Week 6: Additional Topics

Bonus Unit: Test Strategy

SAP Run Simple

openSAP
open.sap.com

Test Strategy

Motivation

Goal: Ship *any* kind of change *within a day*

Prerequisite: Automated Tests

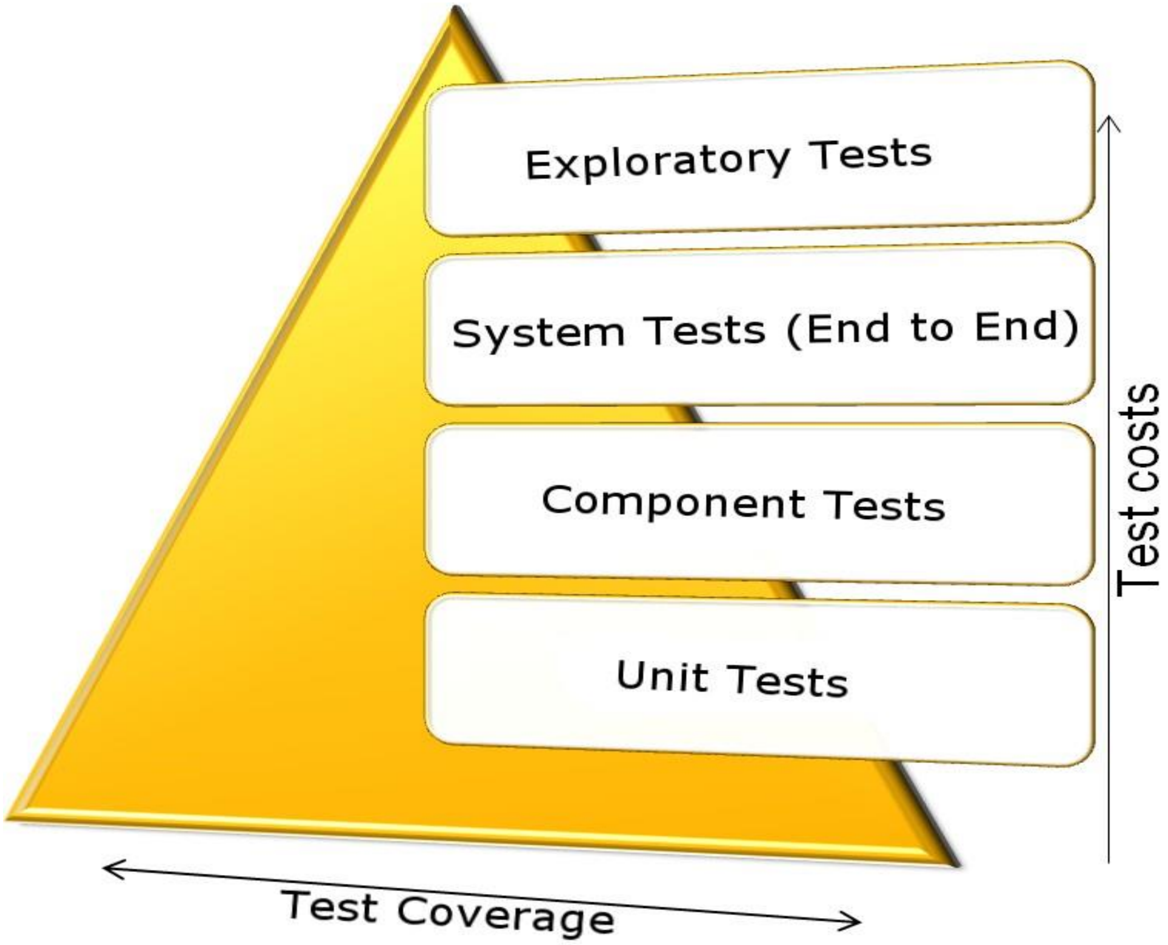
- Basis for automated delivery pipeline
- Hinders deployment of unworthy changes
- Keeps you relaxed . ..

What is an effective test strategy?

Test Strategy

Agile Testing Strategy

Implement **Testing Pyramid** to gain solid safety net

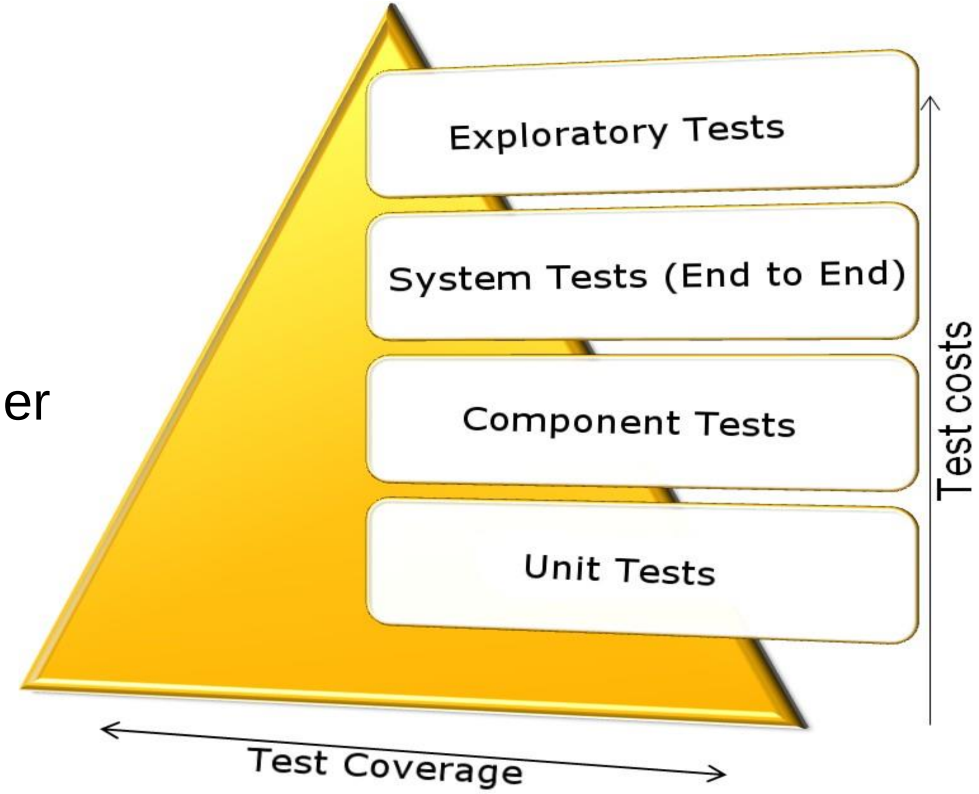


Test Strategy

Agile Testing Strategy

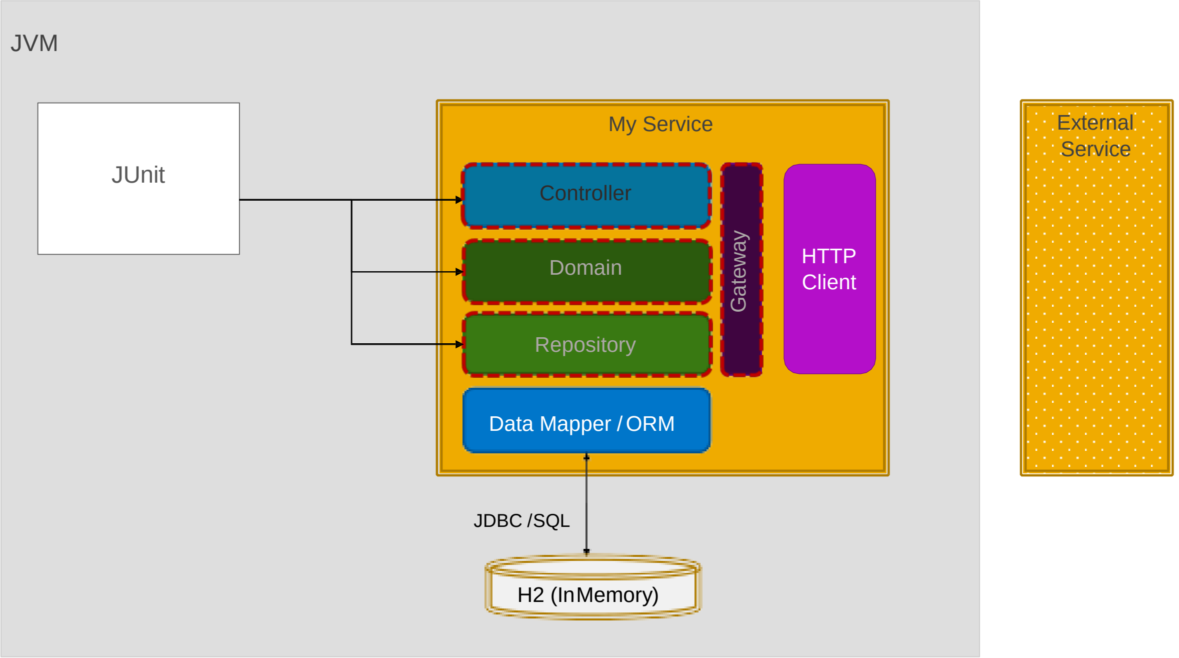
Implement Testing Pyramid
to gain solid safety net

- **Reduce Test Scope**:
  Easier to write and maintain, eases error localization
- **Focus** on testing your ***own*** **code and functionality**
- Test interaction of services using **system tests**
- Different kinds of tests should **complement** each other



Test Strategy

Unit Tests

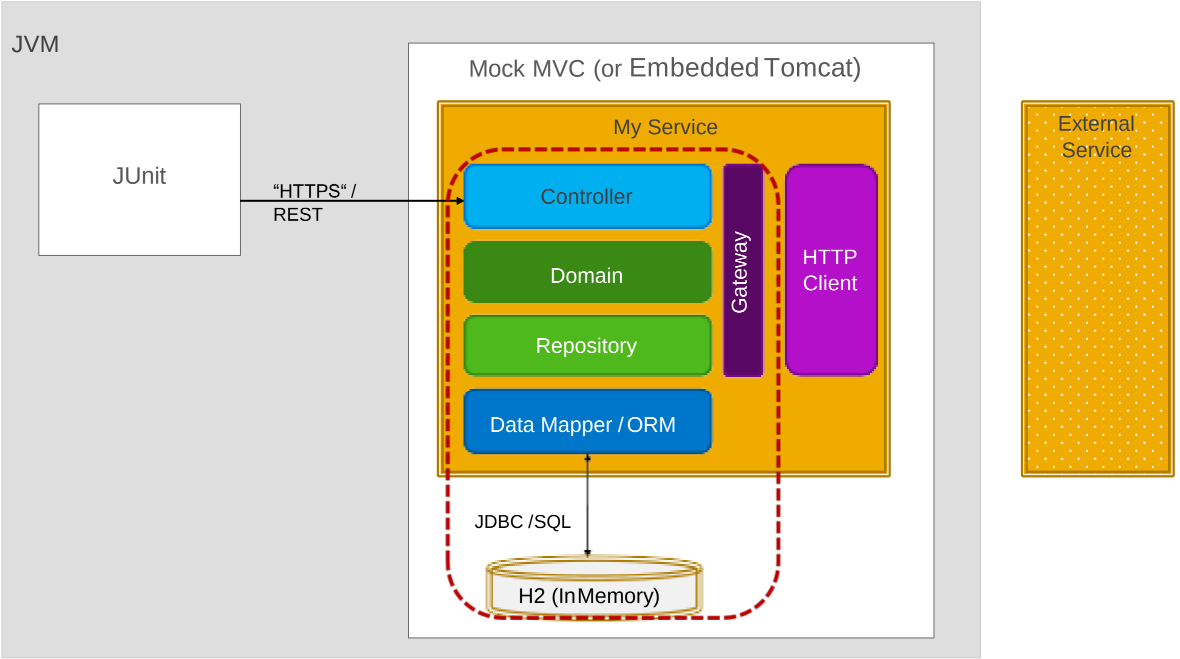


Test business logic in isolation

- Test most combinations and error cases
- Example: Test valid/invalid cases updating an advertisement

Test Strategy

Component Tests

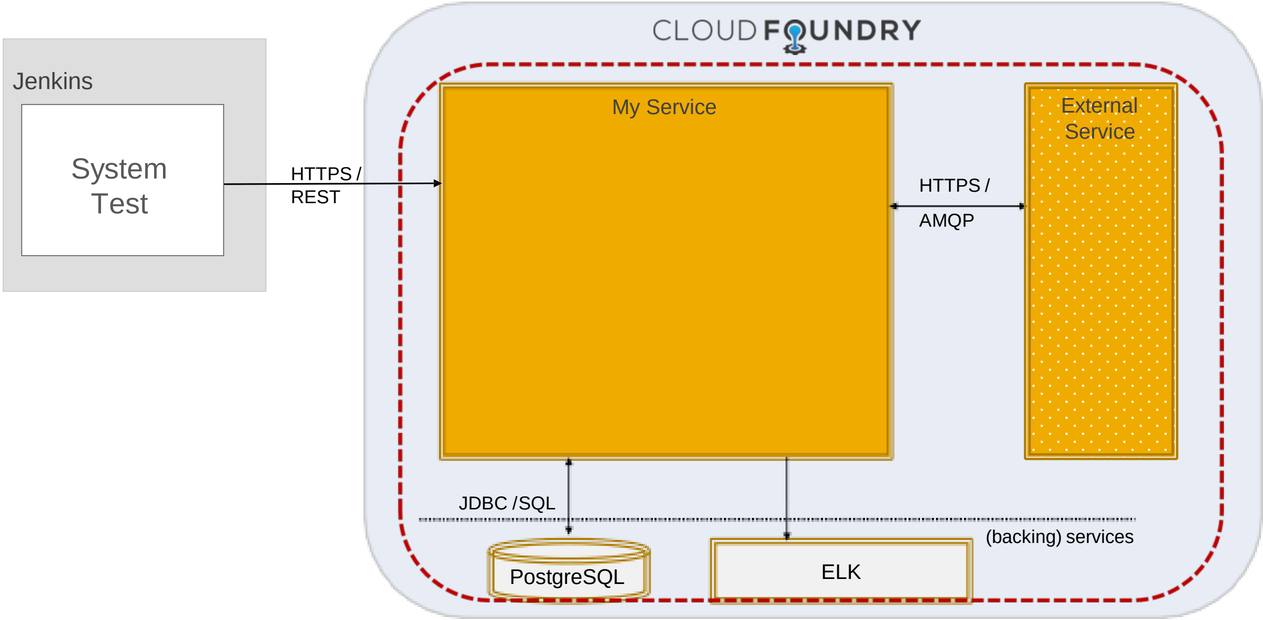


Test in runtime environment, without external dependencies

- Tests code integration with libraries and frameworks incl. security
- **Example:** Update the title of an advertisement via REST-like API

Test Strategy

System Tests



Test combination of services *deployed on Cloud Foundry test space*

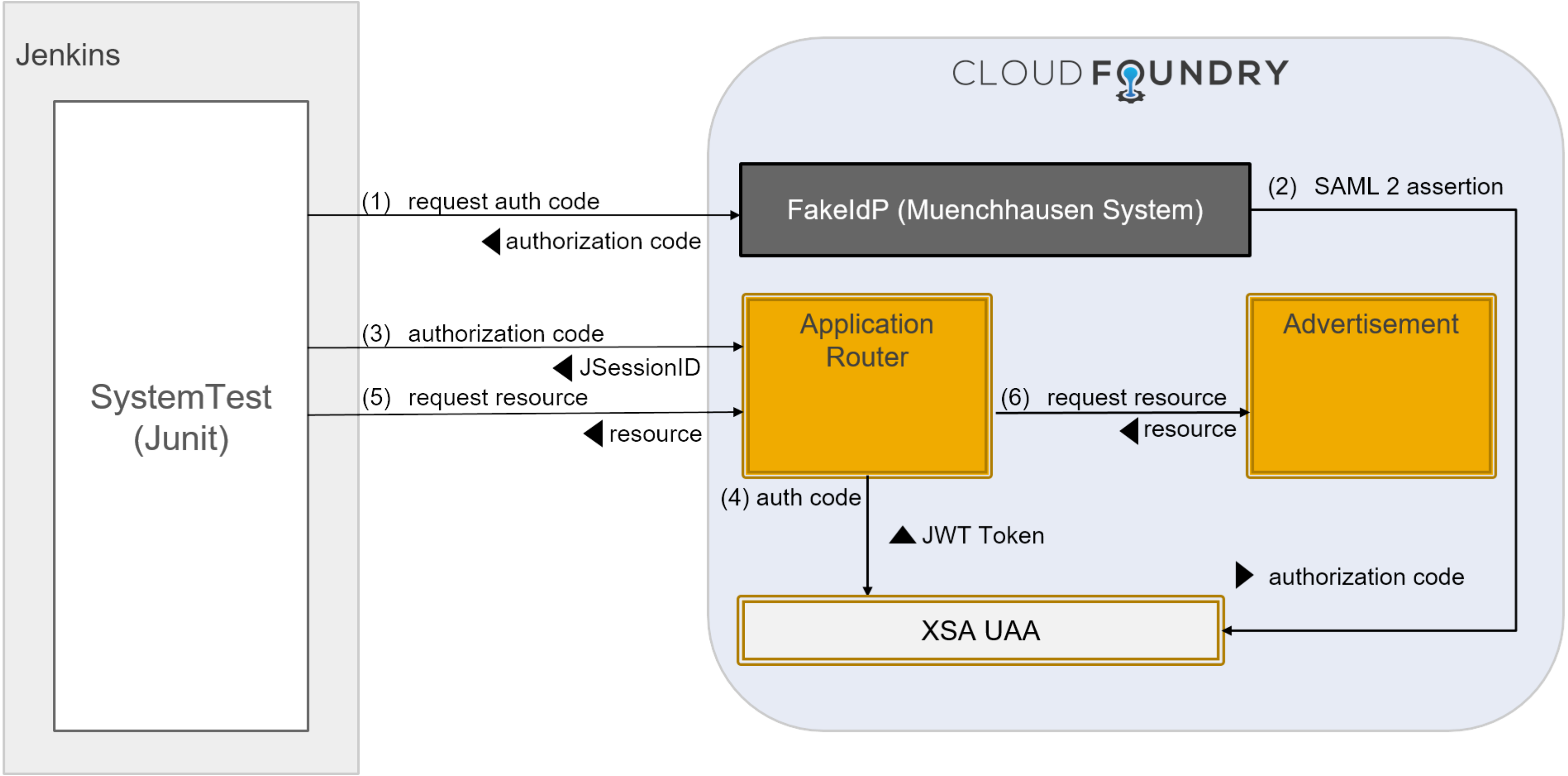
- Tests integration end to end (further tips)
  requires configurable routes to dependent services, and data
- Example: Create advertisement via REST, call external user service

Test Strategy

Excursion: Muenchhausen as Fake Identity Provider

Muenchhausen

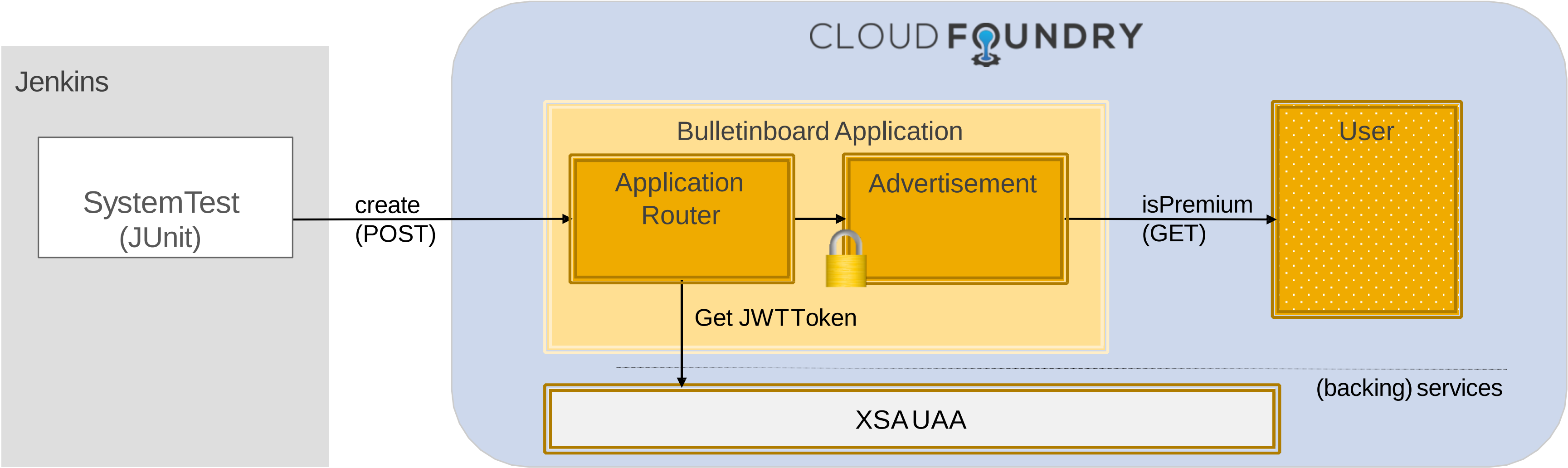
- Creates SAML2 assertion for fake user with specified SAML2 user groups



Test Strategy

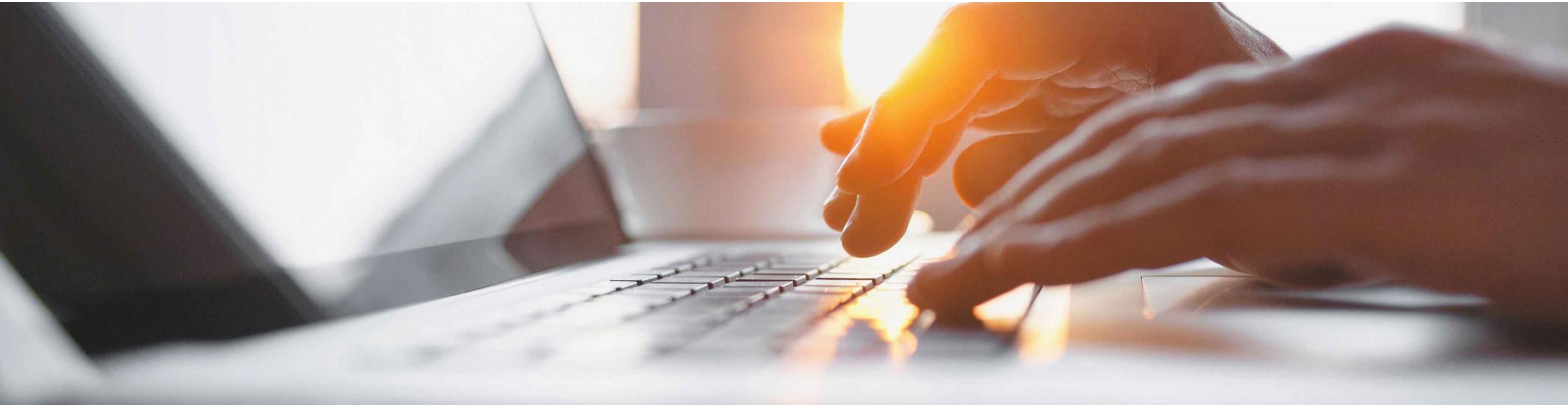
Exercise 25

Exercise 25: Create and Configure System Test using Muenchhausen



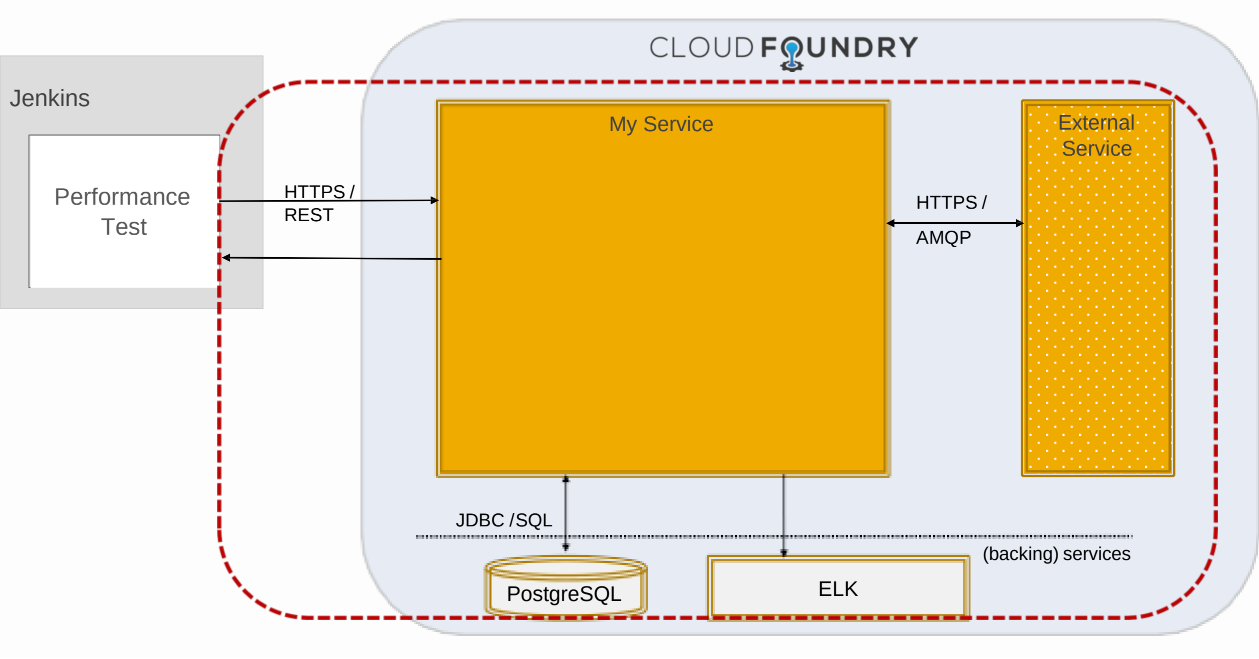
Test Strategy

Review of system test



Test Strategy

Performance Tests - Find performance drops or bottlenecks before production



Test expected load (# concurrent users, # HTTP connections)

- Tests whether response time is below threshold
- Example: Create, read many ads in parallel via multi-tenant approuter and Muenchhausen.

Test Strategy

JMeter Demo

JMeter

- Scriptable
- Master-Slave support
- Plugins for Maven and Jenkins

Exercise: Performance testing with JMeter

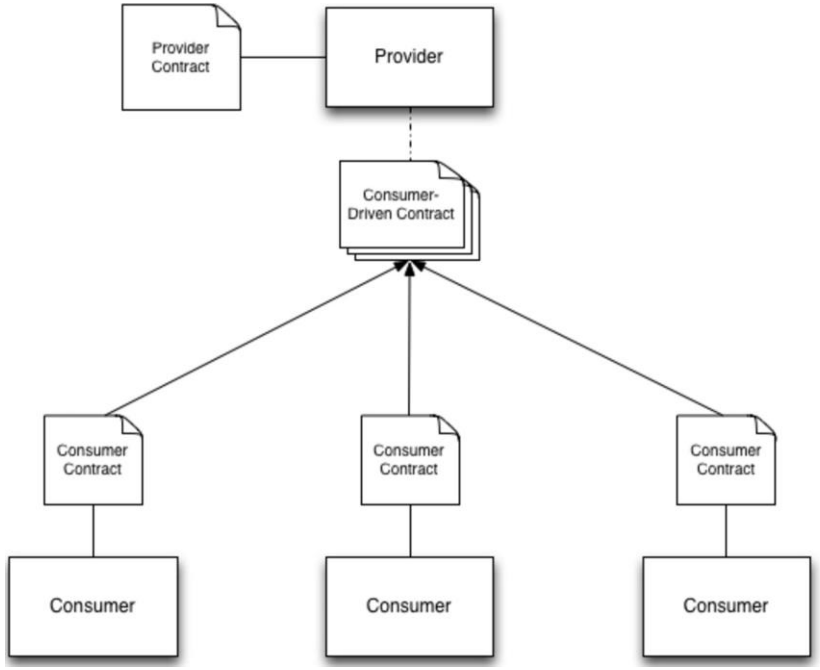
Test Strategy

Performance Testing Remarks

- Performance tests prone to fluctuations, especially in the cloud
- Resilience libraries influence test results (Hystrix)
- Averages or percentiles are misleading, measure maximum instead!
  - Many requests per page => worst case performance is relevant
- In addition make use of performance monitoring
  - Example: Measure latency for downstream requests

Test Strategy

Contract Tests



[The Dilemma of Testing Microservices in Integration]

Test provider’s implementation against consumers’ requirements

- Tests are defined by consumers
- Provider considers consumers’ tests
- Example: self contained JUnit tests in provider’s code base

Test Strategy

Take Aways

- Write tests according to the Test Pyramid (e.g. write as few end-to-end tests as possible)
- Write black box tests (test visible behavior of public methods)
- Practice Test Driven Development (TDD)
- Execute tests locally before submitting changes
- Setup local/test environment similar to production

Note: You are unable to test each and everything!

Testing Strategies in a Microservice Architecture

In addition to testing: extensive monitoring

Testing Strategy

Further reading

i Additional Material

- Testing Strategies in a Microservice Architecture
- Exercise: Create and Configure System Test using Muenchhausen
- Exercise: Performance testing with Jmeter
- SAP AgileSE Wiki: Testing Pyramid
- Muenchhausen
- SAP Jam: Cloud Curriculum

Course

- Continuous Delivery and DevOps Course (SAP internal)