Component Testing with Spring Tools

Billy Korando

SENIOR SOFTWARE CONSULTANT - KEYHOLE SOFTWARE

@BillyKorando

Agenda

Defining Component Integration Testing

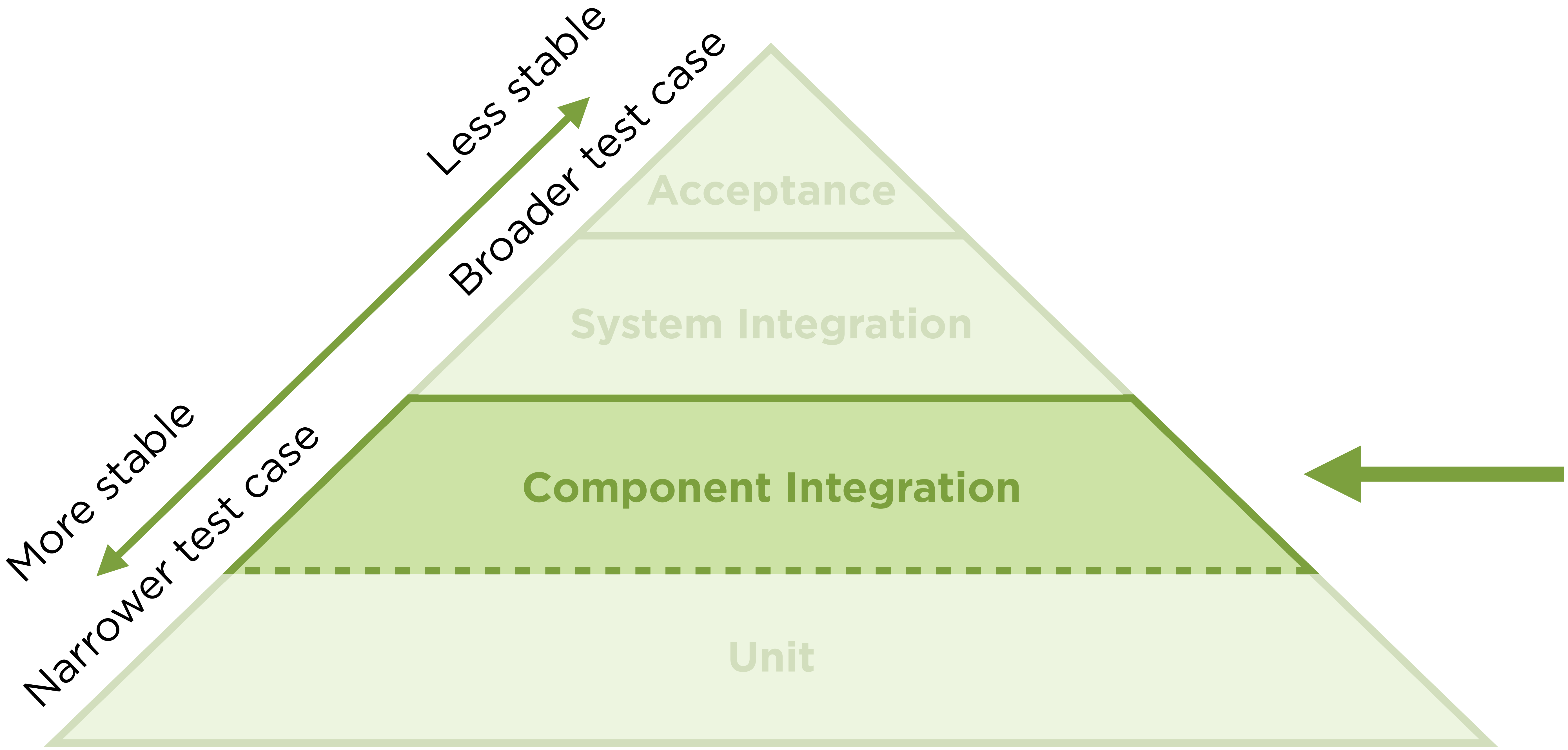
Using Spring Testing Utilities to Test:

- RESTful APIs
- The Data Layer
- Security

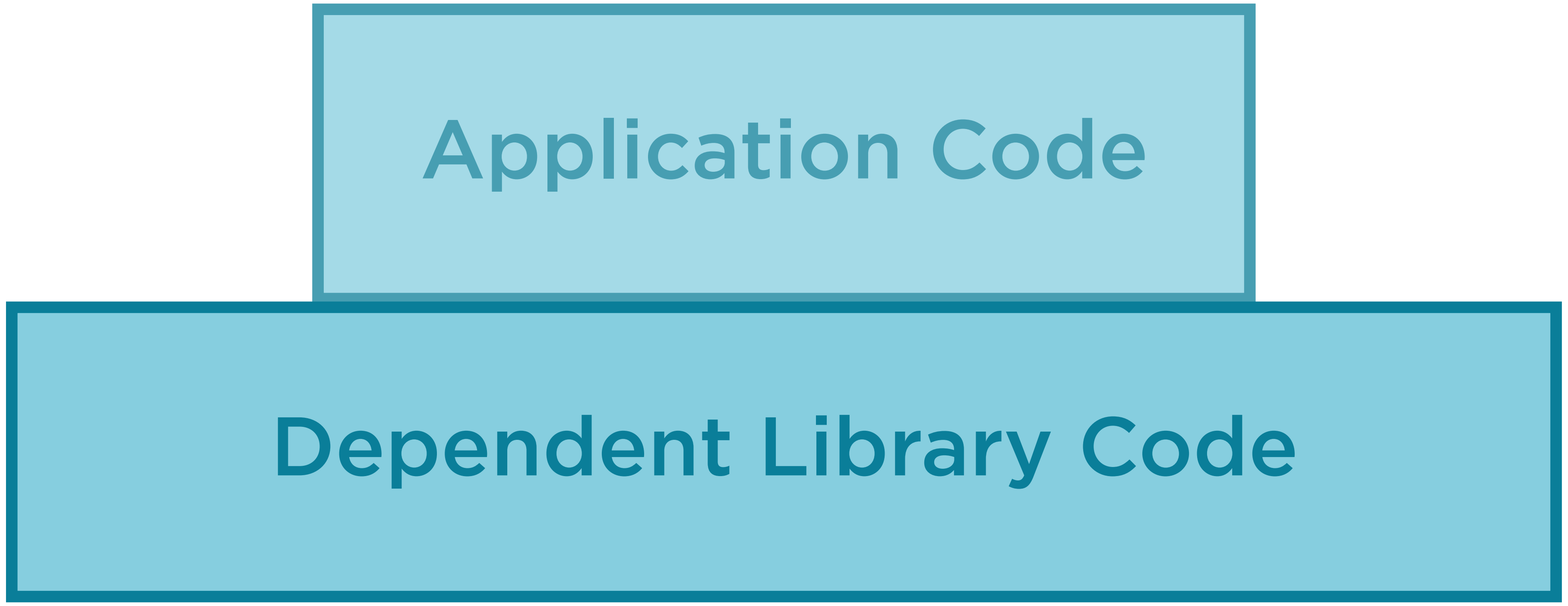
Test JSON Marshaling and Unmarshaling

Testing Aspects

What is Component Integration Testing?



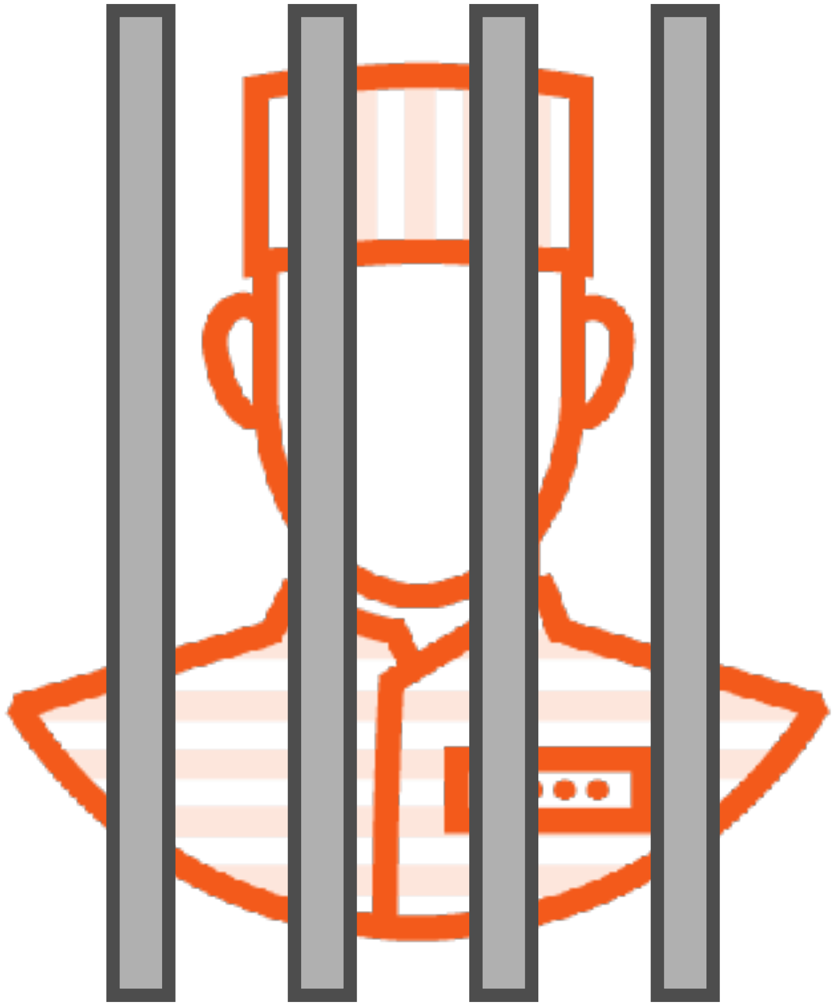
What is Component Integration Testing?



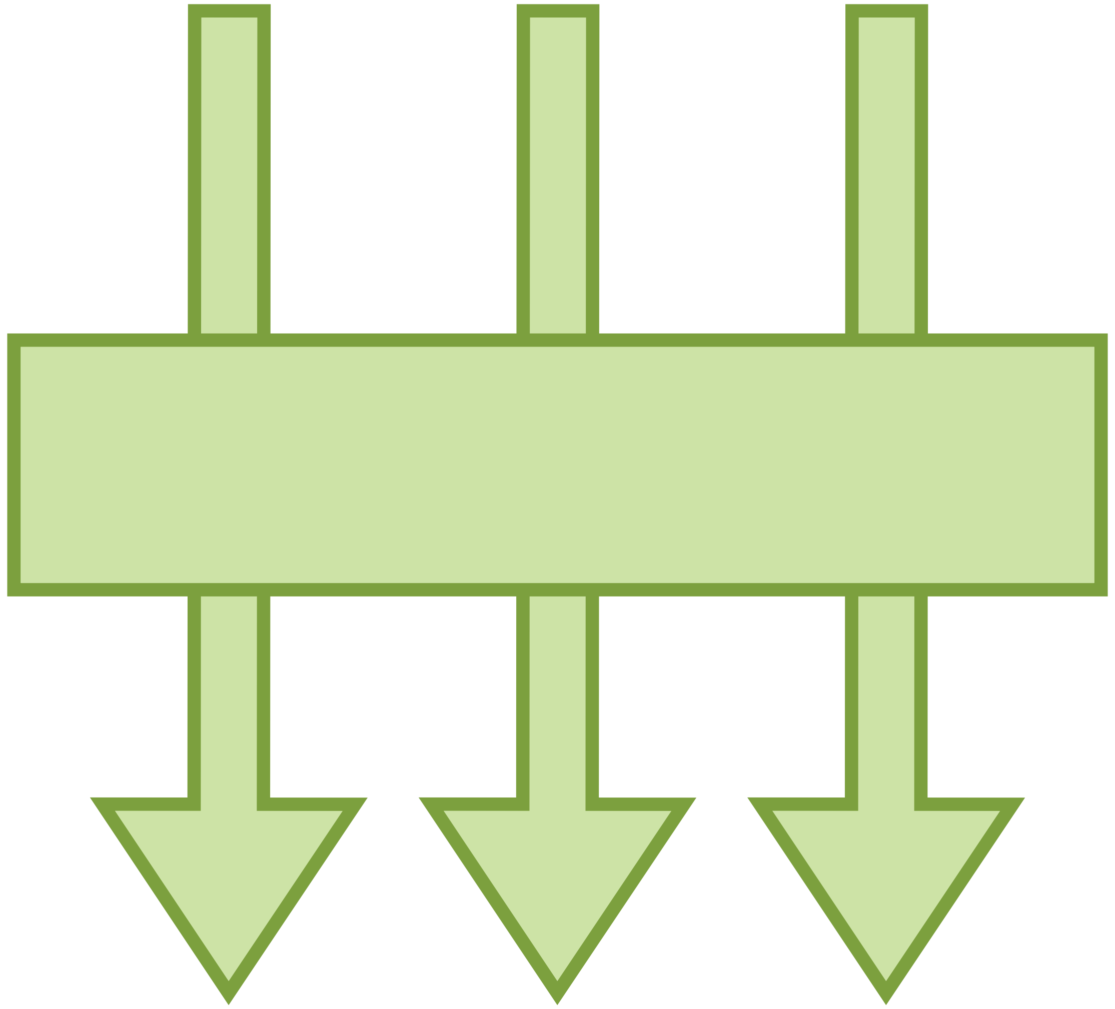
Application Code

Dependent Library Code

Goals of Component Integration Testing



Goals of Component Integration Testing



```java
@RunWith(SpringRunner.class)
@DataJpaTest
public class TestCustomerRepo{

   @Autowired
   private TestEntityManager entityManager;

   @Autowired
   private CustomerRepo repo;

   ...

}
```

◀ Initialize Spring Container

◀ Initialize Layer Under Test

◀ Spring Provides Tools to Make Testing Easier

◀ Spring Manages Test Dependencies

Goals of Component Integration Testing

Verify Cross Cutting Concerns

Keep Up With Latest Security Patches

Keep Up With Latest Library Features

RESTful API Testing with MockMvc

Demo

Use MockMvc to Exercise a RESTful API

Validate Default SpringMVC Behaviors

Data Layer and JPA Testing

Demo

Use Spring Boot Tools to Test the Data Layer

Verify Spring Data's Query Builder Functionality

Security Testing

JSON Testing

Demo

Parse a JSON String

Convert an Object to a JSON String

Verify Jackson Enforces Required Fields

Verify Jackson Ignores Extra Fields

Aspect Testing

Summary

How to and Why Write Component Tests

Sprint Boot Test Tools