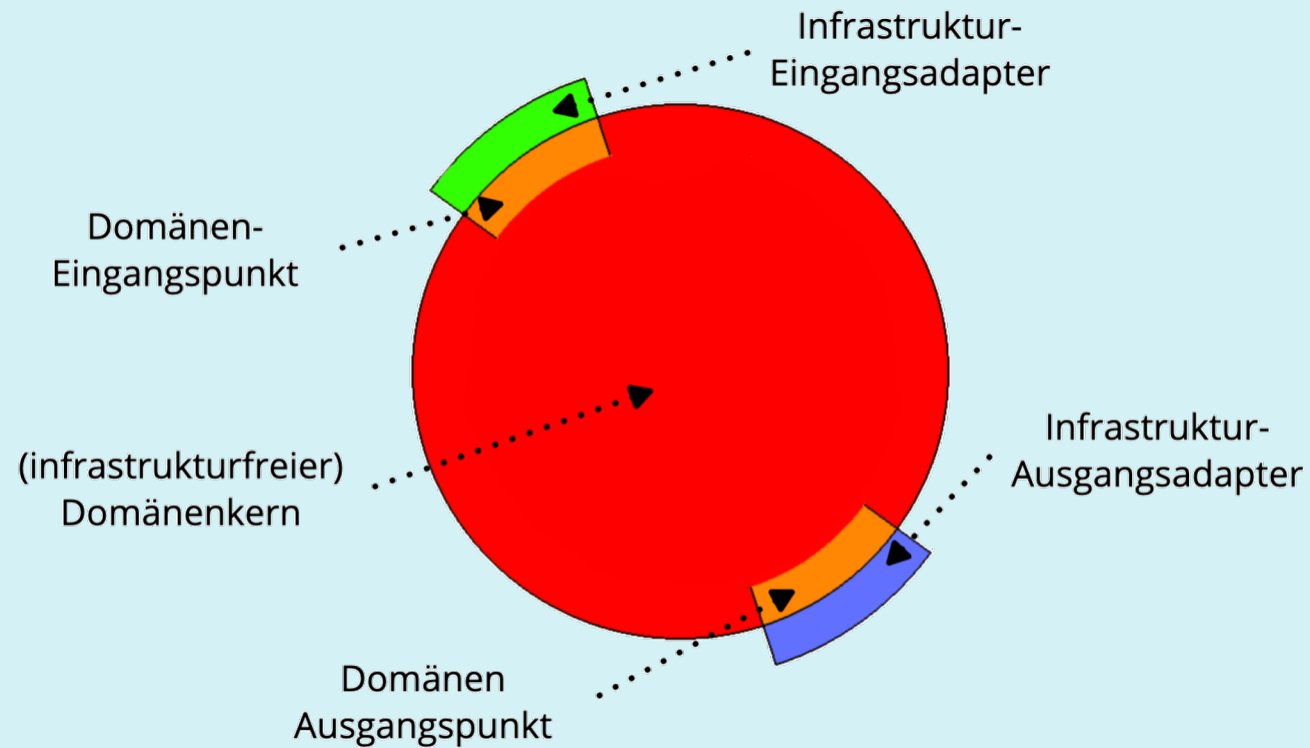
Wie adressieren wir hier unterschiedliche Testziele?

...irgendwo hier drin liegt unser HTTP-Controller
...und irgendwo unsere Domänenlogik
...und irgendwo unsere Persistenzlogik
...und irgendwo unsere Zugriffe auf Umsysteme



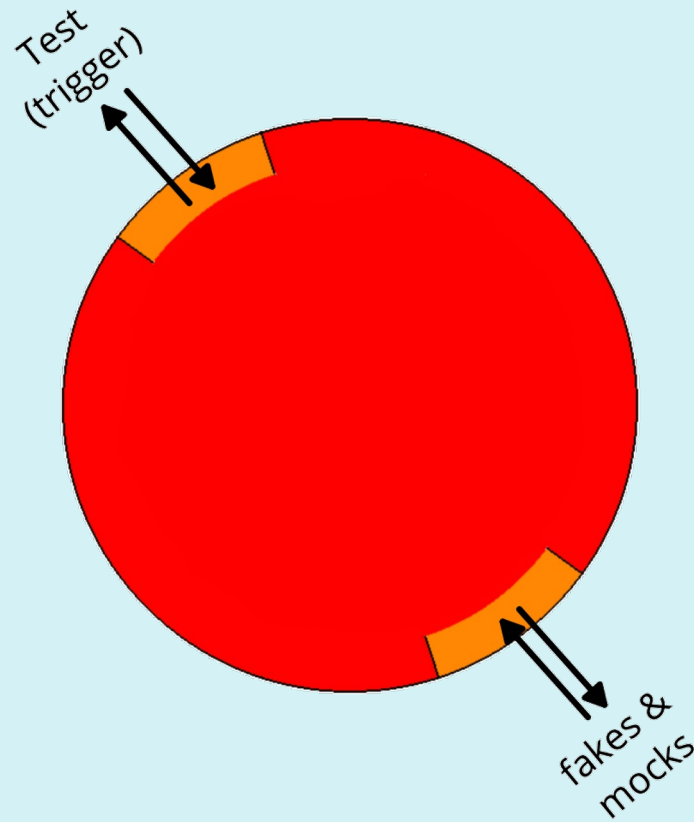


Anatomie eines "Moduls"

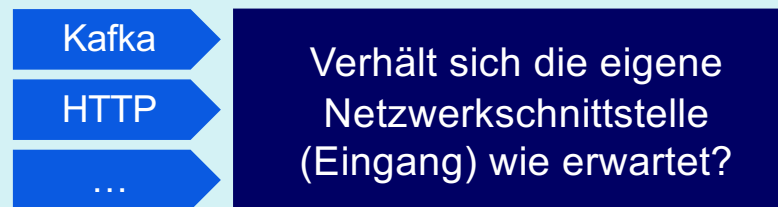
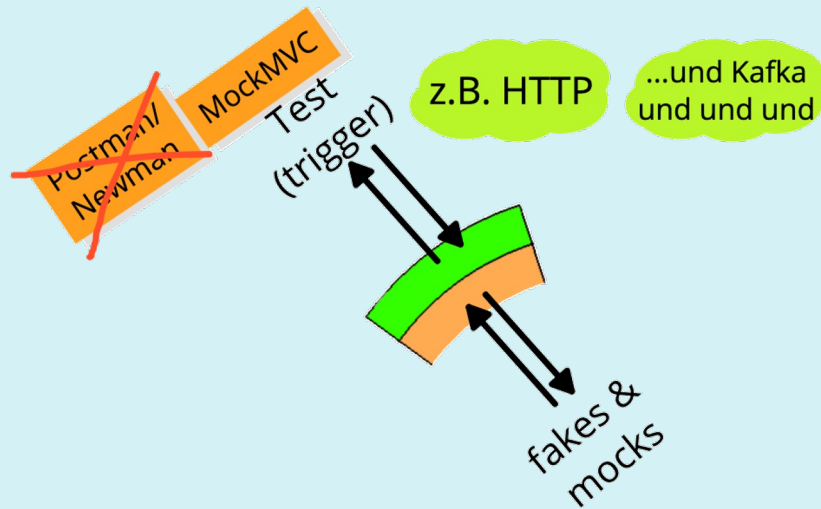


Test des Domänenkerns

Verhält sich meine
Fachlichkeit so wie
ich es erwarte?



Test der/des Eingangsadapter(s)



Test der/des Ausgangsadapter(s)

Funktioniert die Kommunikation mit meinen Abhängigkeiten (andere Services)?

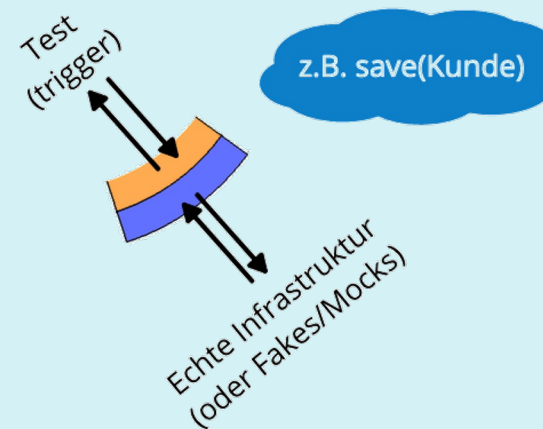
Kafka

HTTP

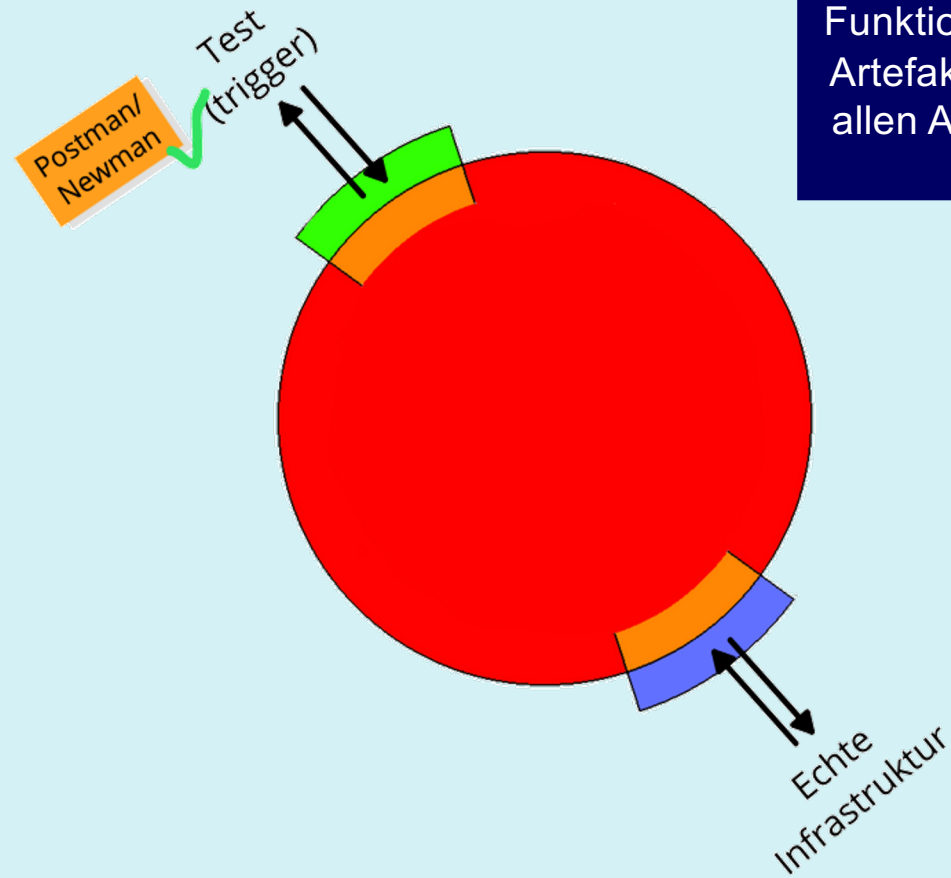
Mail

...

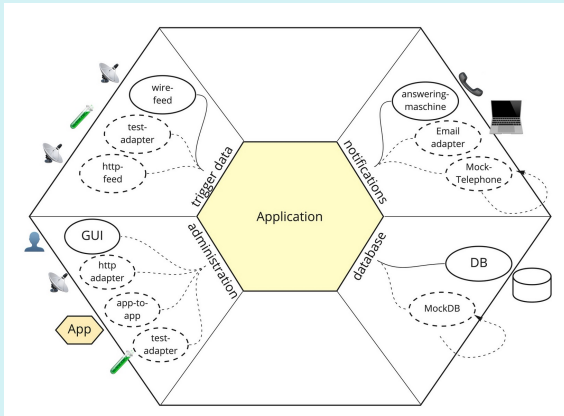
Funktioniert das Lesen/Schreiben aus der/in die DB? Funktioniert Commit/Rollback?



Systemtest








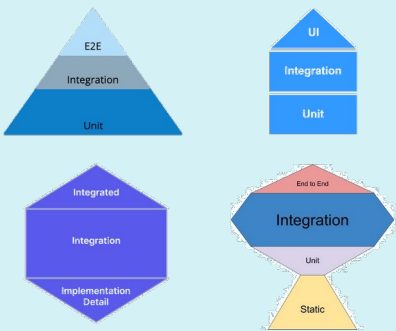
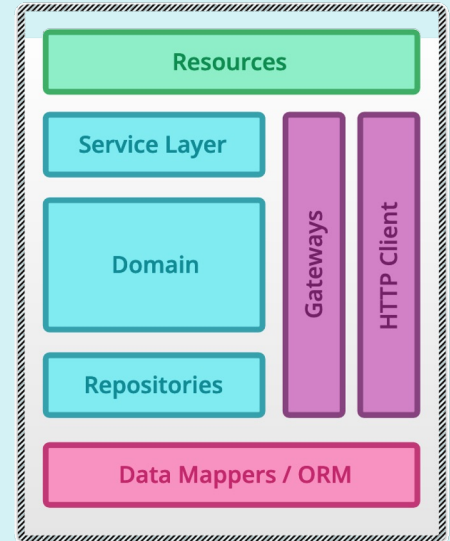
Weiterführendes Material



Ports & Adapters
<https://alistair.cockburn.us/hexagonal-architecture>

Testing Strategies in a Microservice Architecture
<https://martinfowler.com/articles/microservice-testing/>

-  Network Boundary
-  Protocol
-  Domain
-  External
-  Persistence



Defining a Test Strategy for Continuous Delivery
<https://www.simpleorientedarchitecture.com/test-strategy-for-continuous-delivery/>

Ok, soweit in der Theorie! ...und nun?



Tests ohne Grenzen: Wenn Modulgrenzen sich in Luft auflösen | Peter Fichtner | Öffentlich (C1)

Auftrag: Implementiert eine Bundesliga-Tabelle!

Wir sollen...

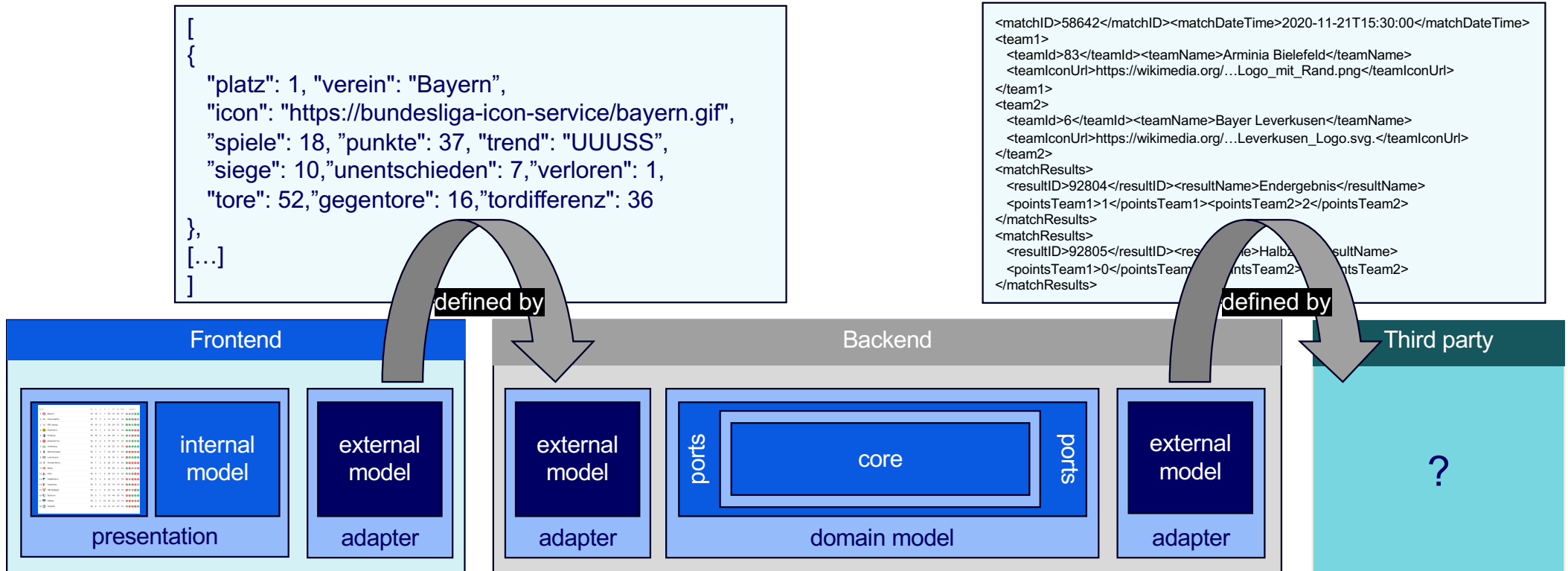
...die Oberfläche implementieren (Frontend)

...die Business-Logik implementieren (Backend)

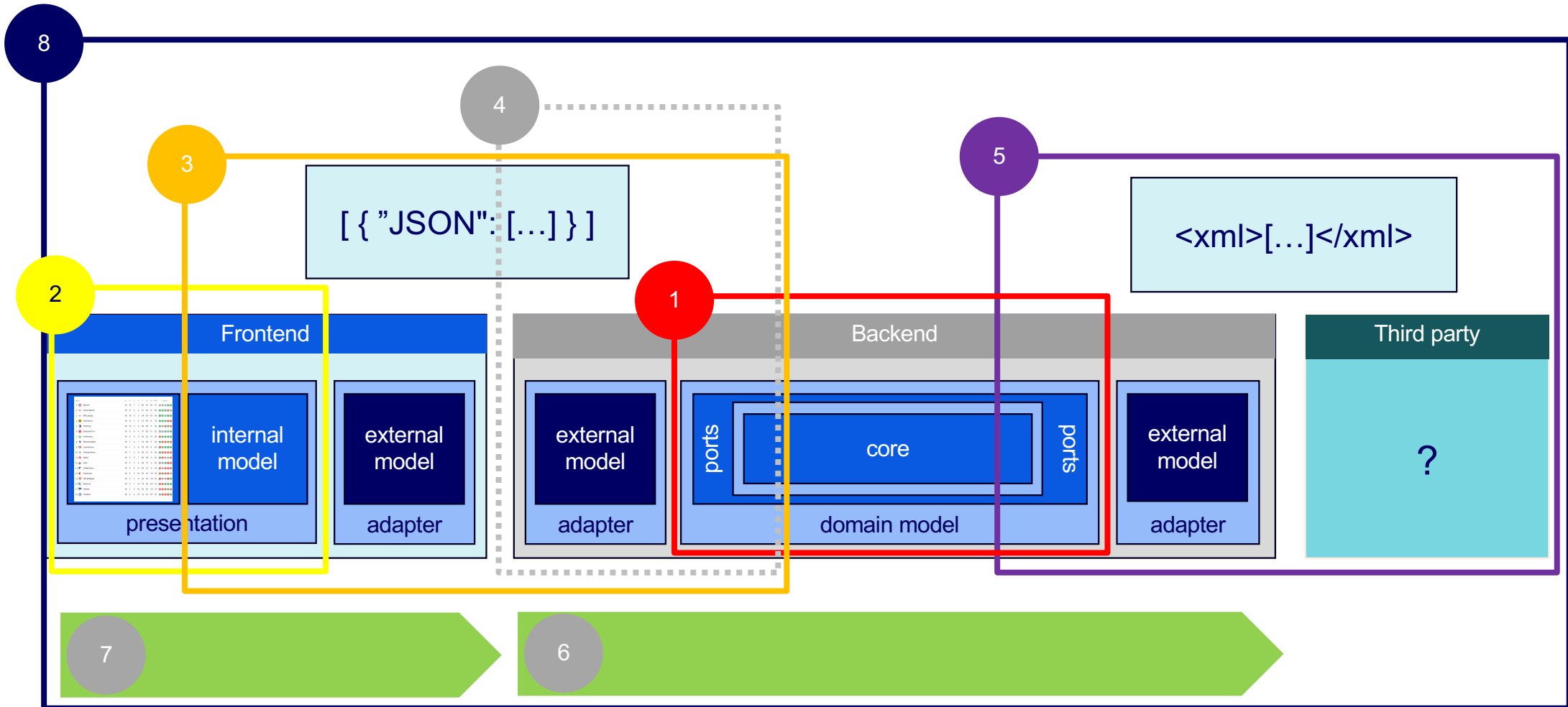
(die Spieltagsdaten stellt Third-Party-Service zur Verfügung)

Verein	Sp	S	U	N	T	GT	TD	Pkte	Letzte 5
1 Bayern	18	10	7	1	52	16	36	37	⚪⚪⚪⚪⚪
2 Union Berlin	18	11	3	4	31	22	9	36	⚪⚪⚪⚪⚪
3 RB Leipzig	18	10	5	3	39	24	15	35	⚪⚪⚪⚪⚪
4 Dortmund	18	11	1	6	33	25	8	34	⚪⚪⚪⚪⚪
5 Freiburg	18	10	4	4	29	25	4	34	⚪⚪⚪⚪⚪
6 Eintracht Fra...	18	9	5	4	37	26	11	32	⚪⚪⚪⚪⚪
7 Wolfsburg	18	8	5	5	36	22	14	29	⚪⚪⚪⚪⚪
8 Mönchenglad...	18	7	4	7	34	29	5	25	⚪⚪⚪⚪⚪
9 Leverkusen	18	7	3	8	30	30	0	24	⚪⚪⚪⚪⚪
10 Werder Brem...	18	7	3	8	29	37	-8	24	⚪⚪⚪⚪⚪
11 Mainz	18	6	5	7	26	29	-3	23	⚪⚪⚪⚪⚪
12 Köln	18	5	7	6	29	31	-2	22	⚪⚪⚪⚪⚪
13 Hoffenheim	18	5	4	9	26	31	-5	19	⚪⚪⚪⚪⚪
14 Augsburg	18	5	3	10	23	33	-10	18	⚪⚪⚪⚪⚪
15 VfB Stuttgart	18	3	7	8	22	32	-10	16	⚪⚪⚪⚪⚪
16 Bochum	18	5	1	12	19	44	-25	16	⚪⚪⚪⚪⚪
17 Hertha	18	3	5	10	20	32	-12	14	⚪⚪⚪⚪⚪
18 Schalke	18	2	4	12	14	41	-27	10	⚪⚪⚪⚪⚪

Komponentenschnitt für eine einfache Testbarkeit



Testschnitt für Tests mit klarer Verantwortung

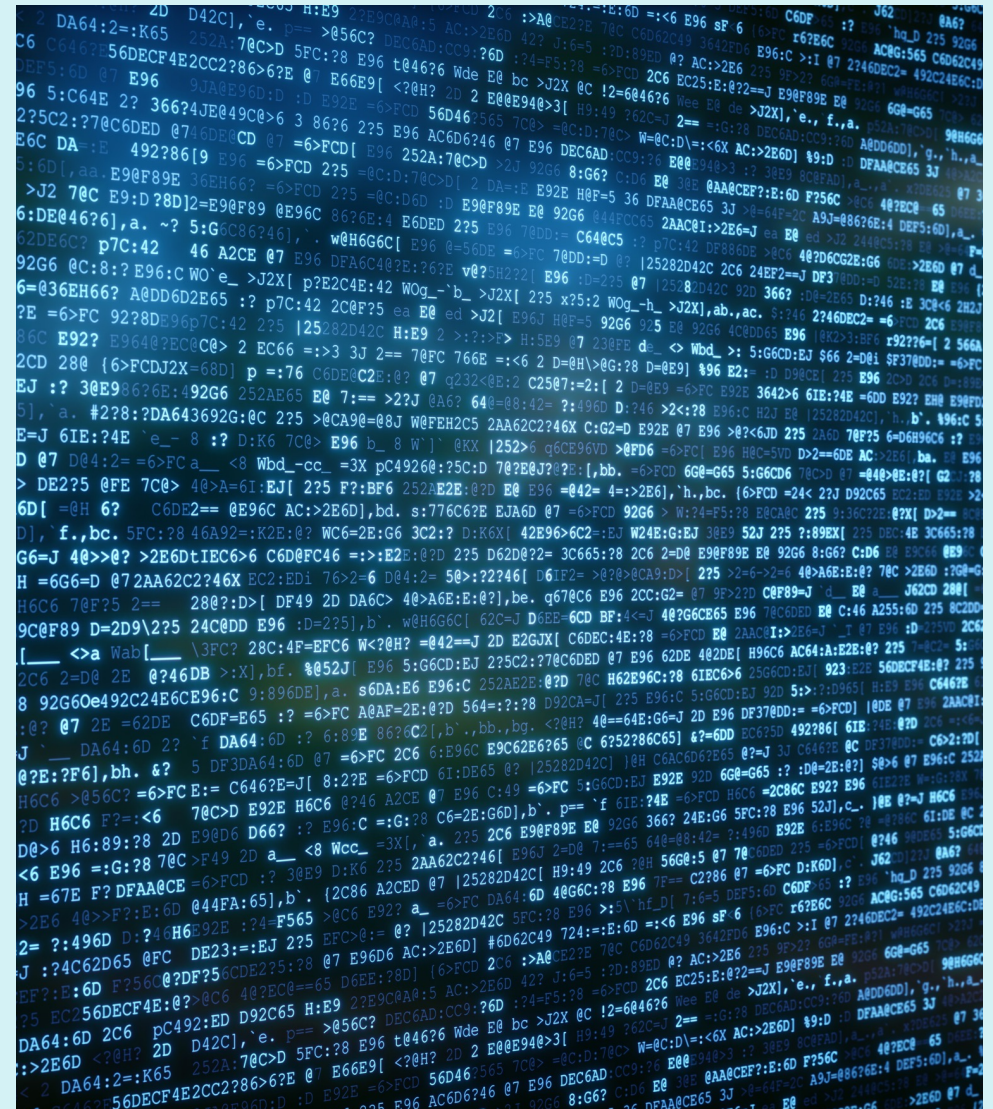


Blick in den Code

...wie sehen jetzt konkret die Tests zu...

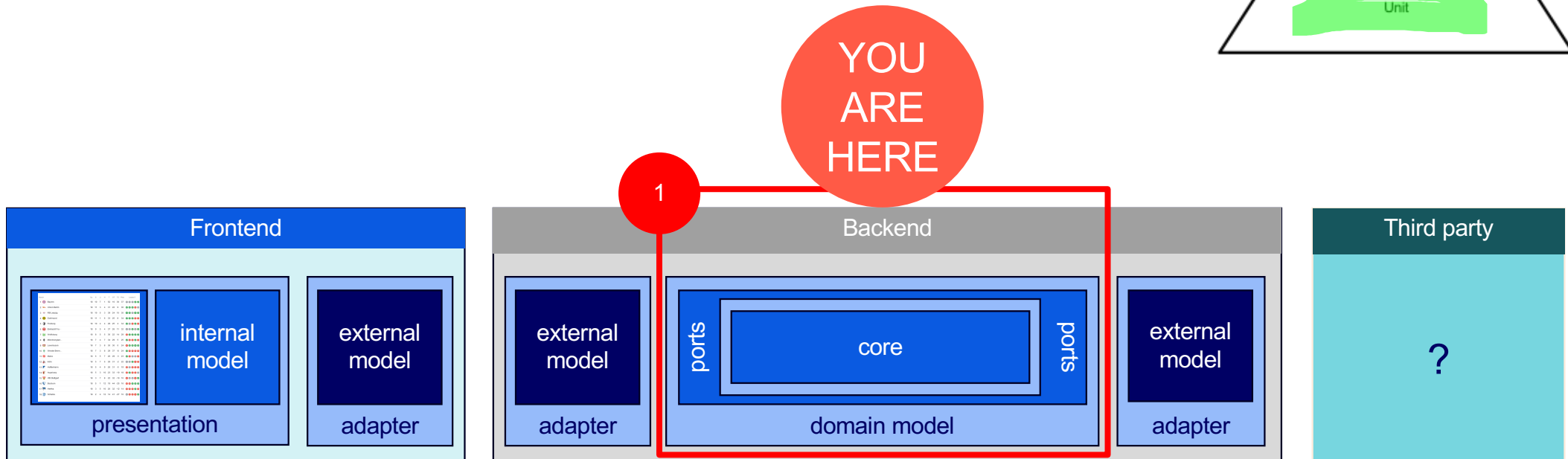
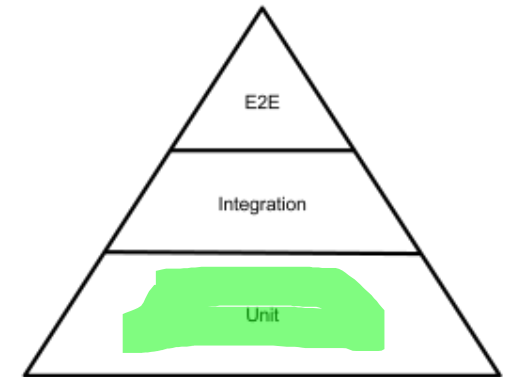
- Fachlichkeit
- Infrastruktur
- Integration
- UI

aus?



Tests ohne Grenzen: Wenn Modulgrenzen sich in Luft auflösen | Peter Fichtner | Öffentlich (C1)

Tests der Fachlichkeit



Tests der Fachlichkeit, Kontext eher fachlich, JUnit BDD-Style

```
@Test
void dieFolgendeMannschaftIstPlatzDrei() {
    gegebenSeienDiePaarungen( //
        paarung("Team 1", "Team 2").endergebnis(1, 0), //
        paarung("Team 2", "Team 1").endergebnis(1, 0), //
        paarung("Team 1", "Team 3").endergebnis(1, 0), //
        paarung("Team 2", "Team 3").endergebnis(1, 0) //
    );
    wennDieTabelleBerechnetWird();
    dannIstDieTabelle( //
        e1 -> {
            assertThat(e1.team().name()).isEqualTo("Team 1");
            assertThat(e1.platz()).isEqualTo(1);
        }, //
        e2 -> {
            assertThat(e2.team().name()).isEqualTo("Team 2");
            assertThat(e2.platz()).isEqualTo(1);
        }, //
        e3 -> {
            assertThat(e3.team().name()).isEqualTo("Team 3");
            assertThat(e3.platz()).isEqualTo(3);
        } //
    );
}
```



<https://github.com/atruvia/bundesliga-tabelle/blob/17d720546b033d1c46e875ea6cfc0d9b163787ea/src/test/java/de/atruvia/ase/samman/buli/domain/DefaultTabelleTest.java#L102>

Tests der Fachlichkeit, on commit

← → ↻ 🏠 <https://github.com/pfichtner/bundesliga-tabelle/runs/21936063316> ☆ 📄 ⬇️ 📄

tests renamed main 42a4581 Re-run all jobs

Java CI with Maven on: push 1

build

Tests Report

GitHub Actions / Tests Report
succeeded 6 hours ago in 1s

Tests Report

tests 28 passed

Report	Passed	Failed	Skipped	Time
target/failsafe-reports/TEST-de.atruvia.ase.samman.buli.infra.adapters.secondary.OpenLigaDbSpieltagRepoT.xml	2			2s
target/failsafe-reports/TEST-de.atruvia.ase.samman.buli.infra.adapters.secondary.OpenLigaDbTeamRepoT.xml	2			2s
target/surefire-reports/TEST-ArchitectureTest.xml	3			1s
target/surefire-reports/TEST-de.atruvia.ase.samman.buli.domain.PaarungTest.xml	2			1ms
target/surefire-reports/TEST-de.atruvia.ase.samman.buli.domain.ports.primary.DefaultTabellenServiceTest.xml	2			452ms
target/surefire-reports/TEST-de.atruvia.ase.samman.buli.domain.TabelleTest.xml	11			102ms
target/surefire-reports/TEST-de.atruvia.ase.samman.buli.infra.adapters.primary.ContractVerificationTest.xml	1			2s
target/surefire-reports/TEST-de.atruvia.ase.samman.buli.infra.adapters.primary.TabellenHttpAdapterTest.xml	2			2s
target/surefire-reports/TEST-de.atruvia.ase.samman.buli.infra.adapters.secondary.OpenLigaDbSpieltagRepoTest.xml	2			20ms
target/surefire-reports/TEST-de.atruvia.ase.samman.buli.infra.adapters.secondary.OpenLigaDbTeamRepoTest.xml	1			1ms

target/failsafe-reports/TEST-de.atruvia.ase.samman.buli.infra.adapters.secondary.OpenLigaDbSpieltagRepoT.xml

2 tests were completed in 2s with 2 passed, 0 failed and 0 skipped.

Test suite	Passed	Failed	Skipped	Time
------------	--------	--------	---------	------

JUnit BDD-Style, allerdings BDD hier nicht ersichtlich

11 tests were completed in 102ms with 11 passed, 0 failed and 0 skipped.

Test suite	Passed	Failed	Skipped	Time
de.atruvia.ase.samman.buli.domain.TabelleTest	11 ✓			102ms

✓ de.atruvia.ase.samman.buli.domain.TabelleTest

- ✓ punktUndTorGleichAberMehrAuswärtsTore
- ✓ zweiMannschaftenZweiSpieleMitToren
- ✓ mannschaftMitMehrPunktenIstWeiterOben
- ✓ zweiMannschaftenKeinSpiel
- ✓ zweiMannschaftenEinSpielKeineTore
- ✓ zweiSpieleTendenz_dieLetztePaarungIstVorneInDerListe
- ✓ nullWappenWerdenNichtUebernommen
- ✓ wappenIstImmerDasDerLetztenPaarung
- ✓ keineSpieleKeineTendenz
- ✓ wennEinWappenInAllenPaarungenNullIstIstEsNull
- ✓ dieFolgendeMannschaftIstPlatzDrei

BDD (Cucumber/Gherkin)

Szenario: Zwei Mannschaften sind punkt- und torgleich

Gegeben sei ein Spielplan

Heim	Gast	Ergebnis
Team 1	Team 2	1:0
Team 2	Team 1	1:0
Team 1	Team 3	1:0
Team 2	Team 3	1:0

Wenn die Tabelle berechnet wird

Dann ist die Tabelle

Team	Platz
Team 1	1
Team 2	1
Team 3	3

<https://github.com/atruvia/bundesliga-tabelle/blob/main/src/test/resources/de/atruvia/ase/samman/buli/domain/Tabelle.feature>

BDD mit Cucumber dann beim Testlauf ersichtlich (on commit)

```
build
succeeded yesterday in 1m 1s
Beta Give feedback Search logs

Build with Maven 47s
1720 Szenario: Zwei Mannschaften sind punkt- und torgleich (Beispiel gezielt nur Team und Platz) # de/atruvia/ase/samman/buli/domain
/Tabelle.feature:20
1721 Gegeben sei ein Spielplan #
de.atruvia.ase.samman.buli.cucumber.StepDefs.der_spielplan(io.cucumber.datatable.DataTable)
1722 | Heim | Gast | Ergebnis |
1723 | Team 1 | Team 2 | 1:0 |
1724 | Team 2 | Team 1 | 1:0 |
1725 | Team 1 | Team 3 | 1:0 |
1726 | Team 2 | Team 3 | 1:0 |
1727 Wenn die Tabelle berechnet wird #
de.atruvia.ase.samman.buli.cucumber.StepDefs.die_tabelle_berechnet_wird()
1728 Dann ist die Tabelle #
de.atruvia.ase.samman.buli.cucumber.StepDefs.ist_die_tabelle(io.cucumber.datatable.DataTable)
1729 | Team | Platz |
1730 | Team 1 | 1 |
1731 | Team 2 | 1 |
1732 | Team 3 | 3 |
1733 [INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.689 s -- in de.atruvia.ase.samman.buli.cucumber.RunCucumberTest
1734 [INFO] Running de.atruvia.ase.samman.buli.ArchitectureTest
1735 2024-04-03T07:11:57.274Z INFO 1817 --- [ main] com.tngtech.archunit.core.PluginLoader : Detected Java version 17.0.10
1736 [INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.500 s -- in de.atruvia.ase.samman.buli.ArchitectureTest
```

<https://github.com/atruvia/bundesliga-tabelle/actions/runs/11120057338/job/30896328864>

BDD mit Cucumber dann beim Testlauf ersichtlich, HTML (on commit)

Feature	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Tabelle erstellen	6	0	0	0	0	6	2	0	2	0.120	Passed



Funktionalität Tabelle erstellen

Szenario Zwei Mannschaften sind punkt- und torgleich (Beispiel volle Tabelle) 0.120

Steps >

Szenario Zwei Mannschaften sind punkt- und torgleich (Beispiel gezielt nur Team und Platz) 0.000

Steps v

Gegeben sei ein Spielplan 0.000

Heim	Gast	Ergebnis
Team 1	Team 2	1:0
Team 2	Team 1	1:0
Team 1	Team 3	1:0
Team 2	Team 3	1:0

Wenn die Tabelle berechnet wird 0.000

Dann ist die Tabelle 0.000

Team	Platz
Team 1	1
Team 2	1
Team 3	3













<https://github.com/pfichtner/bundesliga-tabelle/blob/main/src/test/resources/de/atruvia/ase/samman/buli/domain/Tabelle.feature>

Approvals

Filesystem-based Repository

```
@Test
void tabelleBl12023Spieltag27_gamesRunning_goalsAndFinalResultsAreCorrect() {
    TabellenService sut = new DefaultTabellenService(spieltagFsRepo());
    verifyTabelle(sut.erstelleTabelle("bl1", "2023-games-running-correct-final-result"));
}

private static void verifyTabelle(Tabelle tabelle) {
    var headerNames = attributes.keySet().toArray(String[]::new);
    var headerRow = Stream.of(markdownRow(headerNames));
    var separatorRow = Stream.of(markdownSeparator(headerNames));
    var contentRows = tabelle.entries().stream().map(DefaultTabellenServiceTest::print);
    verify(concat(headerRow, separatorRow, contentRows).collect(joining("\n")), markdown());
}
```

Platz	Logo	Verein	Sp	S	U	N	T	GT	TD	Pkte	Letzte 5	Spiel
1		Bayer Leverkusen	27	23	4	0	68	19	49	73	✓✓✓✓✓	
2		FC Bayern München	27	19	3	5	78	33	45	60	✗✓✓-✓	
3		VfB Stuttgart	27	18	3	6	63	34	29	57	-✓✓✓-	
4		Borussia Dortmund	27	15	8	4	55	32	23	53	✓✓✓✓✗	
5		RB Leipzig	27	15	5	7	60	32	28	50	-✓✓✓✗	
6		Eintracht Frankfurt	28	10	12	6	43	36	7	42	-✗✓✓-	1:1 (Werder Bremen)
7		FC Augsburg	27	9	9	9	44	43	1	36	-✓✓✓✓	
8		SC Freiburg	27	10	6	11	39	48	-9	36	✓✗✓-✗	
9		TSG 1899 Hoffenheim	27	9	6	12	45	52	-7	33	✗✗✗✓✓	
10		Werder Bremen	28	8	7	13	36	44	-8	31	✗✗✗✗-	1:1 (Eintracht Frankfurt)
11		1. FC Heidenheim 1846	27	7	9	11	38	47	-9	30	--✗✗-	
12		1. FC Union Berlin	27	9	5	13	35	42	-7	30	-✓✓✓-	

<https://github.com/atruvia/bundesliga-tabelle/blob/main/src/test/java/de/atruvia/ase/samman/buli/domain/ports/primary/DefaultTabellenServiceTest.java>

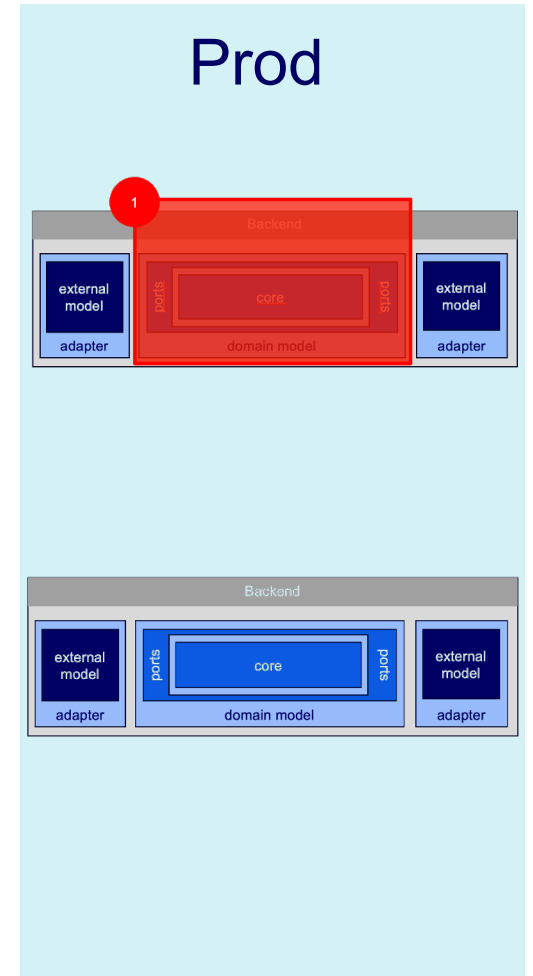
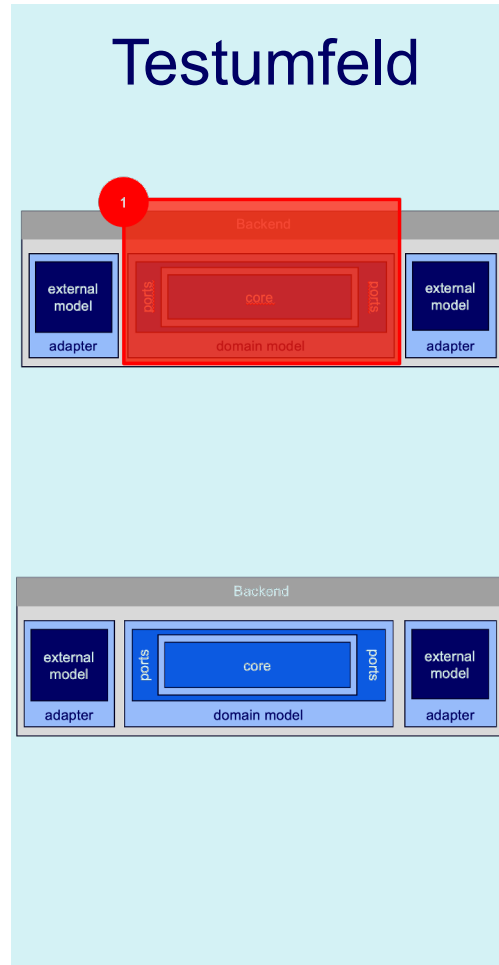
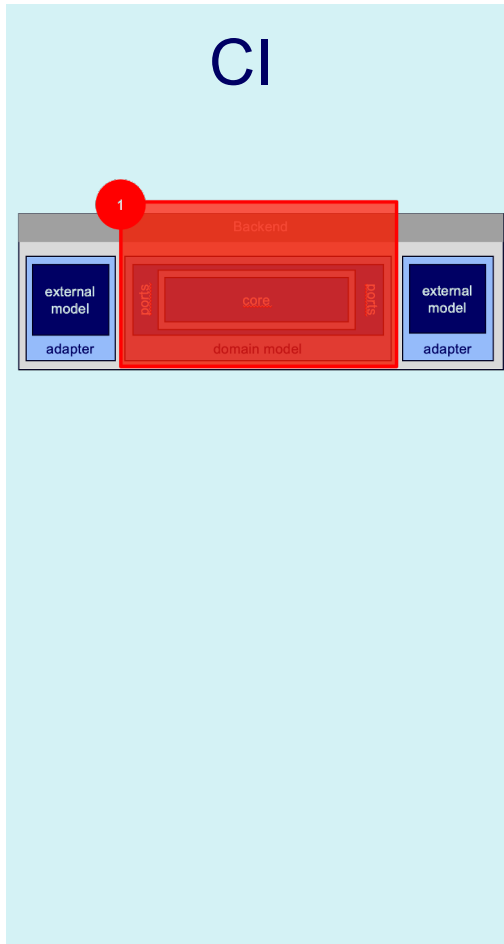
Tests der “Fachlichkeit”, Kontext eher technisch (Detailtest)

```
@Test
void testCheckUnique_allUnique() {
    assertThat(checkUnique("a", "a", "a")).isEqualTo("a");
}
```

```
@Test
void testCheckUnique_notAllUnique() {
    assertThatThrownBy(() -> checkUnique("a", "A", "a")).hasMessageContaining("a, A, a");
}
```

<https://github.com/atruvia/bundesliga-tabelle/blob/main/src/test/java/de/atruvia/ase/samman/buli/util/MergerTest.java>

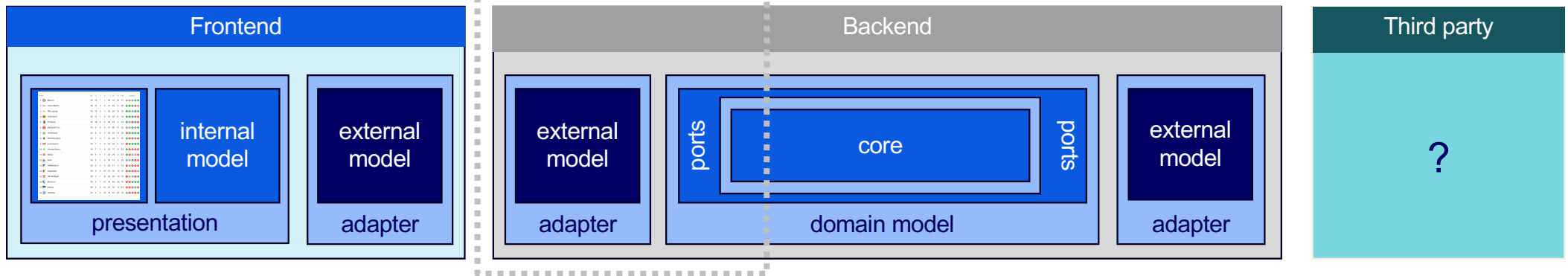
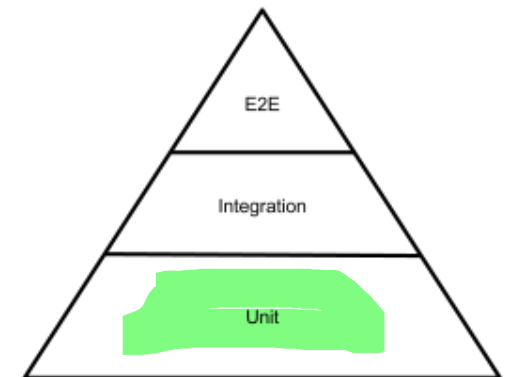
Das Artefakt wandert durch die Umgebungen, warum eigentlich?



Tests des Eingangsadapters

4

YOU ARE HERE



Tests des Eingangsadapters

@Test

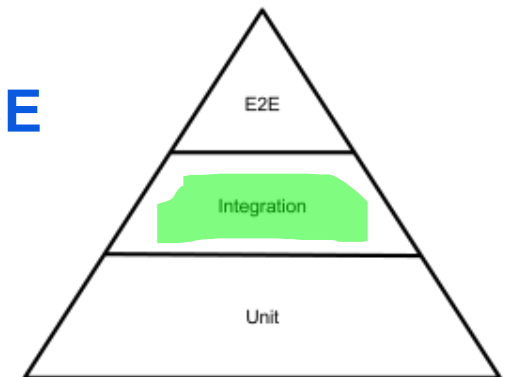
```
void shouldReturnDefaultMessage() throws Exception {
    String league = "bl1";
    String season = "2022";

    TabellenPlatz platz1 = platzWithBase(10, platzWith(SIEG, UNENTSCHIEDEN, NIEDERLAGE).toBuilder());
    TabellenPlatz platz2 = platzWithBase(20, platzWith().toBuilder());
    when(tabelleService.erstelleTabelle(league, season)).thenReturn(List.of(platz1, platz2));

    this.mockMvc.perform(get("/tabelle/" + league + "/" + season)) //
        .andDo(print()) //
        .andExpect(status().isOk()) //
        .andExpect(jsonPath("$. [0].wappen", is(platz1.getWappen().toASCIIString()))) //
        .andExpect(jsonPath("$. [0].team", is(platz1.getTeam()))) //
        .andExpect(jsonPath("$. [0].spiele", is(platz1.getSpiele()))) //
        .andExpect(jsonPath("$. [0].siege", is(platz1.getAnzahlSiege()))) //
        .andExpect(jsonPath("$. [0].unentschieden", is(platz1.getAnzahlUnentschieden()))) //
        .andExpect(jsonPath("$. [0].niederlagen", is(platz1.getAnzahlNiederlagen()))) //
        .andExpect(jsonPath("$. [0].tore", is(platz1.getTore()))) //
        .andExpect(jsonPath("$. [0].gegentore", is(platz1.getGegentore()))) //
        .andExpect(jsonPath("$. [0].tordifferenz", is(platz1.getTorDifferenz()))) //
        .andExpect(jsonPath("$. [0].punkte", is(platz1.getPunkte()))) //
        .andExpect(jsonPath("$. [0].letzte5", is("NUS--"))) //
        //
        .andExpect(jsonPath("$. [1].wappen", is(platz2.getWappen().toASCIIString()))) //
        .andExpect(jsonPath("$. [1].team", is(platz2.getTeam()))) //
        .andExpect(jsonPath("$. [1].spiele", is(platz2.getSpiele()))) //
        .andExpect(jsonPath("$. [1].siege", is(platz2.getAnzahlSiege()))) //
}
```

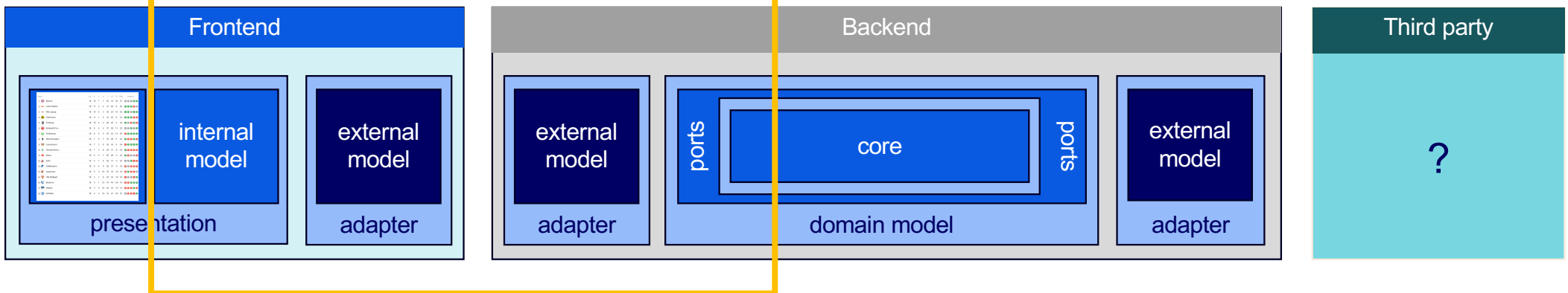
<https://github.com/atruvia/bundesliga-tabelle/blob/8e748055d24595c39855030e6d1731c79ea9ecba/src/test/java/de/atruvia/ase/samman/bu/li/infra/adapters/primary/TabellenHttpAdapterTest.java#L64>

Tests des Eingangsadapters: Detour → Integration FE/BE



YOU
ARE
HERE

3



CDCT to the rescue! Frontend-Seite

```
it('should provide bundesliga data', (done) => {
  provider.addInteraction(new Interaction()
    .given('matchday #3 team has won on matchday #1, draw on matchday #2 and loss on day #3')
    .uponReceiving('a request to get the bundesliga standings')
    .withRequest({
      method: 'GET',
      path: '/tabelle/bl1/2023',
      headers: {Accept: 'application/json'}
    })
    .willRespondWith({
      status: 200,
      headers: {'Content-Type': 'application/json'},
      body: [
        {
          "platz": integer(12),
          "team": string("any team name"),
          "spiele": integer(23),
          "punkte": integer(34),
          "tore": integer(45),
          "gegentore": integer(56),
          "tordifferenz": integer(67),
          "siege": integer(78),
          "unentschieden": integer(89),
          "niederlagen": integer(90),
          "letzte5": term({
            generate: 'NUSxx',
            matcher: 'NUS|^NUS[^NUS]{2}'
          })
        },
        {
          "nplatz": integer().
```

erstellt/
schreibt



```
..interactions: [
  {
    "description": "a request to get the bundesliga standings",
    "providerState": "matchday #3 team has won on matchday #1, draw on matchday #2 and loss on day #3",
    "request": {
      "headers": {
        "Accept": "application/json"
      },
      "method": "GET",
      "path": "/tabelle/bl1/2023"
    },
    "response": {
      "body": [
        {
          "gegentore": 3,
          "letzte5": "NUSxx",
          "niederlagen": 1,
          "platz": 1,
          "punkte": 4,
          "siege": 1,
          "spiele": 3,
          "team": "anyTeamName",
          "tordifferenz": 0,
          "tore": 3,
          "unentschieden": 1
        },
        {
          "gegentore": 3,
          "letzte5": "NUSxx",
          "niederlagen": 1,
          "platz": 1,
          "punkte": 4,
          "siege": 1,
          "spiele": 3,
          "team": "anyTeamName",
          "tordifferenz": 0,
          "tore": 3,
          "unentschieden": 1
        }
      ]
    }
  }
],
..matchingRules: {
  "$.body[0].gegentore": {
    "match": "type"
  },
  "$.body[0].letzte5": {
    "match": "regex",
    "regex": "NUS|NUS.{2}"
  },
  "$.body[0].platz": {
    "match": "type"
  },
  "$.body[0].tordifferenz": {
```

<https://github.com/tobibechtold/bundesliga-tabelle-ui/blob/526b71b097cceca84c953791c4e74a93129c6614/src/app/services/bundesliga-table.service.spec.pact.ts#L30>

CDCT to the rescue! Backend-Seite

```
@Provider("BundesligaBackend")
@PactFolder("pacts")
@SpringBootTest(classes = Main.class, webEnvironment = RANDOM_PORT)
@ContractTest
class ContractVerificationTest {

    @LocalServerPort
    int port;

    @MockBean
    SpieltagRepo spieltagRepoMock;

    @BeforeEach
    void setup(PactVerificationContext context) {
        context.setTarget(new HttpTestTarget("localhost", port));
    }

    @TestTemplate
    @ExtendWith(PactVerificationInvocationContextProvider.class)
    void verifyContracts(PactVerificationContext context) {
        context.verifyInteraction();
    }

    @State("matchday #3 team has won on matchday #1, draw on matchday #2 and loss on day #3")
    void matchdayThreeWinDrawLoss() throws Exception {
        when(spieltagRepoMock.lade(anyString(), anyString()))
            .thenReturn(paarungen("anyTeamName", SIEG, UNENTSCHIEDEN, NIEDERLAGE));
    }
}
```

liest/
verarbeitet



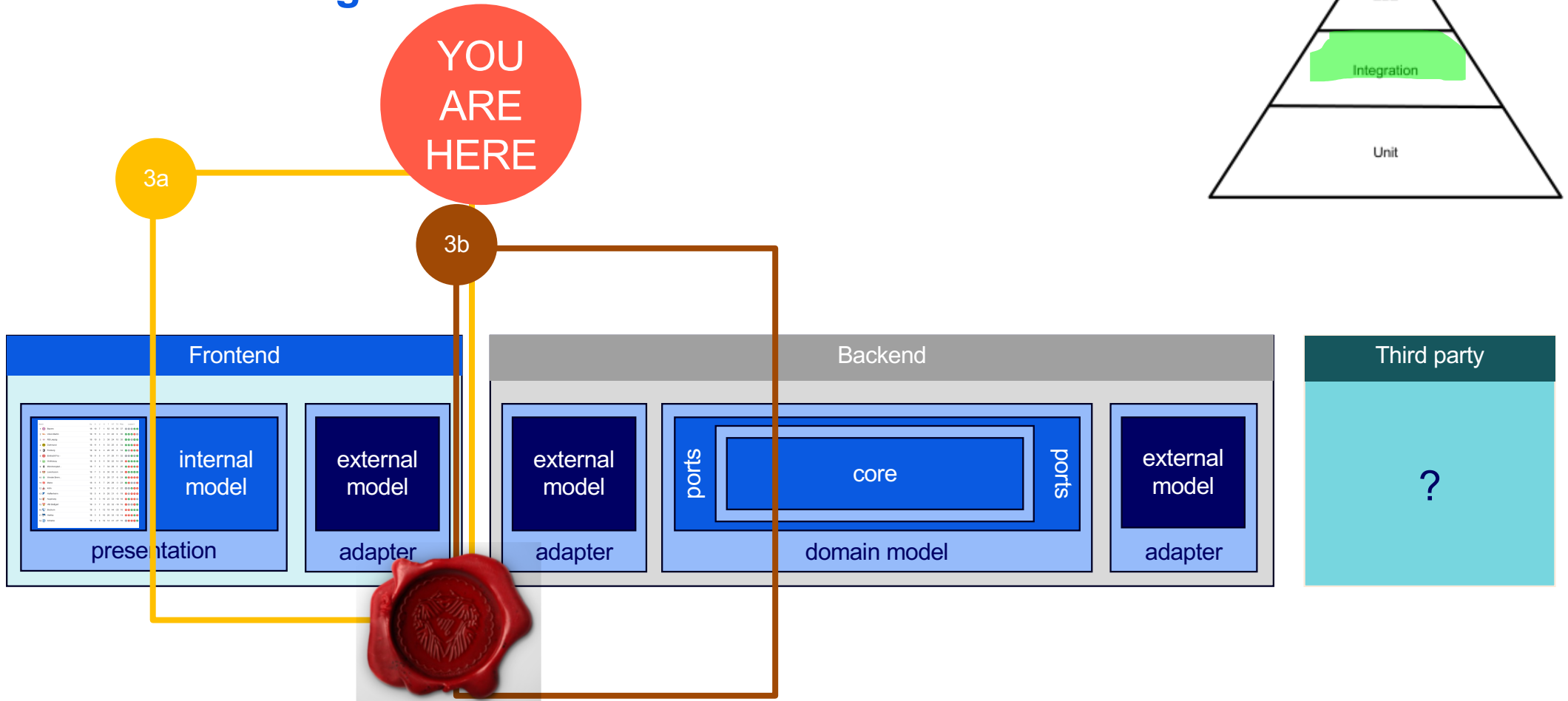
```
"interactions": [
  {
    "description": "a request to get the bundesliga standings",
    "providerState": "matchday #3 team has won on matchday #1, draw on matchday #2 and loss on day #3",
    "request": {
      "headers": {
        "Accept": "application/json"
      },
      "method": "GET",
      "path": "/tabelle/bl1/2023"
    },
    "response": {
      "body": [
        {
          "gegentore": 3,
          "letzte5": "NUSxx",
          "niederlagen": 1,
          "platz": 1,
          "punkte": 4,
          "siege": 1,
          "spiele": 3,
          "team": "anyTeamName",
          "tordifferenz": 0,
          "tore": 3,
          "unentschieden": 1
        }
      ]
    }
  },
  {
    "matchingRules": {
      "$.body[0].gegentore": {
        "match": "type"
      },
      "$.body[0].letzte5": {
        "match": "regex",
        "regex": "NUS|NUS.{2}"
      },
      "$.body[0].platz": {
        "match": "type"
      },
      "$.body[0].tordifferenz": {

```

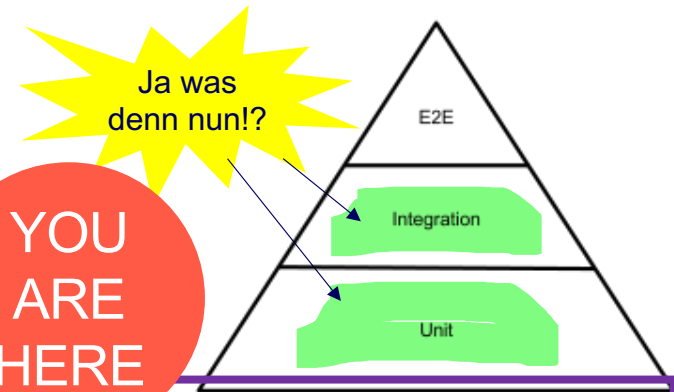


<https://github.com/atruvia/bundesliga-tabelle/blob/8653baf61841e1001675dde47fac453b7b02bae1/src/test/java/de/atruvia/ase/samman/buli/infra/adapters/primary/ContractVerificationTest.java#L31>

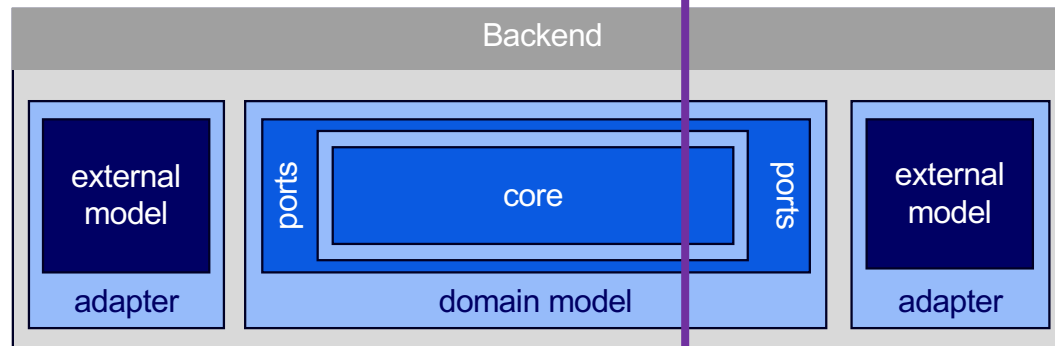
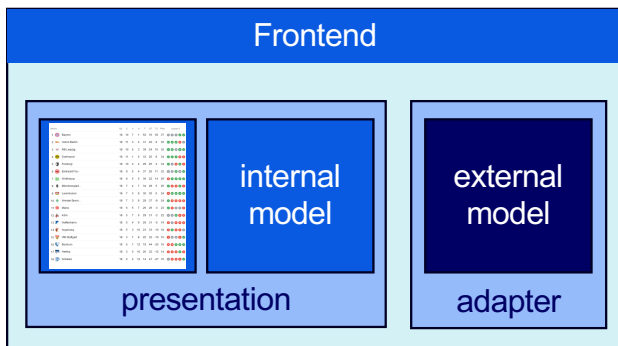
Daher CDCT eigentlich so...



Tests des Ausgangsadapters



5



Tests des Ausgangsadapters

```
@Test
void canRetrieveDataOf2022() throws Exception {
    List<Paarung> paarungen = repo().lade("bl1", "2022");
    Paarung expected0 = Paarung.builder() //
        .heim(new Paarung.Entry(teamFrankfurt, wappenFrankfurt)) //
        .gast(new Paarung.Entry(teamMuenchen, wappenMuenchen)) //
        .build().withErgebnis(1, 6);
    assertThat(paarungen).hasSize(306).element(0).isEqualTo(expected0);
}

@Test
void canRetrieveDataOf2023() throws Exception {
    // 2022: "endergebnis" was first element of array, 2023 it was last -> filter
    // "resultTypeID" = 2 for now
    List<Paarung> paarungen = repo().lade("bl1", "2023");
    Paarung expected0 = Paarung.builder() //
        .heim(new Paarung.Entry(teamBremen, wappenBremen)) //
        .gast(new Paarung.Entry(teamMuenchen, wappenMuenchen)) //
        .build().withErgebnis(0, 4);
    assertThat(paarungen).hasSize(9).element(0).isEqualTo(expected0);
}
```



<https://github.com/atruvia/bundesliga-tabelle/blob/main/src/test/java/de/atruvia/ase/samman/buli/infra/adapters/secondary/OpenLigaDbSpieltagRepoT.java>

Unit

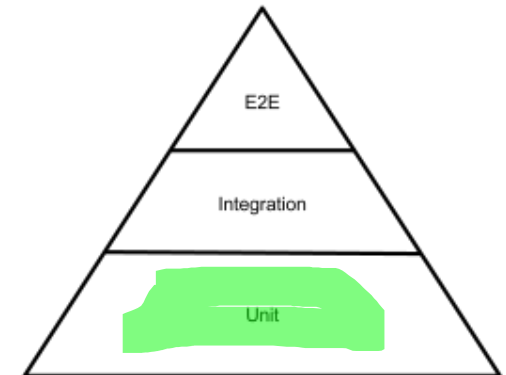
```
OpenLigaDbSpieltagRepo repo() {
    return spieltagFsRepo();
}

public static OpenLigaDbSpieltagRepo spieltagFsRepo() {
    return new OpenLigaDbSpieltagRepo() {
        @Override
        protected String readJson(String league, String season)
            return readString(path("getmatchdata", league, season));
    };
}
```

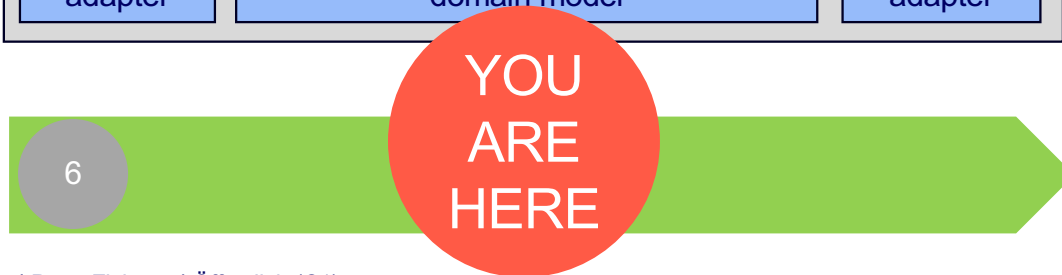
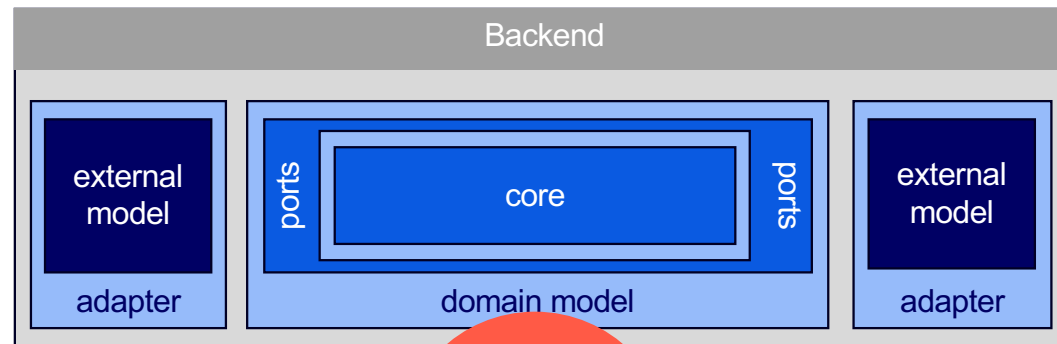
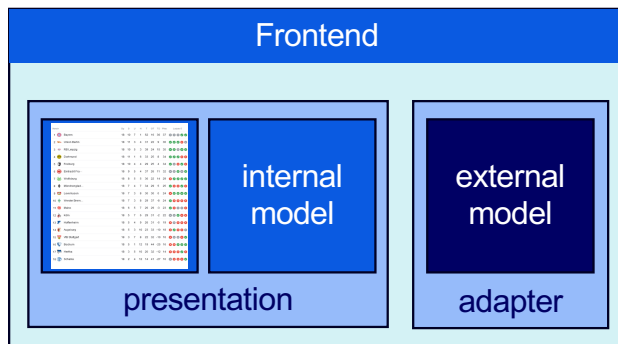
Integration

```
OpenLigaDbSpieltagRepo repo() {
    return new OpenLigaDbSpieltagRepo();
}
```

Smoke-Test des Backends



```
<xml>[...]</xml>
```



6

Smoke-Test des Backends, poor man's

```
# Function to check if the application is up
wait_for_application() {
    local port=$1
    local max_attempts=30
    local attempt=0

    until [ $attempt -ge $max_attempts ]; do
        # Make a generic HTTP request to check if the application is reachable
        if curl -s -I "http://localhost:${port}" | grep -q "HTTP/1.1 2\|3"; then
            echo "Application is reachable on port $port"
            return 0
        fi

        echo "Waiting for the application to start on port $port..."
        sleep 5
        attempt=$((attempt+1))
    done

    echo "Application did not start within the expected time."
    exit 1
}
```

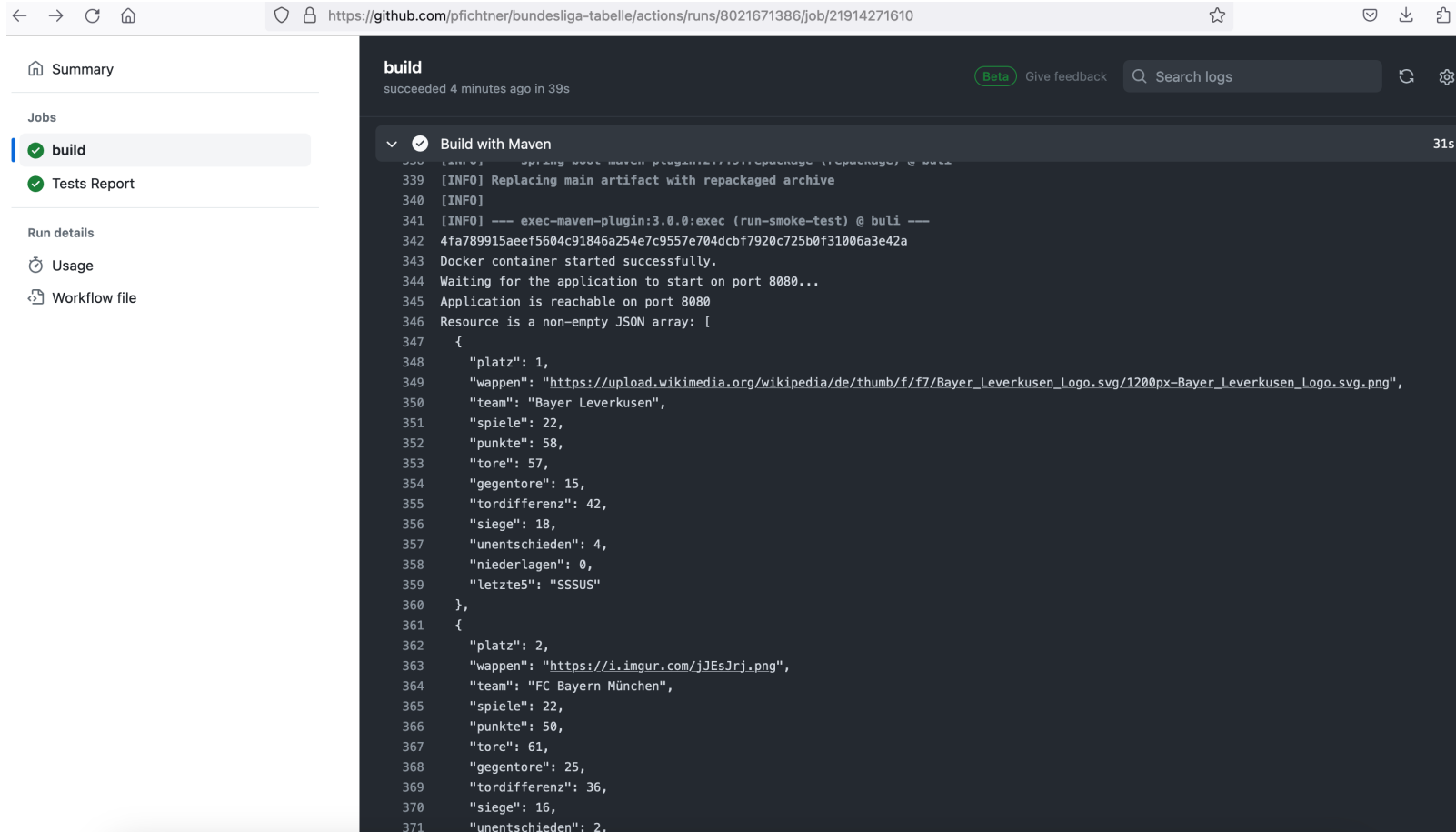
```
# Function to check if the resource is a non-empty JSON array
check_resource_non_empty_json_array() {
    local port=$1
    local url="http://localhost:${port}${RESOURCE_PATH}"

    # Make an HTTP request and capture the response
    local response=$(curl -s -w "%{http_code}" $url)

    local http_status="${response:(-3)}"
    local body="${response::-3}"

    # Check if the HTTP status is 200 and the response is a non-empty JSON array
    if [ "$http_status" -eq 200 ] && [[ $body == \[*\] && $body != "[]" ]]; then
        echo "Resource is a non-empty JSON array: $body"
        return 0
    else
        echo "Error: Unexpected response received for the resource: $body    ..."
        exit 1
    fi
}
```

Smoke-Test des Backends, poor man's (auch on commit)



Summary

Jobs

- build
- Tests Report

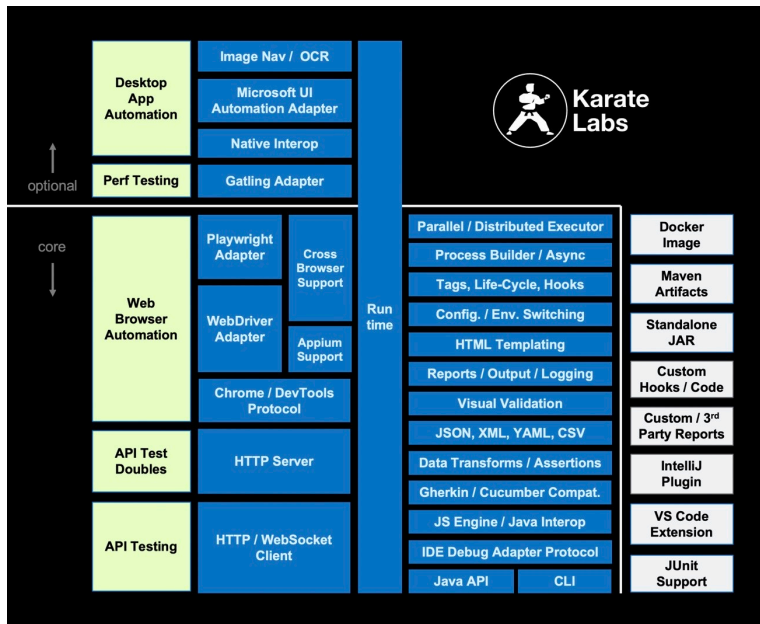
Run details

- Usage
- Workflow file

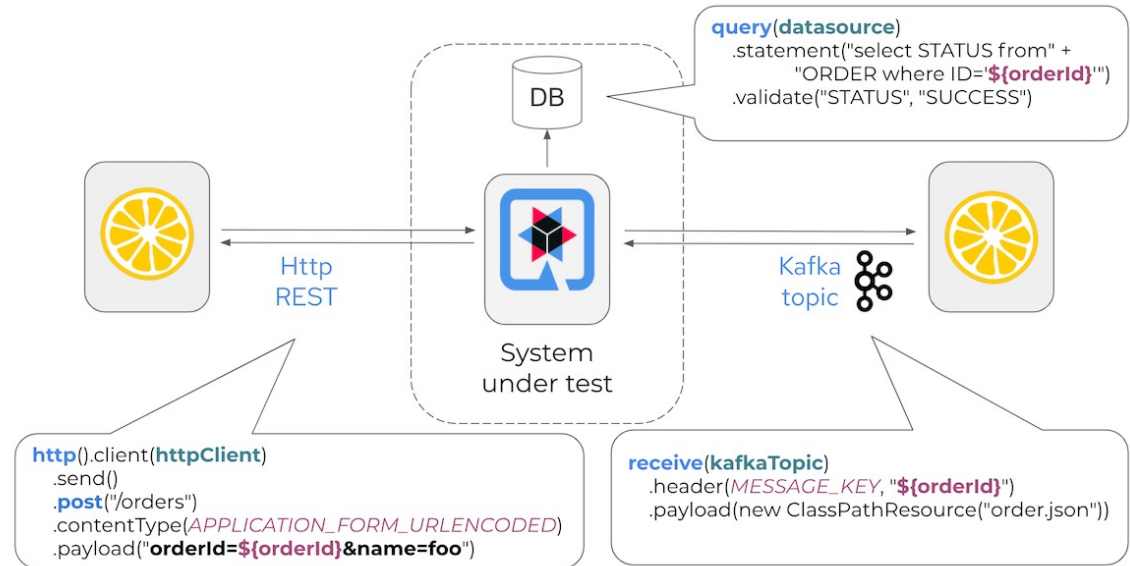
```
build
succeeded 4 minutes ago in 39s
Beta Give feedback Search logs

Build with Maven 31s
330 [INFO] Replacing main artifact with repackaged archive
339 [INFO] Replacing main artifact with repackaged archive
340 [INFO]
341 [INFO] --- exec-maven-plugin:3.0.0:exec (run-smoke-test) @ buli ---
342 4fa789915aeef5604c91846a254e7c9557e704dcbf7920c725b0f31006a3e42a
343 Docker container started successfully.
344 Waiting for the application to start on port 8080...
345 Application is reachable on port 8080
346 Resource is a non-empty JSON array: [
347   {
348     "platz": 1,
349     "wappen": "https://upload.wikimedia.org/wikipedia/de/thumb/f/f7/Bayer_Leverkusen_Logo.svg/1200px-Bayer_Leverkusen_Logo.svg.png",
350     "team": "Bayer Leverkusen",
351     "spiele": 22,
352     "punkte": 58,
353     "tore": 57,
354     "gegentore": 15,
355     "tordifferenz": 42,
356     "siege": 18,
357     "unentschieden": 4,
358     "niederlagen": 0,
359     "letzte5": "SSSUS"
360   },
361   {
362     "platz": 2,
363     "wappen": "https://i.imgur.com/jJEsJrj.png",
364     "team": "FC Bayern München",
365     "spiele": 22,
366     "punkte": 50,
367     "tore": 61,
368     "gegentore": 25,
369     "tordifferenz": 36,
370     "siege": 16,
371     "unentschieden": 2,
```

Smoke-Test des Backends, Tools/Frameworks

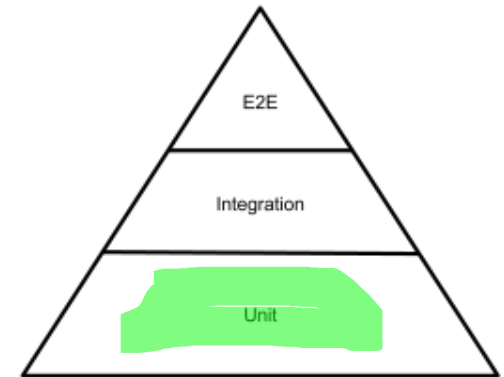


Karate
<https://github.com/karatelabs/karate>



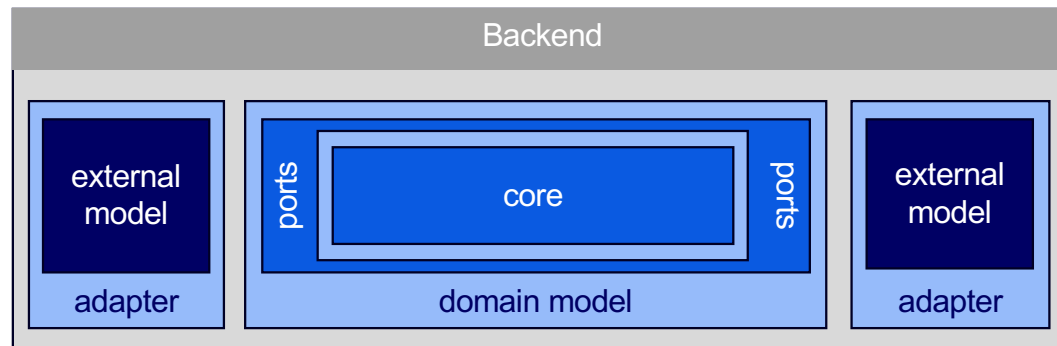
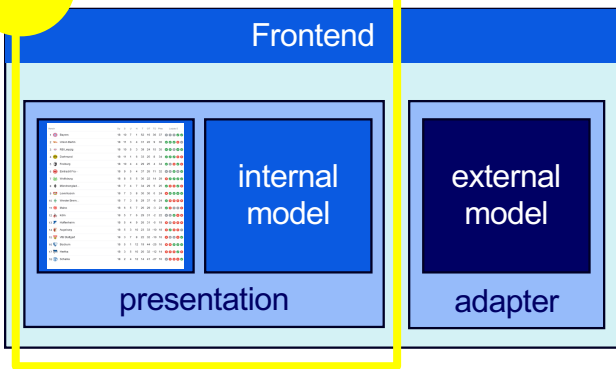
Citrus
<https://citrusframework.org/>

UI-Tests



YOU
ARE
HERE

2



UI-Tests (cypress)

```
let tableService = {
  getTableFromServer(liga: string, saison: string): Observable<Team[]> {
    return of([
      {
        "platz": 12,
        "wappen": "https://i.imgur.com/jJEsJrj.png",
        "team": "Ein recht langer Teamname",
        "spiele": 23,
        "punkte": 34,
        "tore": 45,
        "gegentore": 56,
        "tordifferenz": 67,
        "siege": 78,
        "unentschieden": 89,
        "niederlagen": 90,
        "letzte5": [
          "../assets/sieg.svg",
          "../assets/niederlage.svg",
          "../assets/niederlage.svg",
          "../assets/niederlage.svg",
          "../assets/niederlage.svg"
        ]
      }
    ]);
  }
};
```

```
describe('AppComponent.cy.ts', () => {
  it('mounts', () => {
    cy.mount(AppComponent, {
      imports: [HttpClientTestingModule]
    });
  });

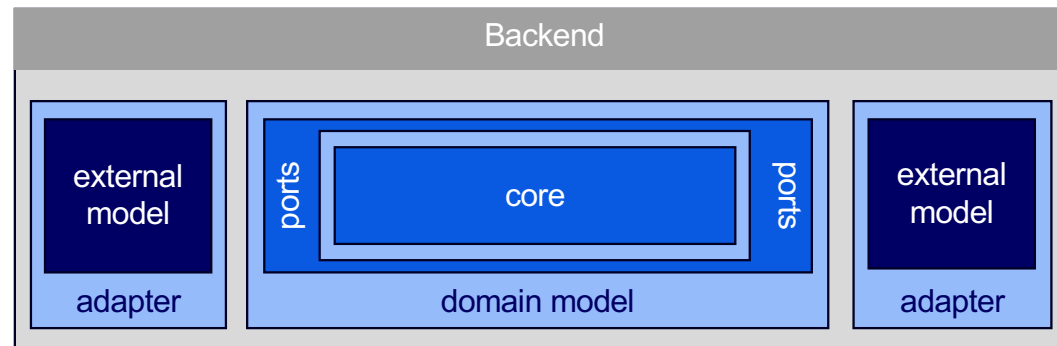
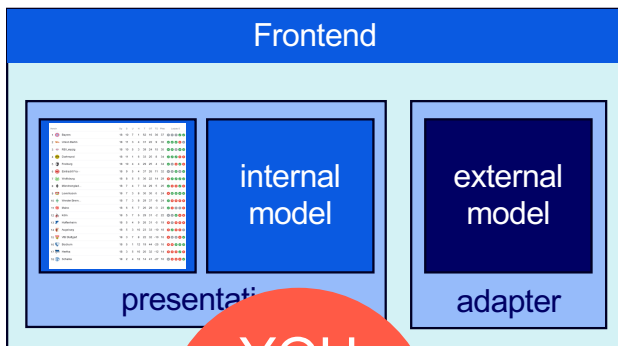
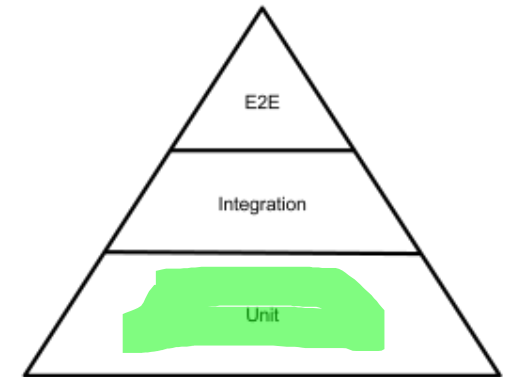
  it('displays one team', () => {
    cy.mount(AppComponent, {
      imports: [HttpClientTestingModule],
      providers: [{provide: BundesligaTableService, useValue: tableService}],
    });
    cy.get('[data-cy="platz"]').should('have.text', '12.');
```



```
    // <-- dot gets appended :)
    cy.get('[data-cy="wappen"]').should('have.attr', 'src', 'https://i.imgur.com/jJEsJrj.png');
    cy.get('[data-cy="teamname"]').should('have.text', 'Ein recht langer Teamname');
    cy.get('[data-cy="spiele"]').should('have.text', '23');
    cy.get('[data-cy="punkte"]').should('have.text', '34');
    cy.get('[data-cy="tore"]').should('have.text', '45');
    cy.get('[data-cy="gegentore"]').should('have.text', '56');
    cy.get('[data-cy="tordifferenz"]').should('have.text', '67');
    cy.get('[data-cy="siege"]').should('have.text', '78');
    cy.get('[data-cy="unentschieden"]').should('have.text', '89');
    cy.get('[data-cy="niederlagen"]').should('have.text', '90');
    // cy.get('[data-cy="letzte5"]').should('have.text', '');
  });
});
```

<https://github.com/tobibechtold/bundesliga-tabelle-ui/blob/main/cypress/AppComponent.cy.ts>

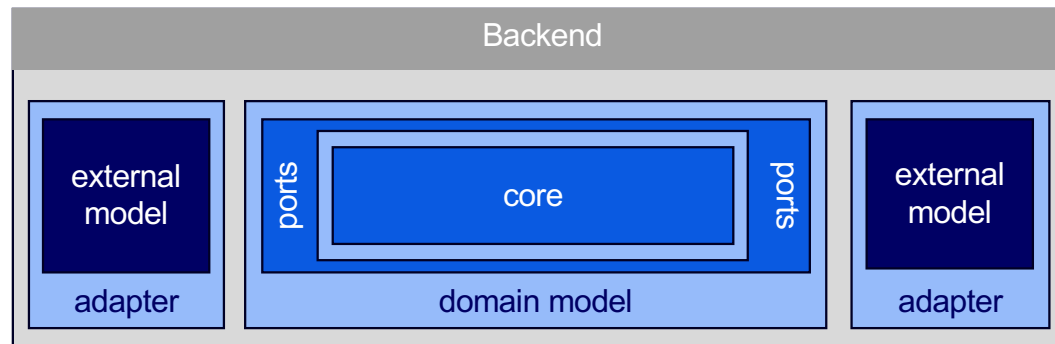
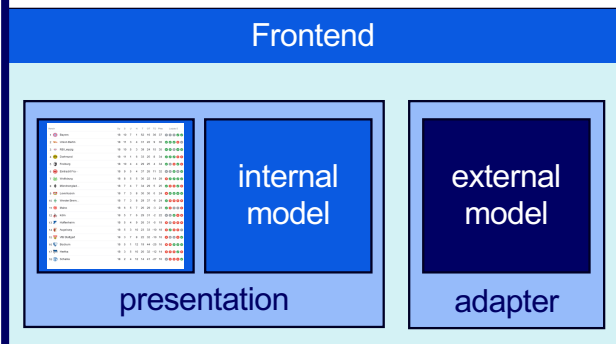
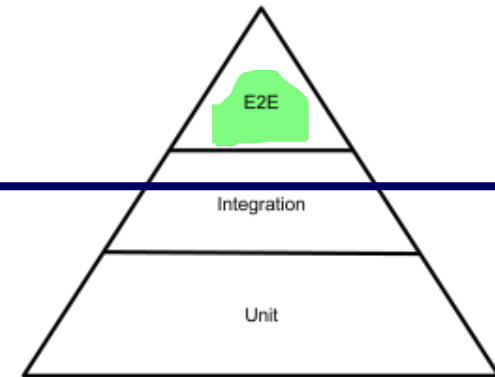
Smoke-Test des Frontends



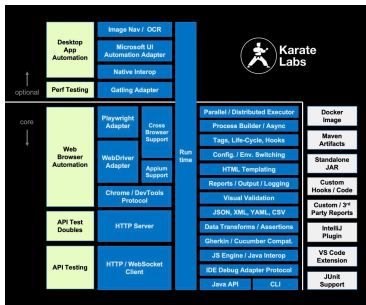
Gesamt-Integrations-/Ende-zu-Ende Tests

8

YOU
ARE
HERE



Gesamt-Integrations-/Ende-zu-Ende Tests



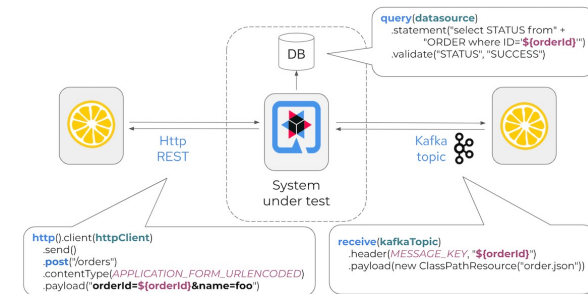
Karate

<https://github.com/karatelabs/karate>



Selenium

<https://www.selenium.dev/>



Citrus

<https://citrusframework.org/>



Playwright

<https://playwright.dev/>

Fazit

- Software schneiden: nicht nur Produktivcode, sondern auch die Tests
- Möglichst wenige Tests sollten vom Verhalten anderer Systeme abhängen
- Verhalten dort testen, wo es implementiert ist
 - Zielgerichtete Tests...
 - werden häufiger ausgeführt (auch/weil viele davon sehr schnell)
 - geben schnelleres und präziseres Feedback
 - helfen Fehler zu vermeiden
 - verringern die Reaktionszeit (auch) bei Fehlern, die erst in Produktion bemerkt werden
 - Viele Tests können...
 - ganz vom integrativen Charakter befreit werden
 - und leichtgewichtiger umgesetzt werden

→ Shift Left



Vielen Dank!

