

Consumer-Driven Contract Testing

Studiengang: BSc in Informatik | Vertiefung: Digital Business Systems
Betreuer: Prof. Dr. Stephan Fischli
Experte: Dr. Federico Flueckiger
Industriepartner: Die Schweizerische Post AG, Bern

Nach dem monolithischen Ansatz ist mit der verteilten Architektur die Kommunikation zwischen einzelnen Systemen zu einem elementaren Bestandteil der Softwareentwicklung geworden. End-to-End Tests testen eine Systemlandschaft und stellen damit auch das Zusammenspiel der einzelnen Komponenten sicher. Sie haben jedoch einige gravierende Nachteile. Consumer-Driven Contract Testing hat zum Ziel, die Interaktion der Systeme isoliert zu testen und damit die Nachteile zu reduzieren.

Ziele

Die in der vorangegangenen Projektarbeit erarbeitete Theorie zu Consumer-Driven Contract Testing wird in der Praxis bestätigt. Anschliessend wird eine Empfehlung für die Integration und den Einsatz dieser Testmethodik für die Schweizerische Post erarbeitet.

Methode

- Vorbereitung: Um herauszufinden, ob Consumer-Driven Contract Testing sein Potential gegenüber End-to-End Testing ausschöpfen kann und ob sich der Einsatz lohnt, wird eine Beispielumgebung angelehnt an die Softwarearchitektur der Schweizerischen Post aufgebaut.
- Simulation: Mit dem De-facto-Standard Tool Pact wird Contract Testing für die Kommunikation über eine REST-API und über Apache Kafka umgesetzt und einige vordefinierte Szenarien simuliert.
- Evaluation: Anschliessend wird basierend auf den gewonnenen Erfahrungen die Integration der Testmethodik für verschiedene Anforderungen evaluiert.

Ergebnis

Consumer-Driven Contract Testing ist mächtiger als die bereits eingesetzte, von Providerseite definierte OpenAPI-Spezifikation und verhindert die Verteilung von zueinander inkompatibler Software.

Während der Einstieg in die im Detail komplexe Thematik nicht einfach ist, kann Pact danach seine Vorteile gegenüber End-to-End Tests voll ausspielen. Allerdings beschränkt sich Contract Testing auf das Testen von Kommunikationskanälen, so dass die Testmethodik nicht alle End-to-End Tests ersetzen kann. Die Erfahrungen mit der PactNet Library sind enttäuschend. Die offizielle Dokumentation ist unzureichend, und die Beispiele sind zu wenig tiefgreifend. Des Weiteren sind aufgrund einer grösseren Änderung in der Library die wenigen vorhandenen unabhängigen Beispielprojekte veraltet.

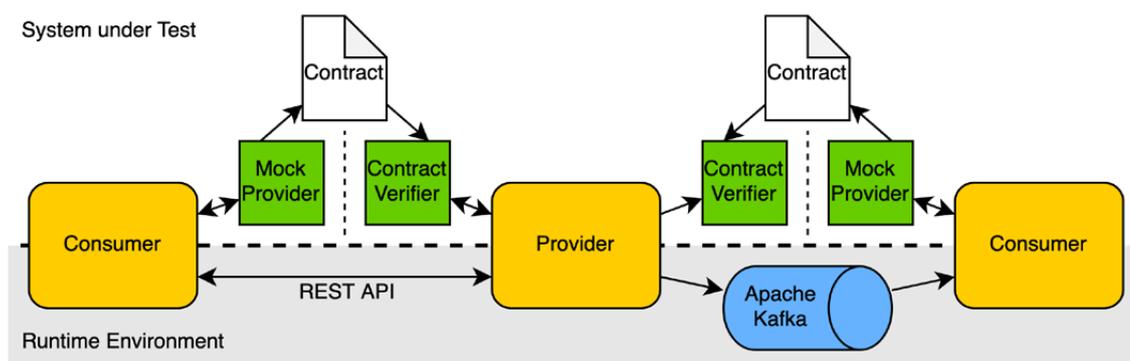
Im Gegensatz zur .NET Implementierung bieten die Libraries für Java und JavaScript mehr Funktionen, eine bessere Dokumentation mit umfangreicheren Beispielen und werden besser gepflegt.

Schlussfolgerung

Die Integration von Contract Testing für REST-APIs ist unter den aktuellen Bedingungen bei den gegebenen Systemen der Schweizerischen Post wenig sinnvoll. Auch von einer Nutzung über Messaging-Systeme wird aufgrund der Unreife von PactNet derzeit abgeraten. Das Projekt soll aber weiter beobachtet werden und bei einer Verbesserung der Situation zumindest für kritische Systeme in Betracht gezogen werden.



Roman Patrick Stocker
roman.stocker@gmx.ch



Architektur Beispielumgebung