MOCK TESTING

MATT HAVILAH (mhavilah@gmail.com)

FEBRUARY, 2019

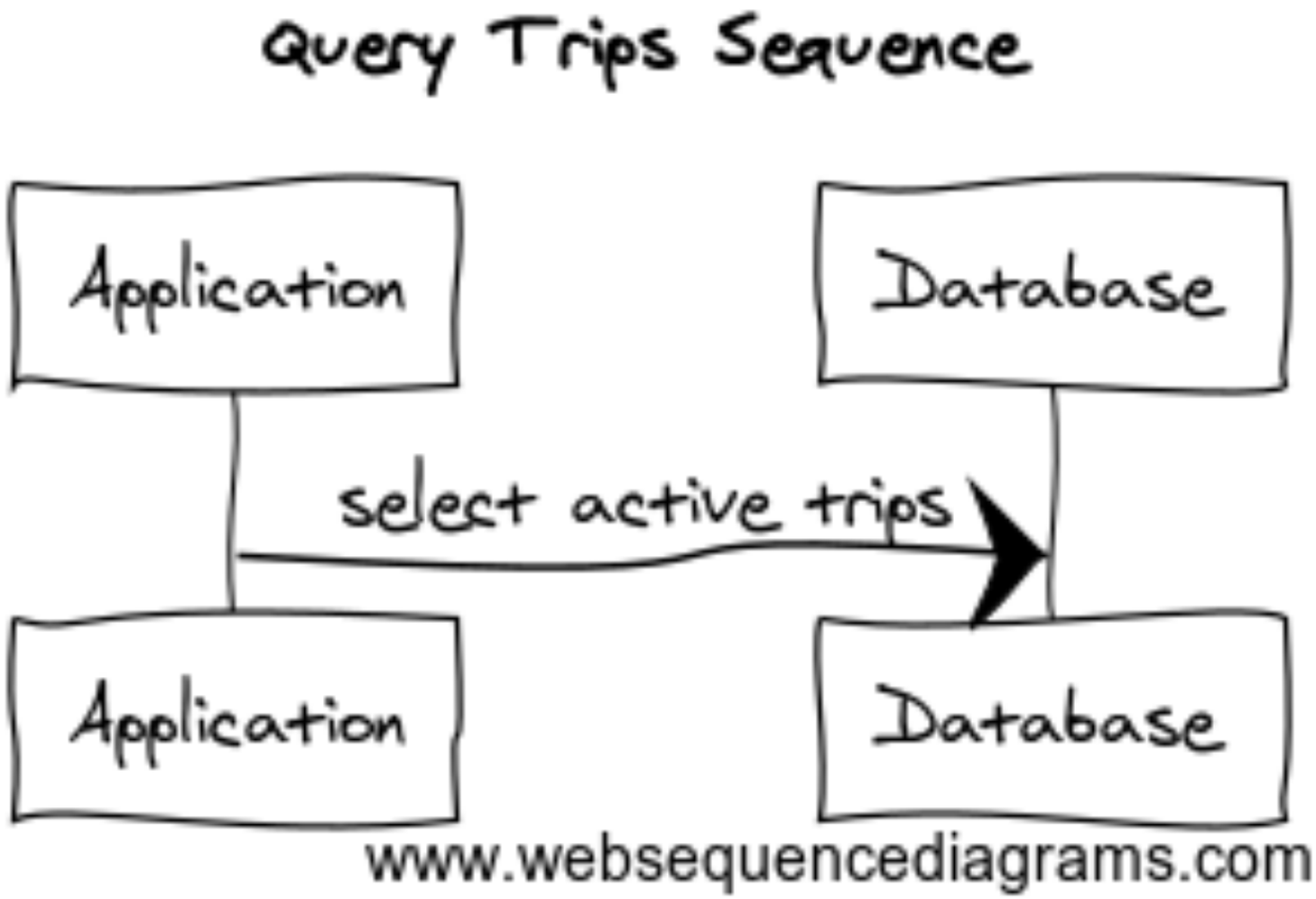
HOW DO WE TEST “COMPLEX” DEPENDENCIES ?

HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- Databases

HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- Databases

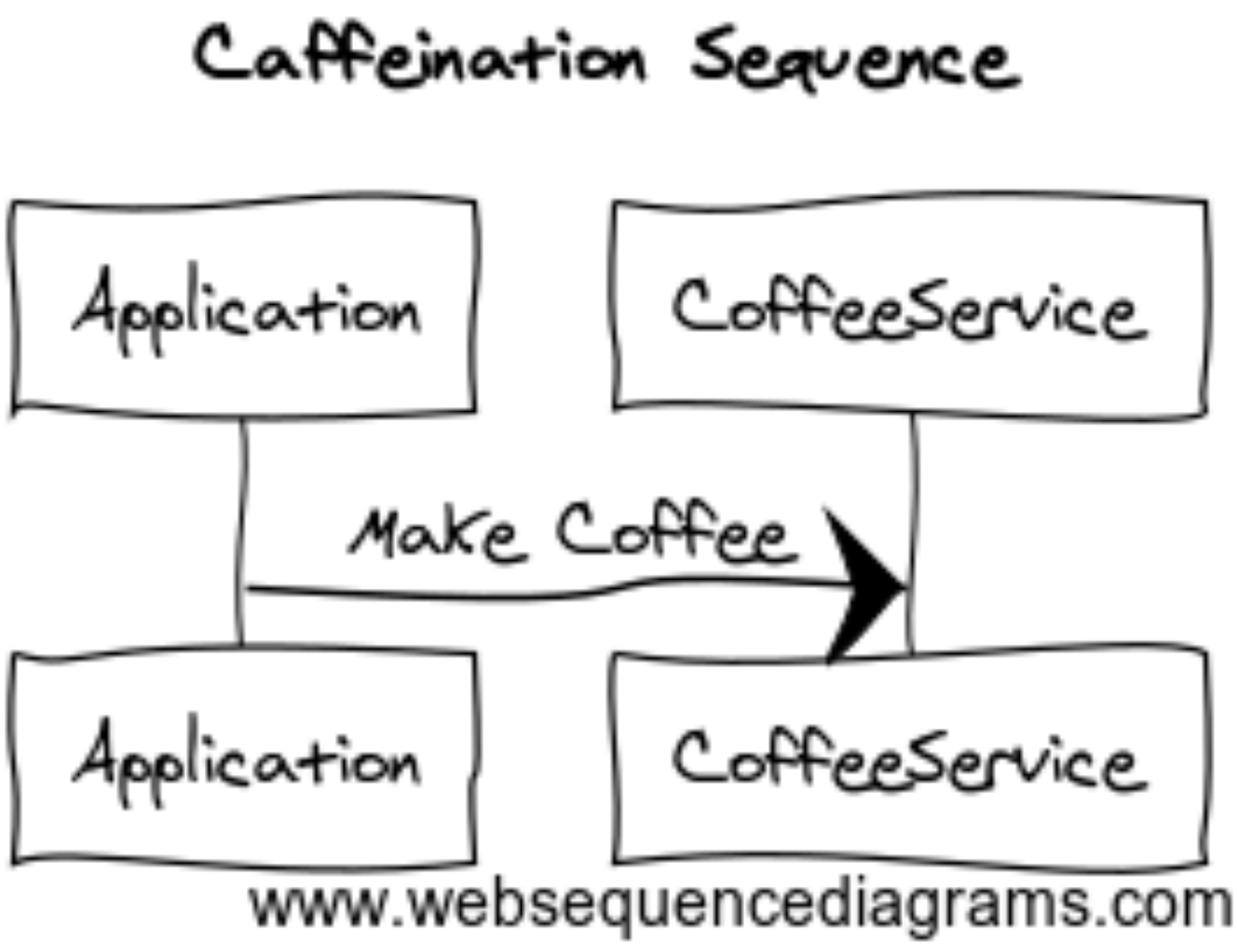


HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- Databases
  - Slow
  - Tedious to install, initialise
  - Network connectivity issues
  - Cleaning test state

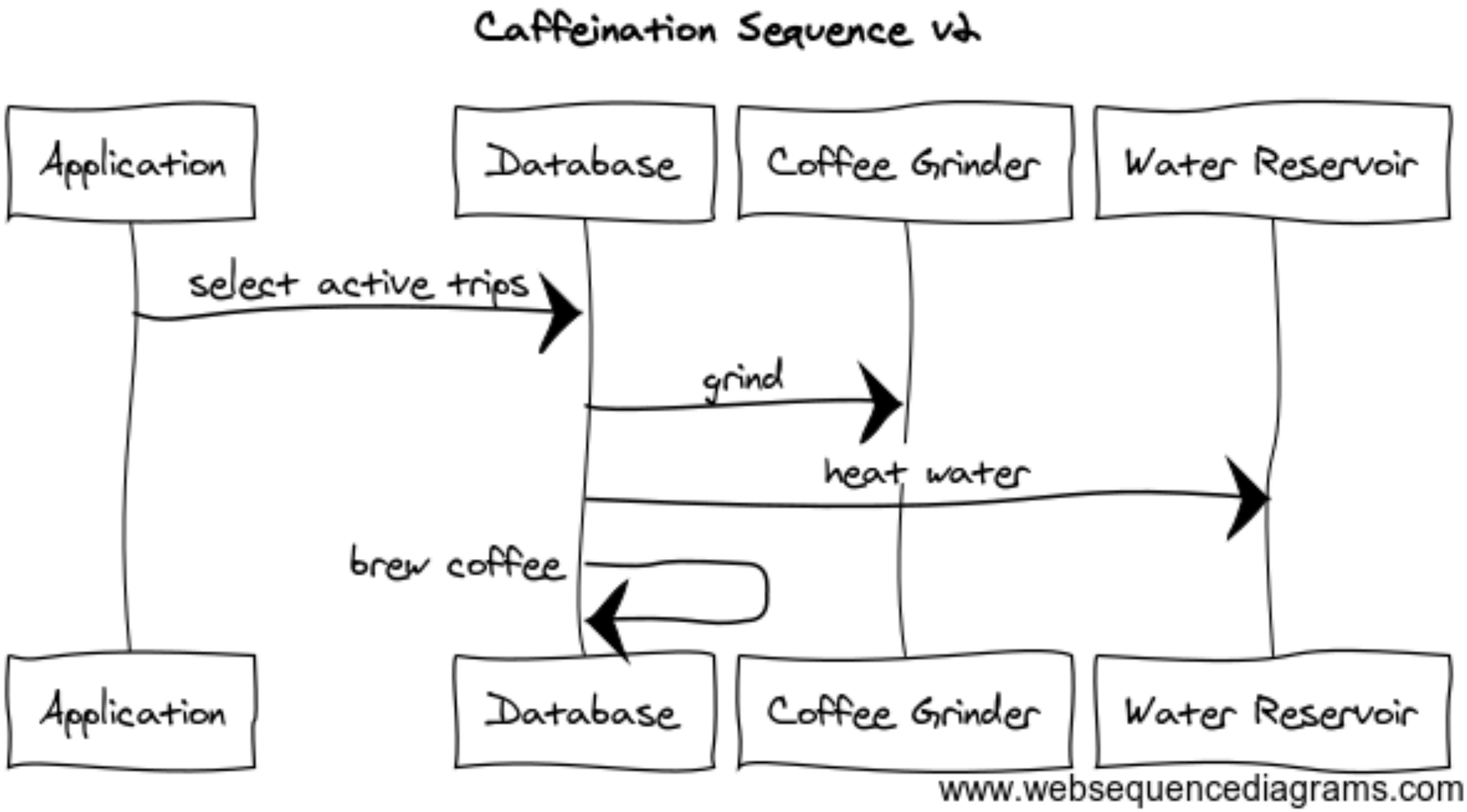
HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- Network Services



HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- "Complex" Network Services



HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- Network Services
  - E-commerce platform
  - Payment Gateway Service
  - Inventory level Service

HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- Network Services
  - E-commerce platform
    - Payment Gateway Service
    - Inventory level Service
  - Trading Platform
    - Real-time share price feed

HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- "Real world" resources

HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- "Real world" resources
  - Mobile apps - GPS Location
  - System time/clock

HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- "Real world" resources
  - Mobile apps - GPS Location
  - System time/clock
- "Real time" streams
  - Web-sockets and message queues
  - Mouse events and user interaction

HOW DO WE TEST "COMPLEX" DEPENDENCIES ?

- "Real world" resources
  - Mobile apps - GPS Location
  - System time/clock
- "Real time" streams
  - Web-sockets and message queues
  - Mouse events and user interaction
- Other resources
  - Display
  - IoT device/sensor

TEST OPTIONS

- Test with the “real thing”

TEST OPTIONS

- Test with the “real thing”
  - Acceptance/End-to-end Testing
  - Unit tests with “real” implementations

TEST OPTIONS

- Test with the “real thing”
  - Acceptance/End-to-end Testing
  - Unit tests with “real” implementations
  - “Accurate” representation, but,…

TEST OPTIONS

- Test with the “real thing”
  - Acceptance/End-to-end Testing
  - Unit tests with “real” implementations
  - “Accurate” representation, but,…
  - Slow, Expensive
  - Difficult to setup/teardown
  - How to simulate failure scenarios ?

TEST OPTIONS

- Test with a “simulation”

TEST OPTIONS

- Test with a “simulation”
  - A “smart stub” replicating real world behaviour
  - eg, Train location simulator, Payments Gateway simulator

TEST OPTIONS

- Test with a “simulation”
  - A “smart stub” replicating real world behaviour
  - eg, Train location simulator, Payments Gateway simulator
  - Good for testing “sunny” day scenarios

TEST OPTIONS

- Test with a “simulation”
  - A “smart stub” replicating real world behaviour
  - eg, Train location simulator, Payments Gateway simulator
  - Good for testing “sunny” day scenarios
  - May require significant effort to develop
  - May have defects themselves due to complexity
  - Focus on verifying state (assertThat ….)

TEST OPTIONS

- Test with a “Mock”

TEST OPTIONS

- Test with a “Mock”
  - Mocks *imitate* collaborator behaviour
  - without implementing complex logic

TEST OPTIONS

- Test with a “Mock”
  - Mocks *imitate* collaborator behaviour
  - without implementing complex logic
    - Setup the mock with an *expectation* of interaction/ behaviour

TEST OPTIONS

- Test with a “Mock”
  - Mocks *imitate* collaborator behaviour
  - without implementing complex logic
    - Setup the mock with an *expectation* of interaction/ behaviour
    - Exercise the unit/system under test

TEST OPTIONS

- Test with a “Mock”
  - Mocks *imitate* collaborator behaviour
  - without implementing complex logic
    - Setup the mock with an *expectation* of interaction/behaviour
    - Exercise the unit/system under test
    - Verify the interaction/behaviour took place
  - Can easily imitate failure scenarios (exception behaviour)

SIMPLE EXAMPLE

- Cafe Application

SIMPLE EXAMPLE

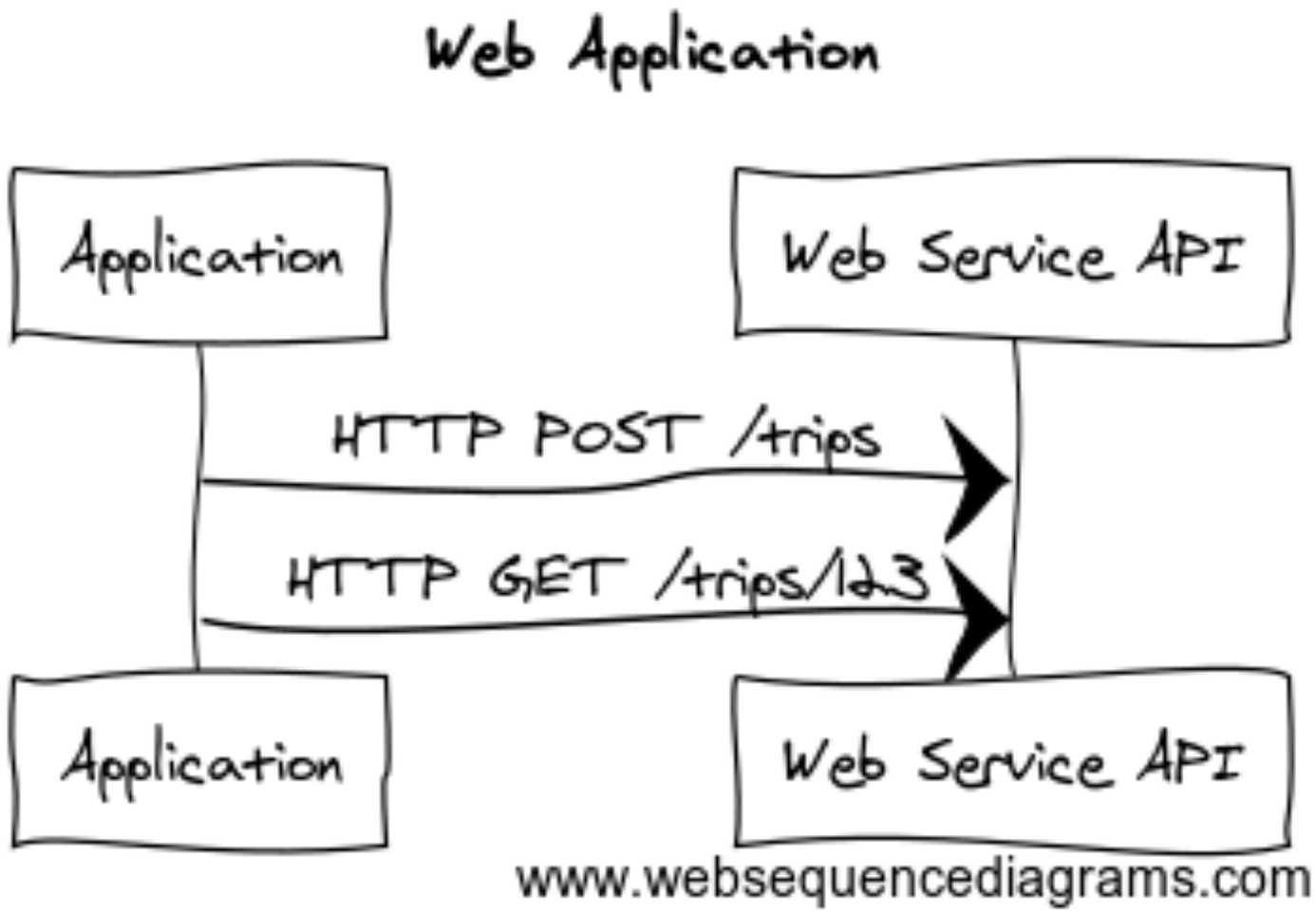
- Cafe Application
  - Services
    - Make Beverage (Coffee, Tea etc)

SIMPLE EXAMPLE

- Cafe Application

NEXT STEPS – API TESTING

- Remote Service APIs are just another collaborator



NEXT STEPS – API TESTING

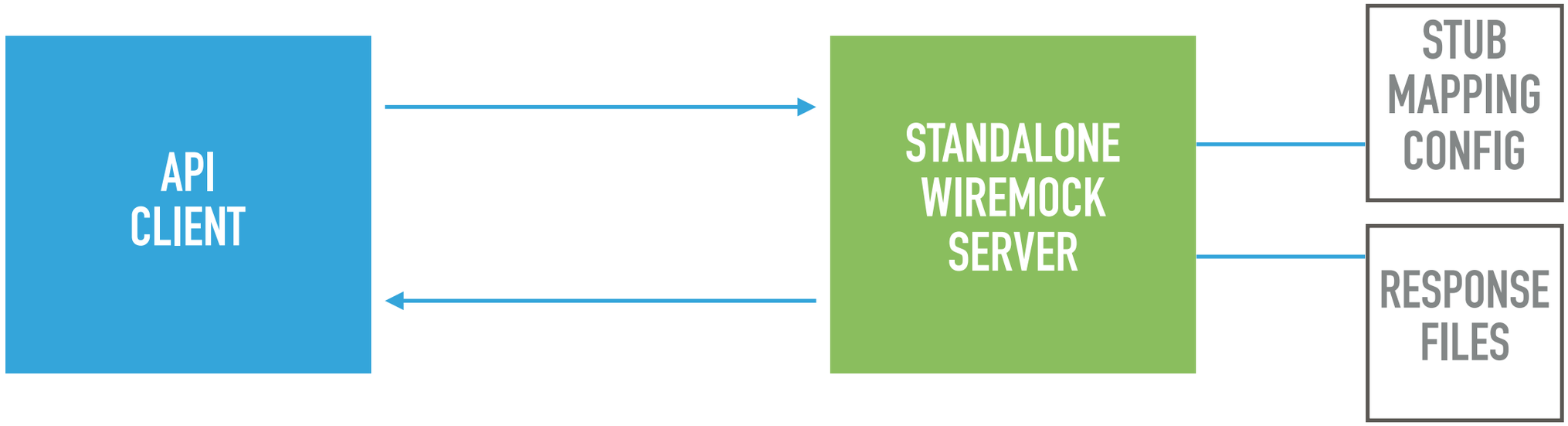
- Remote Service APIs are just another collaborator
  - Examples:
    - Mock the Connection
      - SpringRestTemplate
      - JDBCConnection
      - HttpUrlConnection

NEXT STEPS – API TESTING

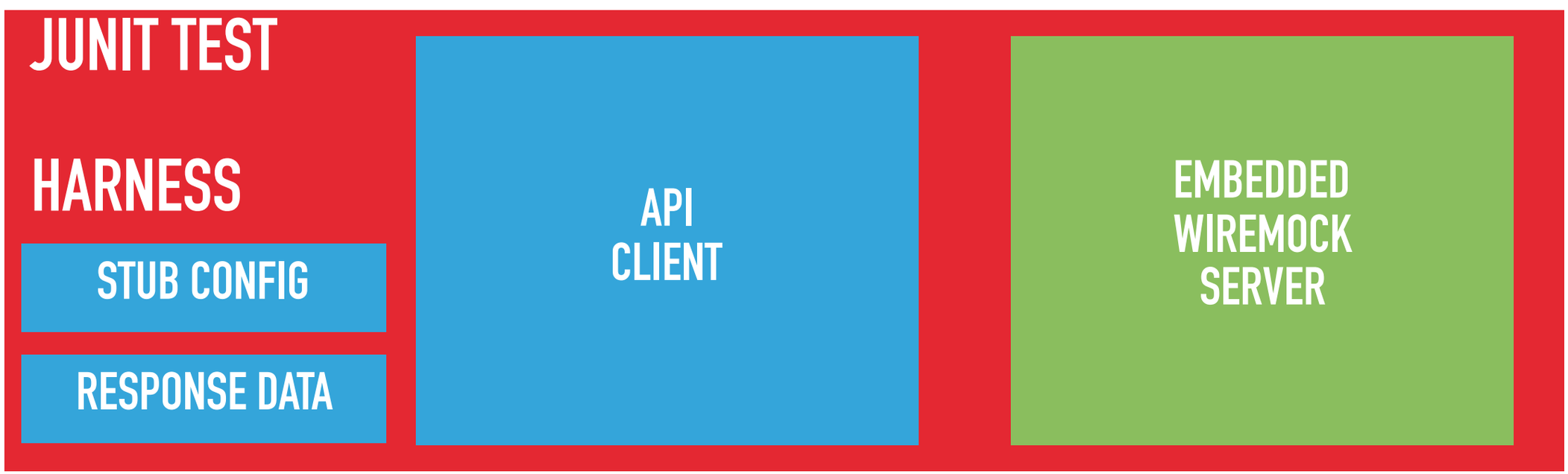
- Remote Service APIs are just another collaborator
  - Or use a *WireMock Server*
    - Actual HTTP Server
    - Similar to Mocks, specify the expected *behaviour*
    - Exercise the client under test
    - Verify expected APIs got called

NEXT STEPS – API TESTING

STANDALONE DEPLOYMENT



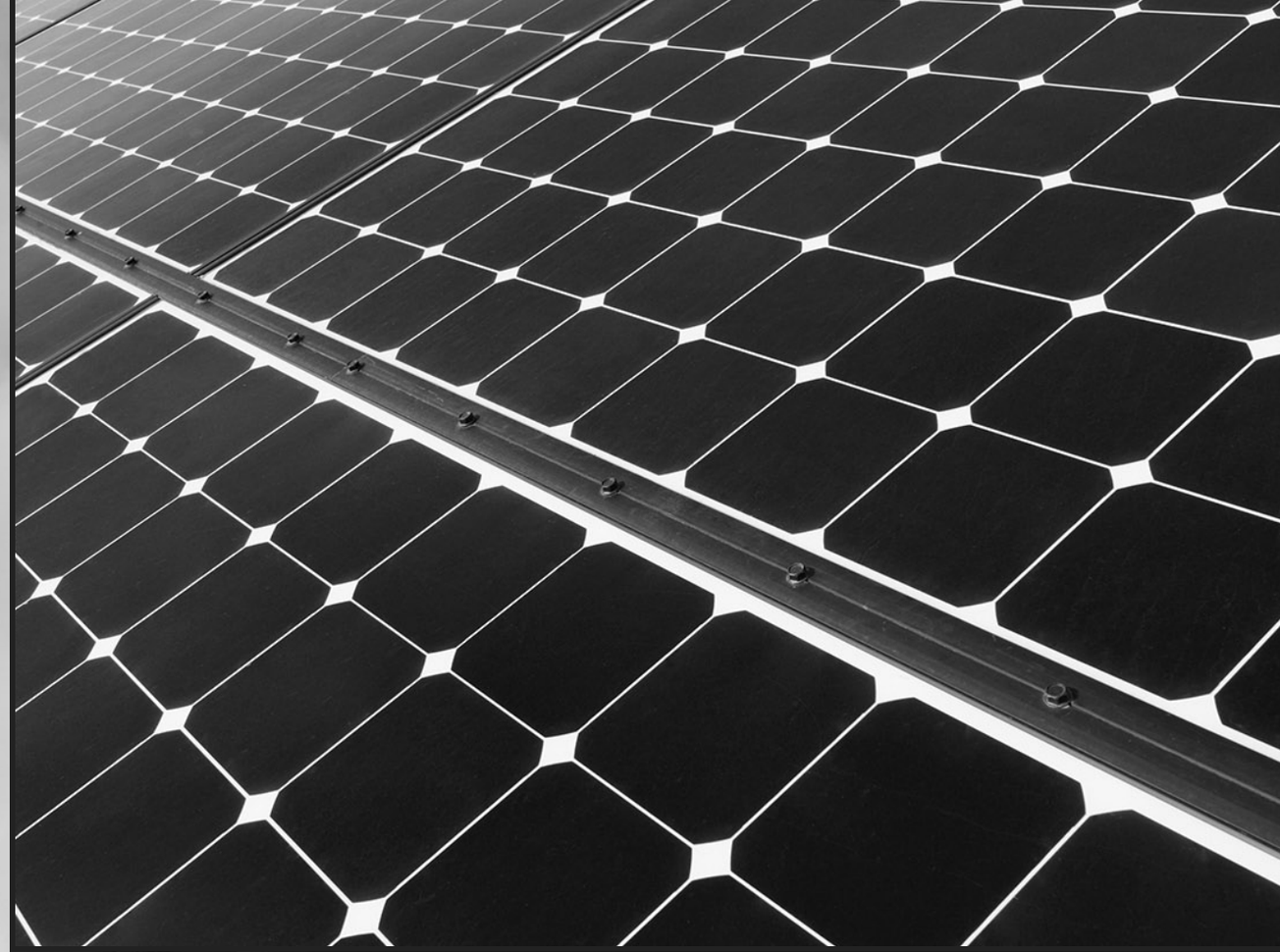
EMBEDDED DEPLOYMENT



NEXT STEPS – API TESTING

```
class RequestMatchingTest {

    private RestTemplate restTemplate;
    private WireMockServer wireMockServer;

    @Test
    @DisplayName("Should compare the actual URL with the expected URL regex")
    void shouldCompareActualUrlWithExpectedUrlRegex() {
        givenThat(get(urlMatching("/api/([a-z]*)\\?id=1"))
                .willReturn(aResponse().withStatus(200))
        );

        String serverUrl = buildApiMethodUrl(1L);
        ResponseEntity<String> response = restTemplate.getForEntity(serverUrl,
                String.class
        );
        assertThat(response.getStatusCode()).isEqualTo(HttpStatus.OK);
    }

    private String buildApiMethodUrl(Long messageId) {
        return String.format("http://localhost:%d/api/message?id=%d",
                this.wireMockServer.port(),
                messageId
        );
    }
}
```

QUESTIONS



REFERENCES

- https://martinfowler.com/bliki/TestDouble.html
- https://martinfowler.com/articles/mocksArentStubs.html
- https://www.baeldung.com/mockito-2-java-8
- https://dzone.com/articles/creating-smart-stubs-mock
- https://dzone.com/articles/mockito-custom-answers

REFERENCES CON'D…

- Wiremock References
  - http://wiremock.org/docs/
  - https://dzone.com/articles/wiremock-mock-your-rest-apis
  - http://wiremock.org/docs/simulating-faults/
  - https://www.petrikainulainen.net/programming/testing/wiremock-tutorial-introduction/
  - https://www.petrikainulainen.net/programming/testing/wiremock-tutorial-request-matching-part-one/
  - https://www.petrikainulainen.net/programming/testing/wiremock-tutorial-request-matching-part-four/
- https://github.com/abstracta/wiresham - TCP Wire mock server