**CSCI 391 – ST: Microservices, Quiz 5 – Implementing Microservice Communication.**

Name:

Student Id:

*For questions 1 & 2, select the type of contract breakage that is possible for each scenario.*

1. Fields or methods being removed, or new required fields being added to a microservice endpoint.
   1. Semantic Breakage.
   2. Structural Breakage.
2. The behavior of a microservice endpoint changes in such a way as to break consumers’ expectations.
   1. Semantic Breakage.
   2. Structural Breakage.
3. What key idea helps avoid breaking changes by implementing a client (reader) that can ignore changes that are not relevant?

Tolerant Reader.

1. If a piece of software uses semantic versioning and its version is incremented from 1.0.0 to 1.1.0 in the next release, what changes can I infer have occurred based on the rules of semantic versioning?
   1. New functionality has been added that should be backwards compatible.
   2. Backwards incompatible changes have been made.
   3. Bug fixes have been made to existing functionality.
2. Which option for managing the rollout of breaking changes to a microservice contradicts the principle of independent deployability, as it requires both the microservice and all of its consumers to be updated simultaneously?
   1. Emulate the old interface.
   2. Coexist incompatible microservice versions.
   3. Lockstep deployment.

*Extra Credit:*

Due to the TTL (Time to Live) of DNS records and their frequent caching in multiple locations, which technology can help prevent referencing outdated IP addresses of a microservice instance, and allow us to refer to this technology instead of individual microservice IPs in our DNS records?

Load Balancer.