

Softwaretechnik 2 (ST2)

Informatik Bachelor, SoSe 2025

Prof. Dr.-Ing. Stefan Bente

Workshop M3:

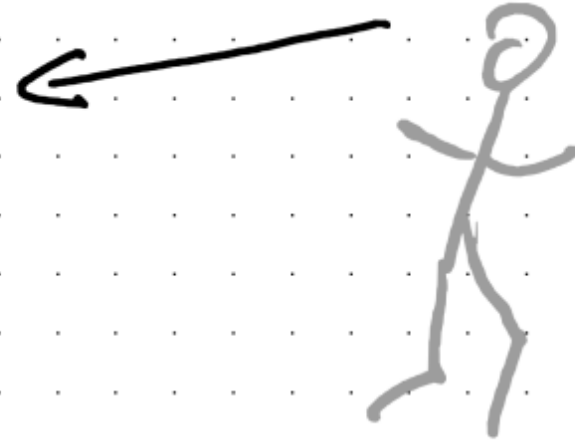
Tests als Programmierwerkzeug

Technology
Arts Sciences
TH Köln

Acceptance Test

- Acceptanz test

Kunde



Typen von automatisierten Tests

■ Unit Tests

- Strikt lokal, „1 Klasse“
- kein Kontext, keine Infrastruktur

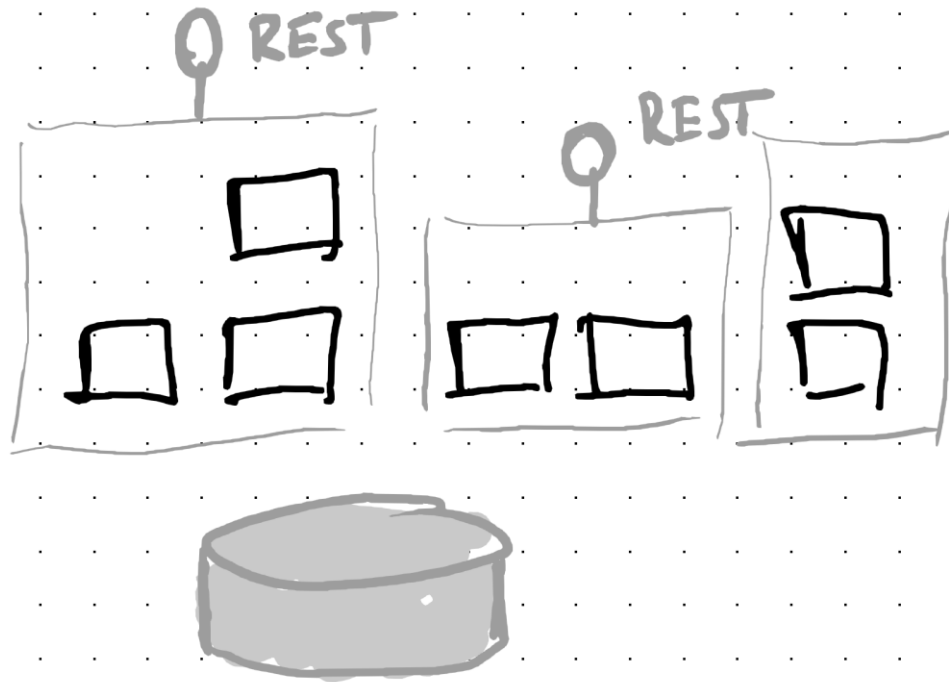
■ Integration Tests

- Interaktion zwischen Modulen
- Mocking von Umsystemen und Infrastruktur

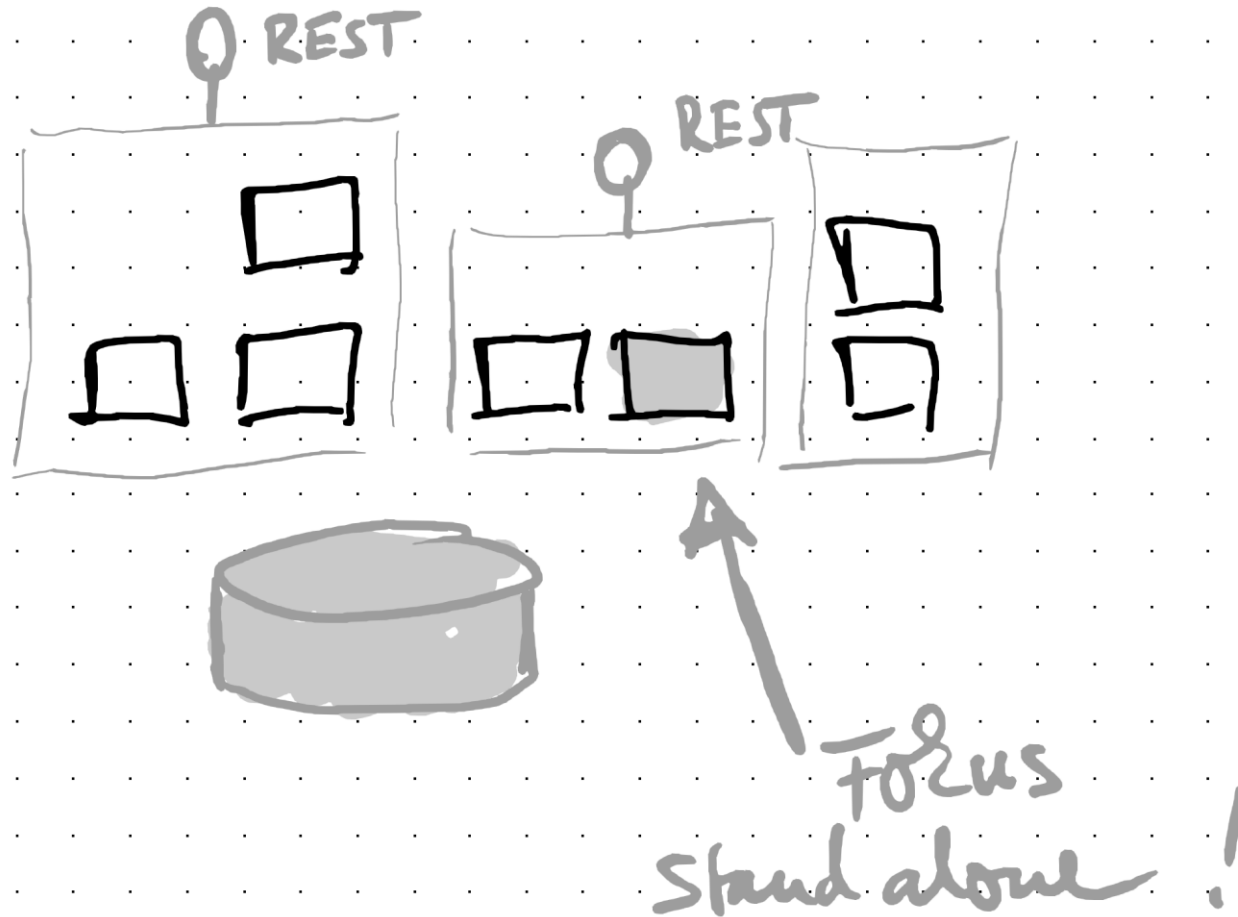
■ System Tests

- Echte Produktionsumgebung, mit Infrastruktur
- Testdaten

Mein System



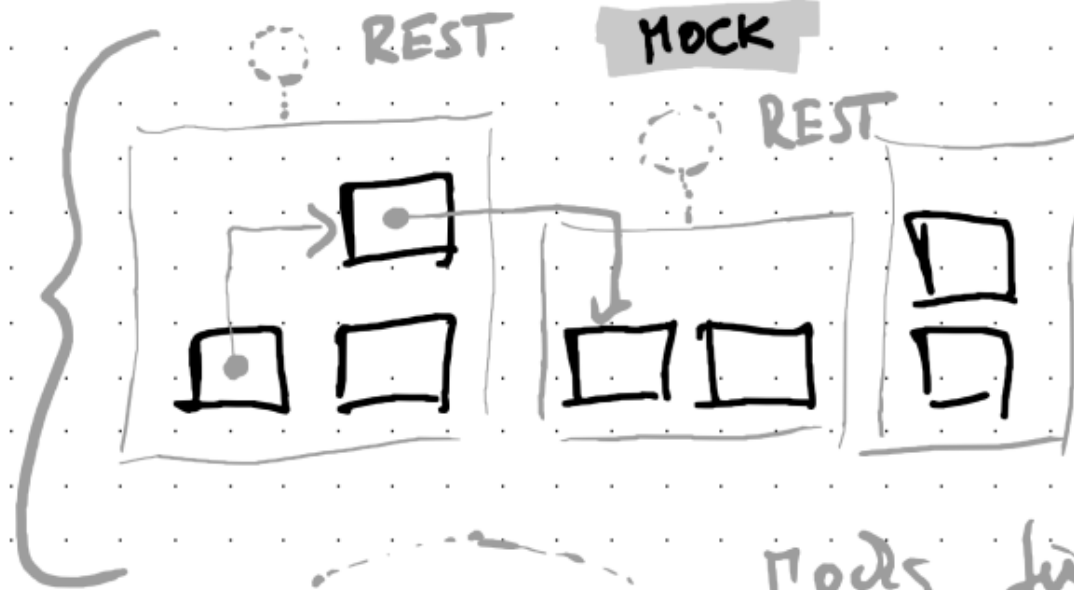
Unit Test



Integration Test

② Integration Test

Zusammen-
spiel
der
Module



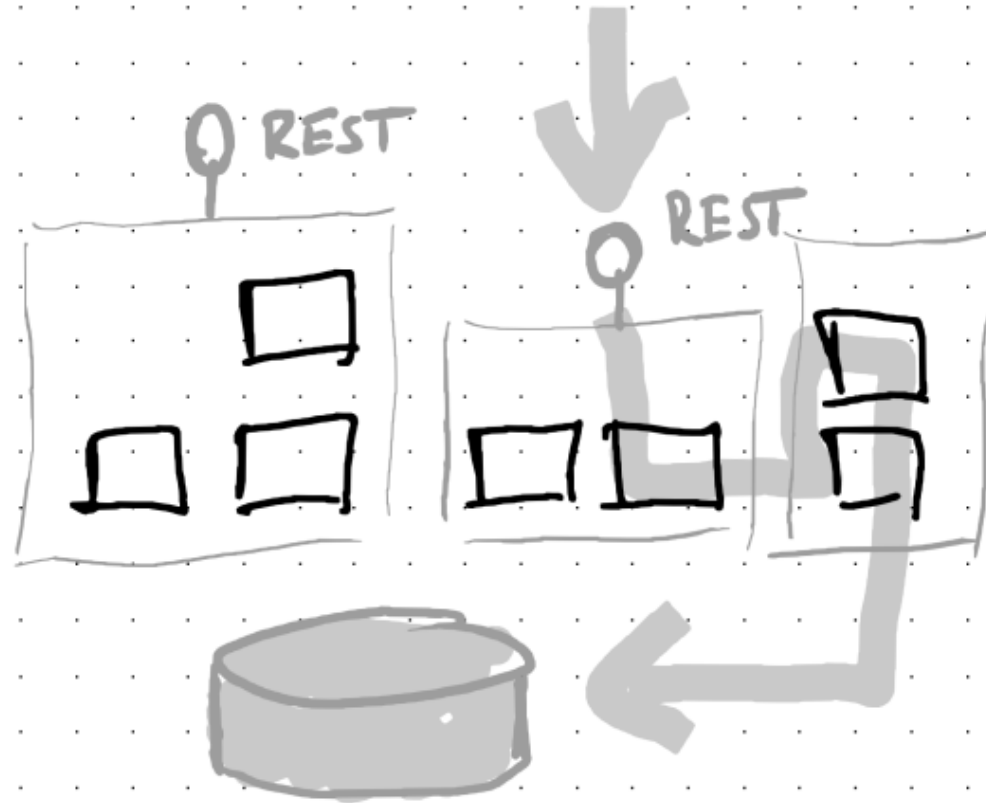
Mocks für
Frameworks

Mock = „Fälschung, Attrappe“

- **REST-Controller** benötigt Webserver
 - ... aber den wollen wir nicht in der Test-Umgebung
- **Spring MockMvc**
 - Mock eines Web Servers, ohne Web-Traffic
- **DB-Infrastruktur** zu komplex für Tests
 1. In-Memory-DB **H2** (streng genommen kein Mock)
 2. Mocking des Persistenz-Zugriffs-Layers

System Test

③ System Test



org.junit.jupiter.api.Assertions

- `assertNull(...)` + `assertNotNull(...)`
- `assertEquals(...)`
- `assertSame(...)` + `assertNotSame(...)`
- `assertArrayEquals(...)`
- `assertTrue(...)` + `assertFalse(...)`
- `assertThrows(...)`

@BeforeEach

@AfterEach

// given

// when

// then

