

Tolerant Reader

You are building an application within a [Microservices Architecture](#) and you must deal with the reality of evolving microservice APIs

How do I make sure that a new release of a microservice API does not trigger a cascade of upgrades in the service's consumers?

If you place all of the responsibility of API change on the service itself, then you may find yourself in the situation of having to support multiple versions of a microservice at the same time. Likewise, if you try to notify each microservice client application of changes in an API, then you may have to go with a relatively heavy-weight Service Registry approach that requires that all clients look up your microservice through a service registry that only returns exact matches for version number requests.

Therefore,

Encourage your microservice consumers to be Tolerant Readers: they just pick out the information from the JSON they are interested in; as most API changes are additions of data, such consumers can happily ignore such changes.

Tolerant readers can benefit from version information in your JSON, whether they receive it through an API or through *Topic Messaging*. An easy solution is to have two monotonically increasing version numbers:

- A `featureVersion` that is increased any time that things are added in a backwards-compatible way.
- A `compatibilityVersion` that is increased any time that things are changed or removed, breaking backwards compatibility.

An application can then check whether the feature version is at least what it expects and the compatibility version is smaller than it expects. If that is the case, the data that is expected should be there and in the expected format.

The *Tolerant Reader* pattern was introduced in [Daigneau](#)