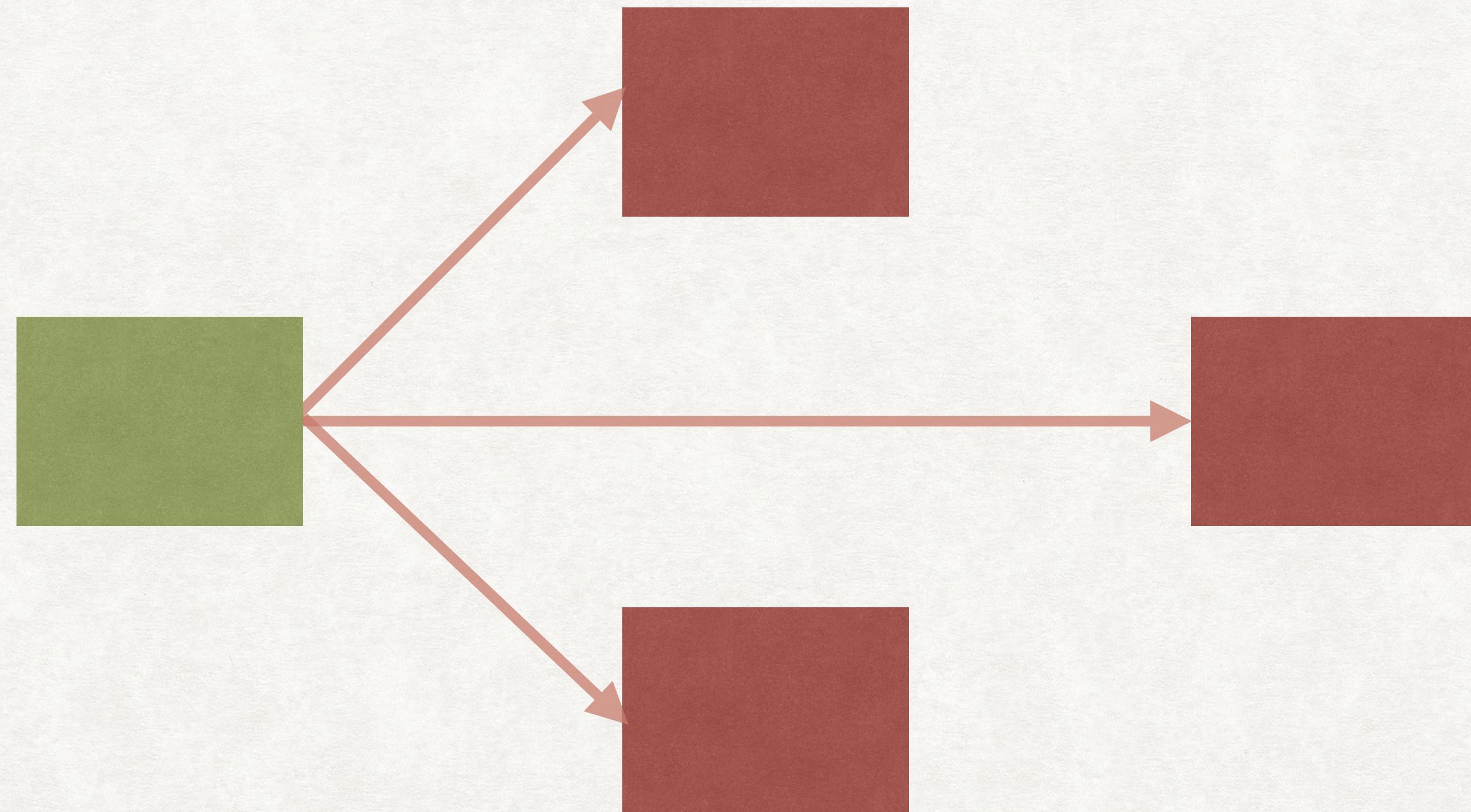


CONTRACT DRIVEN DEVELOPMENT WITH
SPRING CLOUD
CONTRACT

PROBLEMS WITH DECOUPLED SERVICES

- How do we test an application that communicates with other services?



PROBLEMS WITH DECOUPLED SERVICES

- Create end to end tests
 - Real tests like production, but comes at a cost
 - May require a few other services, databases
 - Difficult to debug if a problem occurs
 - Late feedback makes it difficult to discover and fix bugs early

PROBLEMS WITH DECOUPLED SERVICES

- Mock dependencies in unit and integration tests
 - Very fast feedback and no infrastructure
 - May differ vastly from reality
 - Can end up in production with tests that pass, but would fail in production

PROBLEMS WITH DECOUPLED SERVICES

- Spring Cloud Contract helps to fill the gap
 - Ensures that HTTP messaging stubs are exactly how the server will behave
 - Promotes acceptance test driven development
 - Publish contracts that can immediately be used by the consumer and producer
 - Generates and guides tests
- In reality we want to have a **balance** of end to end, mocking and contract tests

CONTRACT DRIVEN DEVELOPMENT

ALSO KNOWN AS CONSUMER DRIVEN CONTRACTS

- Specify how we expect the client to interact with our service
- There may be multiple contracts per endpoint
- The API shape (or structure) does not define behaviour
- Contracts provide quick feedback on evolving APIs

EXAMPLE CONTRACT

```
org.springframework.cloud.contract.spec.Contract.make {  
    request {  
        method 'GET'  
        url '/todo'  
        headers {  
            header('Content-Type', 'application/json')  
        }  
    }  
    response {  
        status 200  
        body([])  
    }  
}
```

WHAT HAPPENS TO THE CONTRACT?

```
@Test
public void validate_task_list_is_empty() throws Exception {
    // given:
    MockMvcRequestSpecification request = given();

    // when:
    ResponseOptions response = given().spec(request)
        .get("/todo");

    // then:
    assertThat(response.statusCode()).isEqualTo(200);

    // and:
    DocumentContext parsedJson = JsonPath.parse(response
        .getBody().asString());
}
```

WHAT HAPPENS TO THE CONTRACT?

A STUB IS CREATED

- The contracts also form stubs
- Stubs can be used to mock out services
- Allows consumers to work on their applications, whilst serverside logic is developed
- Decouples dependencies during the development process