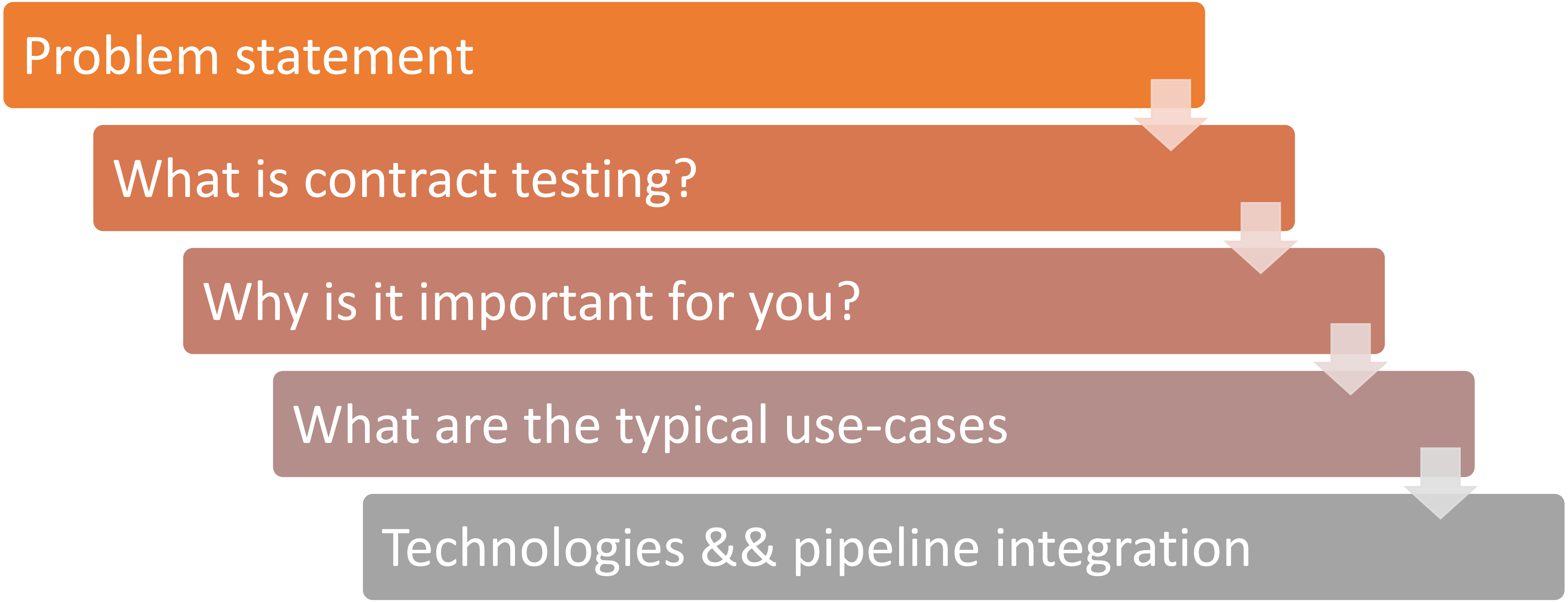
Contract Testing – An efficient way for integration testing

Matyas Zoltanka

HUSTEF

HUNGARIAN SOFTWARE TESTING FORUM

Purpose of the talk



Problem statement

What is contract testing?

Why is it important for you?

What are the typical use-cases

Technologies && pipeline integration

Problem statement

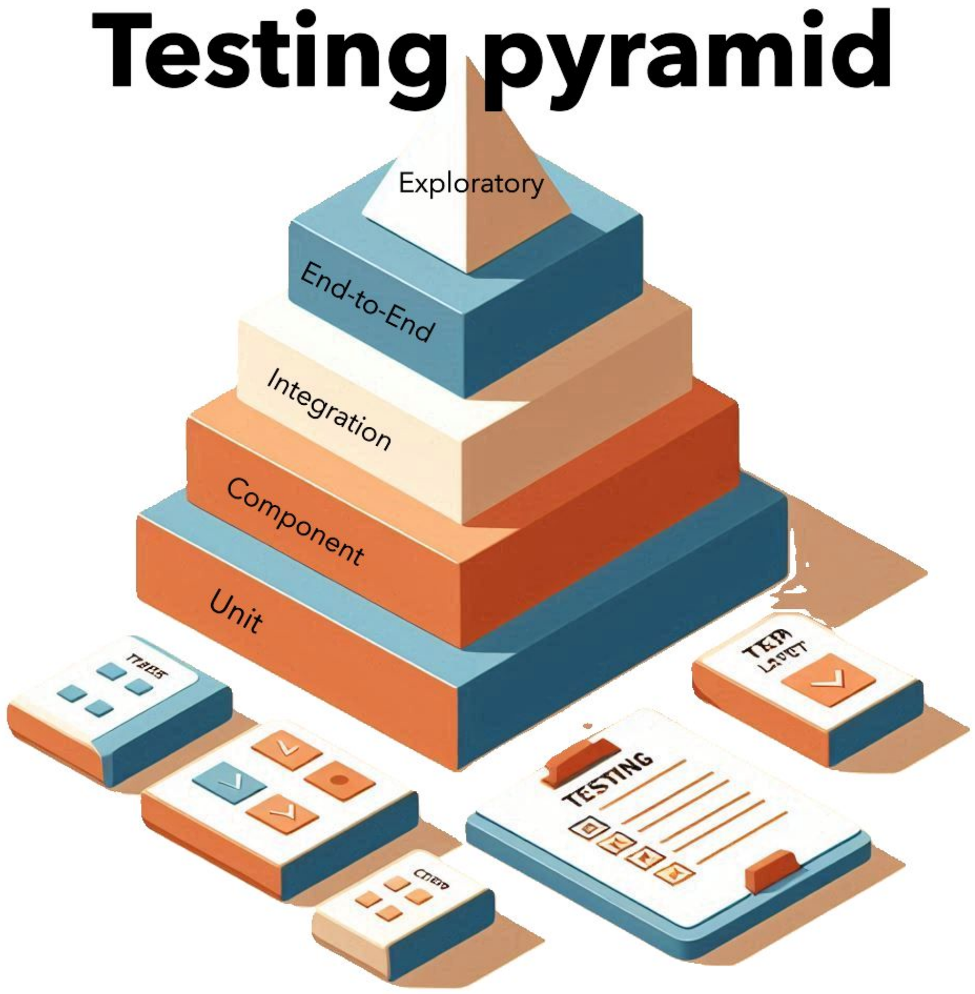
- Automated integration testing would improve most of the projects

- But it's rarely implemented. Why?
  - Complex
  - Fragile
  - Not trivial to implement

- Is there a way to do it easier: **Yes, there is!**

Key details - contract testing

- Purpose: Ensures inter-system compatibility
- Testing level: Integration
- Types
  - Consumer driven
  - Producer Driven



What is contract testing?

Technique for ensuring component or system compatibility

Key concepts:

Consumer: system/component that consumes data from the provider

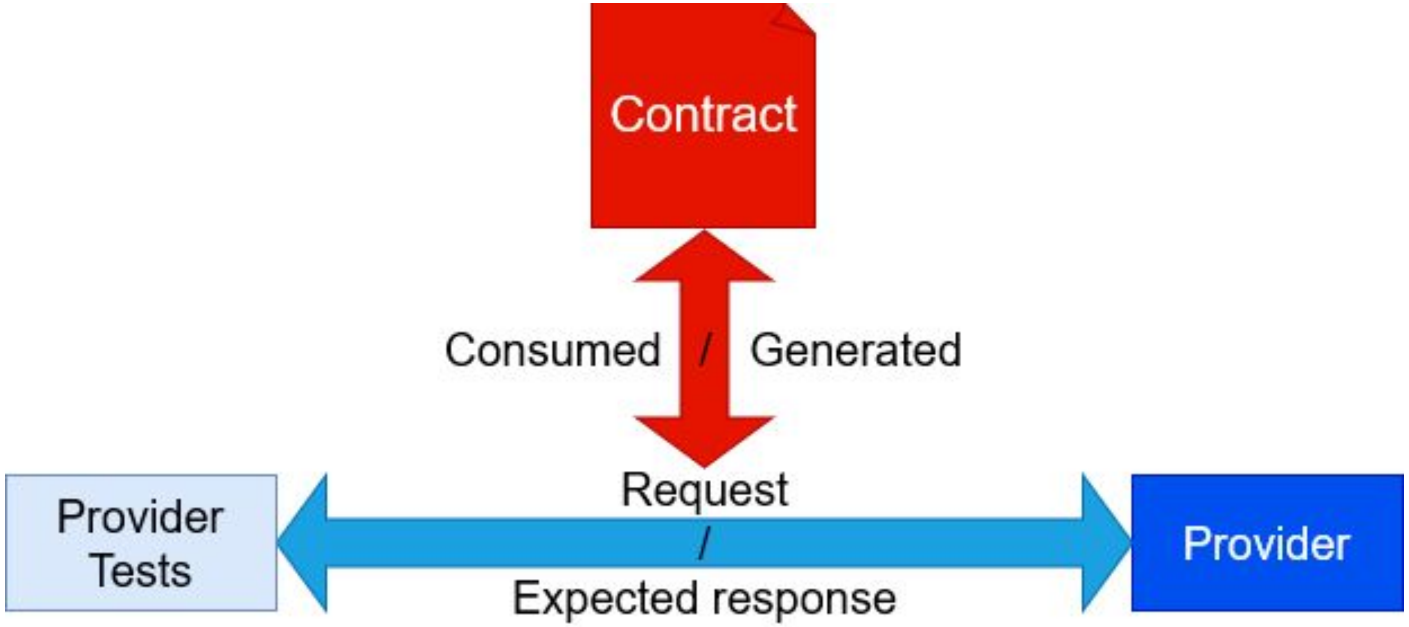
Contract: defined agreement that specifies the expected behavior

Provider: system/component that provides the data

Provider-Driven contract testing

Contract is defined by service provider

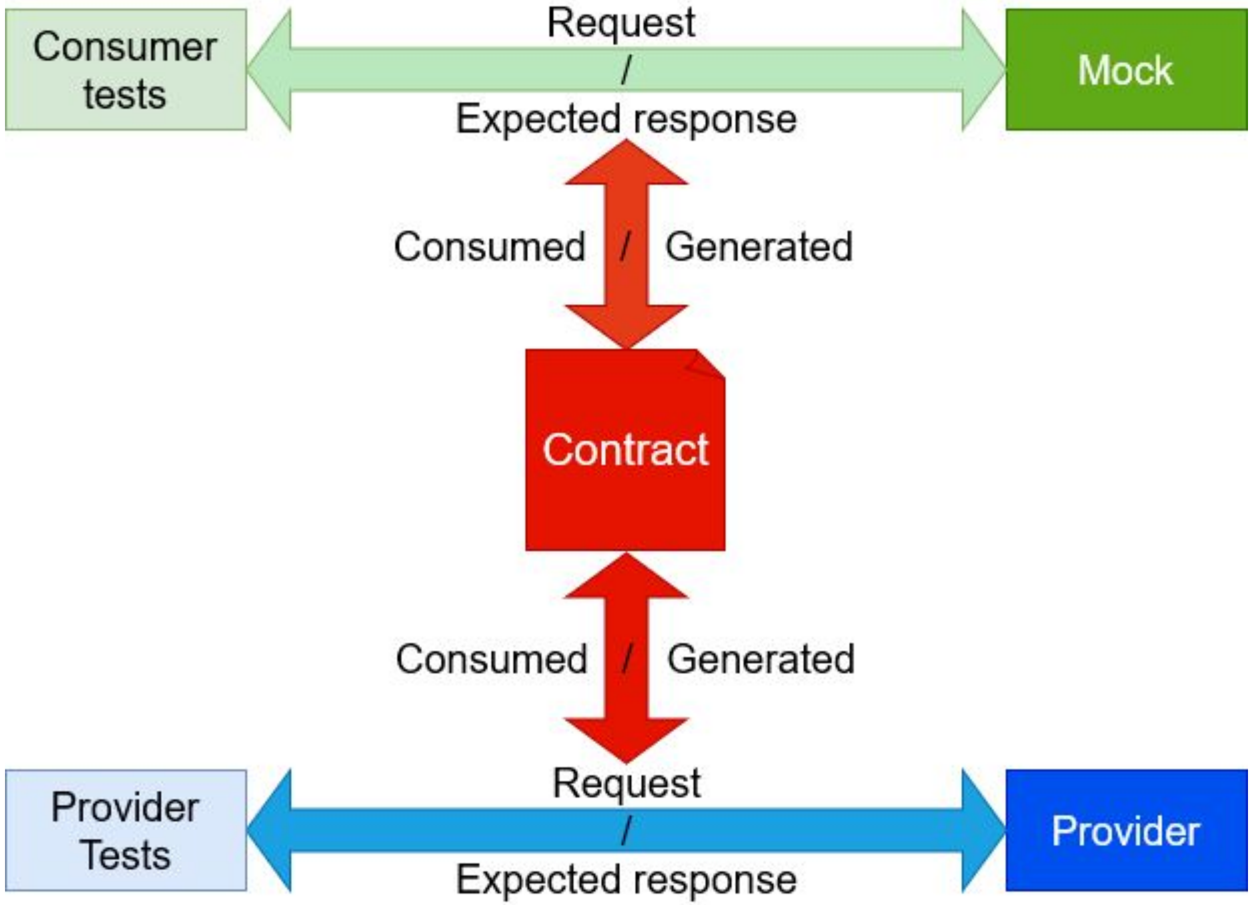
Contract and tests sent to consumer



Consumer-Driven contract testing

Contract is defined at consumer

Contract and tests sent to producer



Consumer tests

Request / Expected response

Mock

Consumed / Generated

Contract

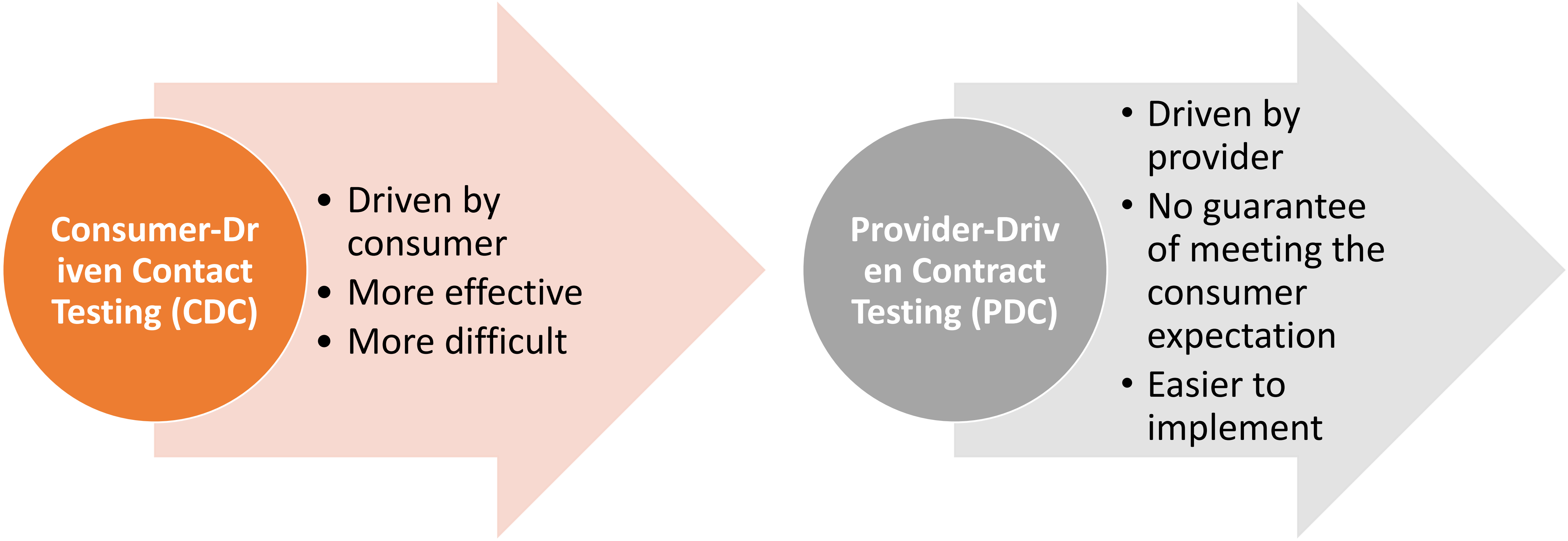
Consumed / Generated

Provider Tests

Request / Expected response

Provider

Comparison of contract testing types



Consumer-Driven Contact Testing (CDC)

- Driven by consumer
- More effective
- More difficult

Provider-Driven Contract Testing (PDC)

- Driven by provider
- No guarantee of meeting the consumer expectation
- Easier to implement

Key benefits of contract testing

1. Local execution

No generic test account, deployed system is needed

2. Improved stability and velocity

No environment / infrastructure related problems, no remote communication

3. More maintainable end-product

Easy debug, fast feedback regarding changes.

Typical contract testing use cases

- Microservice integration
- Collaboration for coupled systems
- To overcome infrastructure/ environmental challenges

Technologies for contract testing

- Pact
- Spring Cloud Contract
- Wiremock

Pipeline integration

Easy pipeline

No system deployment, generic test account required

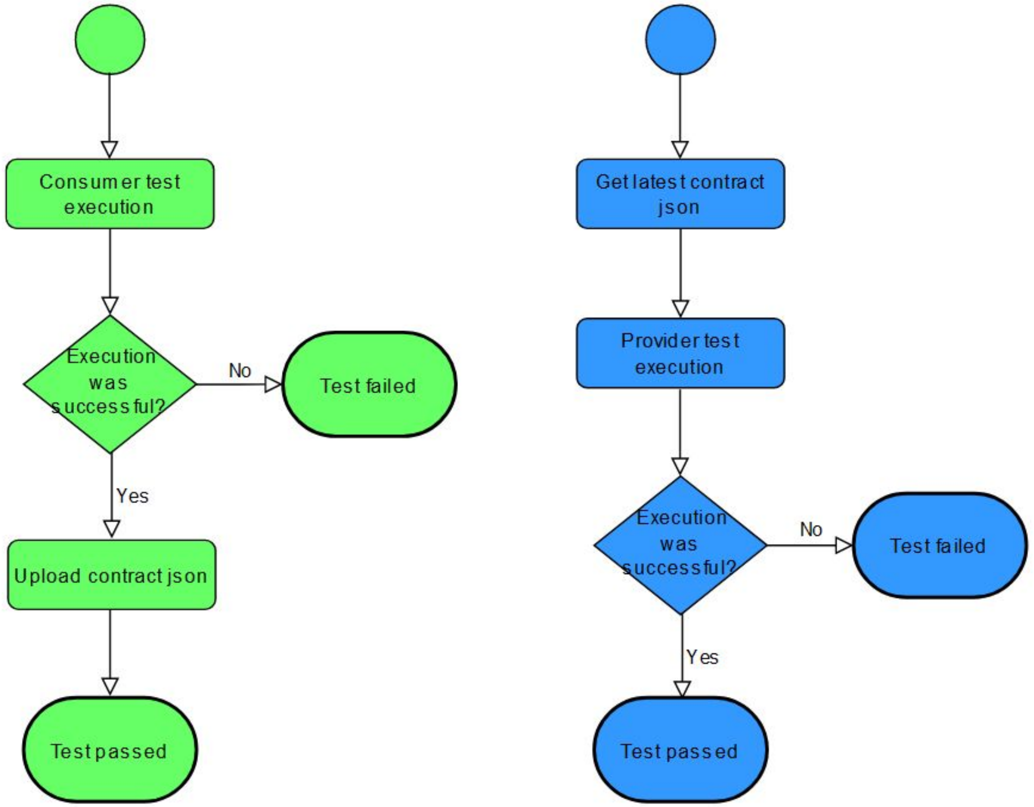
Stable execution

Local execution

Steps can be generalized

Similar steps

CI/CD integration within days



Summary

CONTRACT TESTING - ENSURES INTER-SYSTEM COMPATIBILITY

KEY BENEFITS - MS INTEGRATION, COLLABORATION

TECHNOLOGY && PIPELINE -MULTIPLE TOOL, PIPELINE INTEGRATION WITHIN DAYS