

Just Say No to More End-To End Tests

18 December 2017

PACT
FOUNDATION

Daniel Pittner



@dpittner84



Agenda

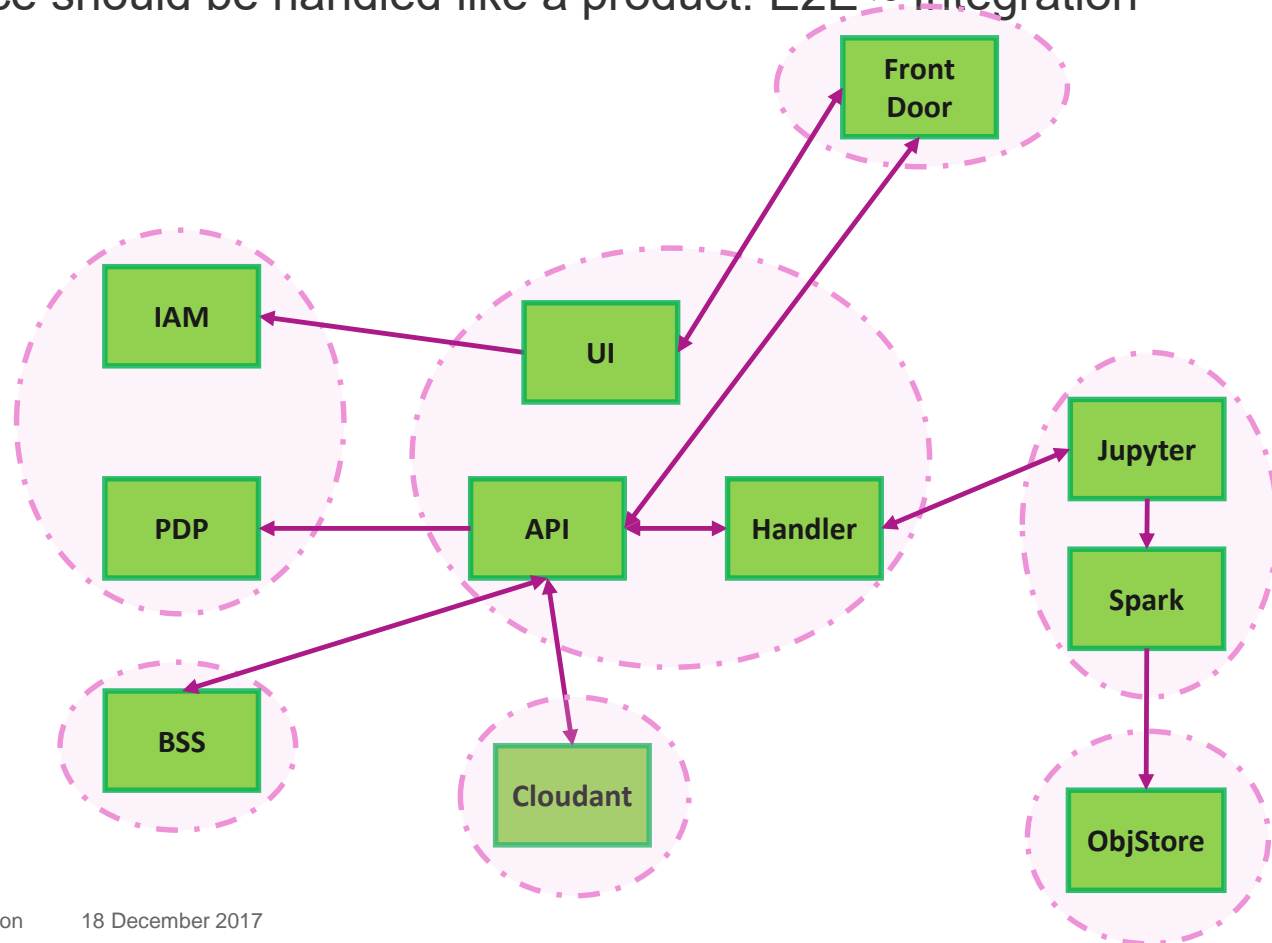
- Introduction
- Integrated tests and associated problems
- Consumer-Driven Contracts with Pact.io
- Continuous Integration
- Takeaways

„I use the term ***integrated test*** to mean any test whose result (pass or fail) depends on the correctness of the implementation of more than one piece of non-trivial behavior. (*JB Rainsberger*)

SVT CVT
 E2E Integration Test

Recap of Microservices

- Scope to specific business domain → Architecture
- Single Team responsibility (You build it you run it) → Organization
- What determines the scope of E2E? → Arbitrary
- Each service should be handled like a product: E2E ≈ Integration

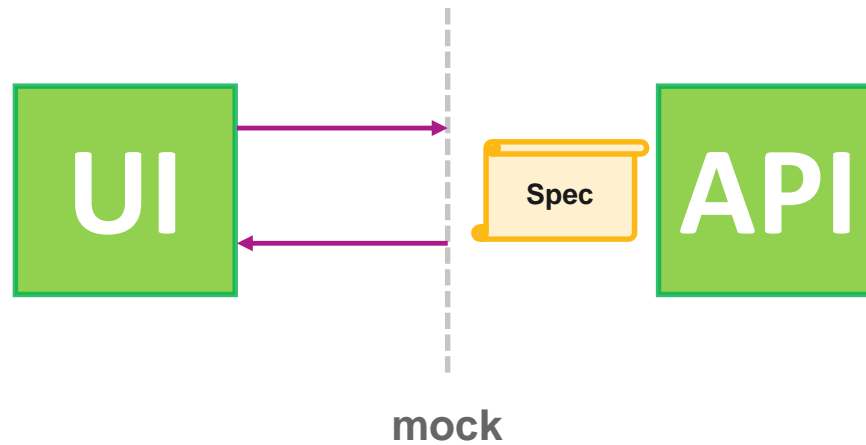


Integrated Tests: The Problems

- **Slow**
- **Easy to break**
- **Hard to fix**
- **Coordination** – How do you ensure the correct versions of „*all the things*“
- **Game theory suggests wrong incentives for desired outcome**
- **Scales combinatorically**

3 Services, 4 endpoints each, 2 paths for each endpoint
 $(2 * 4)^3 = 512$ *variants*

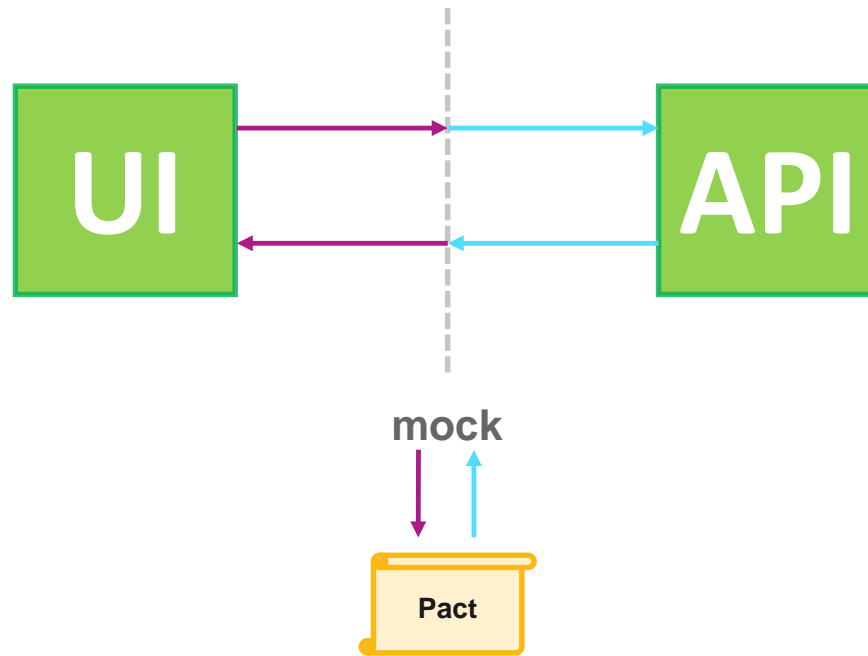
Stubs... And why they are insufficient



Consumer-Driven Contracts: A Service Evolution Pattern

- Specification by Example
 - Closed and Complete
 - Multiple
 - Bounded stability
- Driven by service consumers
- Better incentivisation model
- Follow Postells law

Test Symmetry... How it works



Fast

Stable

Easy to debug

Reliable

HTTP only

Pact

Scales linearly

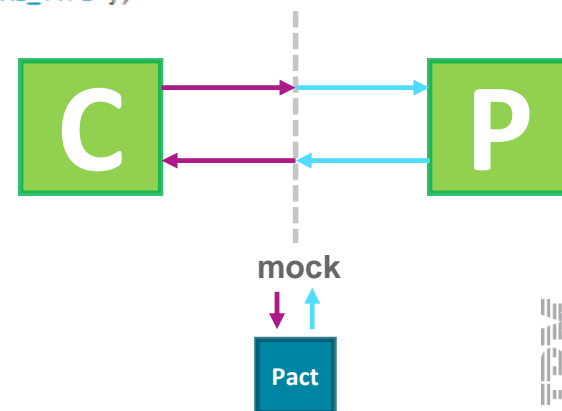
Setup
required

Independently shippable

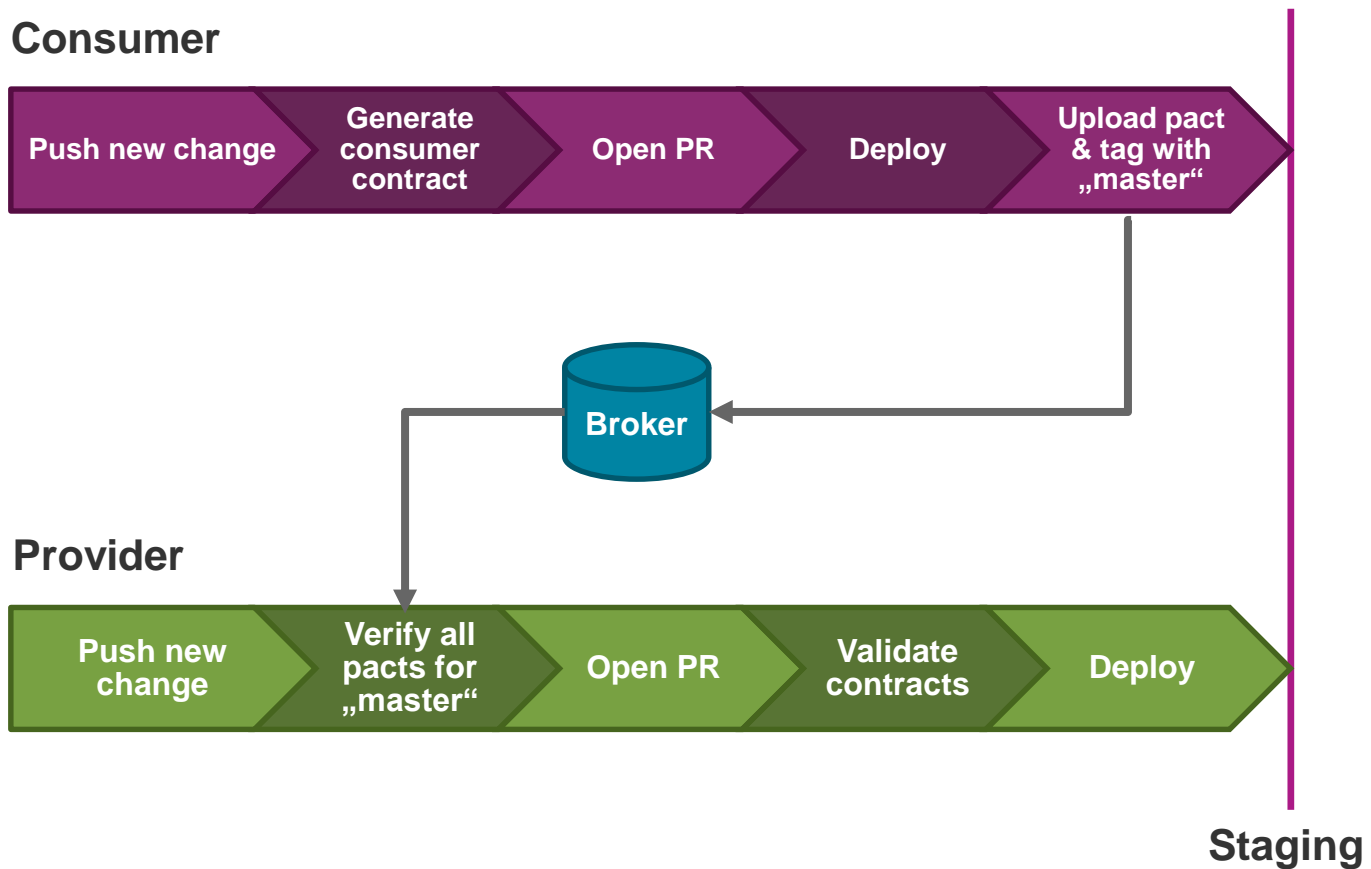
No extra infrastructure

Consumer Side Example

```
beforeAll((done) => {
  provider.addInteraction({
    state: "I have a valid statement",
    uponReceiving: "a request to create a job",
    withRequest: {
      method: "POST",
      path: "/v2/sql_jobs",
      headers: {
        "Content-Type": "application/json",
        "Accept": "application/json",
        "Authorization": `Bearer ${ACCESS_TOKEN}`
      },
      body: {
        statement: like("select * from cos://user:pass@endpoint/bucket/input.csv"),
        output: like("cos://user:pass@endpoint/bucket/output.csv")
      }
    },
    willRespondWith: {
      status: 200,
      headers: {
        "Content-Type": term({ matcher: MIME_TYPE_REGEX, generate: MIME_TYPE })
      },
      body: {
        job_id: term({ matcher: GUID_REGEX, generate: GUID })
      }
    }
  }).then(() => {
    done();
  });
});
```



Continuous Integration with Pact



What we learned so far

- Communication is still key
- Think of Postel's law (Robustness principle)
- Provider side setup requires one time effort
- Organically growing adoption
- Didn't break a single consumer since we started

Links and References

References

Introduction to Pact:

<https://docs.pact.io/>

Consumer-Driven Contracts:

<https://martinfowler.com/articles/consumerDrivenContracts.html>

Pact.io Convinceme:

<https://docs.pact.io/faq/convinceme.html>

Talk from Atlassian:

<https://www.youtube.com/watch?v=-6x6XBDf9sQ&feature=youtu.be>

Just Say no to More End-to-End Tests

<https://testing.googleblog.com/2015/04/just-say-no-to-more-end-to-end-tests.html>

Soundcloud engineering talk:

<https://speakerdeck.com/alonpeer/move-fast-and-consumer-driven-contract-test-things>