A Reusable Automated Acceptance Testing Architecture for Microservices in Behavior-Driven Development

Type: Research-oriented
Tejaswini Koduri
Shilpa Avvaru

CSC 8350: Advanced Software Engineering
Department of Computer Science, Georgia State University
Spring, 2017

Abstract:
Cloud Computing and Mobile Cloud Computing are reshaping the way applications are being developed and deployed due to their unique needs such as massive scalability, guaranteed fault tolerance, near zero downtime, etc. and also daunting challenges such as security, reliability, continuous deployment and update capability. Microservices architecture, where application is composed of a set of independently deployable services, is increasingly becoming popular due to its capability to address most of these needs and challenges. In recent years, the Behavior-Driven Development (BDD) has become one of the most popular agile software development processes, and frequently used in microservices development. The key to success of BDD is the executable acceptance tests that describe the expected behavior of a feature and its acceptance criteria in the form of scenarios using simple and business people readable syntax. The reusability, auditability, and maintainability become some of the major concerns when BDD test framework is applied for each microservice repository and no previous research addresses these concerns. In this paper, we present a reusable automated acceptance testing architecture to address all these concerns.

Fewer features can be found in Client-Server and Mobile Agents, yet both architectures can be useful in applications which do not require scalability, resilience, service integration and other features found in either SOA or microservices. All architectures in fact are used for particular services types depending on their requirements priorities. Finally, when it comes to fault tolerance and low resources cost, microservices architecture only satisfies many of these features. However, they are only achievable by Client-Server, SOA and Mobile Agents when configured in ways supporting this matter. For fault tolerance, applications or services can be developed in order to overcome any faults by relying on more hardware resources. If fault tolerance was not as important as resources cost, then less resources can be used in which the application is built with minimum resources needed. So it is always a matter of priorities to be found in each application developed. Microservices architecture is not said to be the best among the others, but it actually can fulfil more features at once compared to the previously proposed architectures.

It is not about following what is most modern and trendy, it is important to investigate whether the application would fully function efficiently using the selected architecture. Microservices advantages and drawbacks were illustrated to show whether the developed application would need that amount of scalability, integrity, resilience and agility with the payoffs of known and unknown consequences. A comparison was done in which the four illustrated architectures were compared in terms of the most desired features to be found in this modern time.
Main References:

Other Reference (Optional)