Torry Harris Business Solutions

SOA – Service Versioning Best Practices

Table of Contents

1. Executive Summary

This document aims at highlighting some of the best practices related to Service Versioning in SOA. These best practices are based on practical problems that Torry Harris faced when implementing SOA projects.

Versioning is one of the important aspects of SOA Governance. Few governance strategies recommend having one baselined version of the service to avoid versioning complexity. On the other hand, few other governance strategies prefer using multiple versions of the same service so that changes and enhancements to service interface do not affect existing consumers. This document aims to highlight the pros and cons of each approach and proposes best practices that suit both the cases.

Actual technical solution for implementing service versioning is considered as out of scope of this document as there are several simple and vendor-specific approaches. The focus of this document is to describe the principle of Service versioning and highlight the best practices.

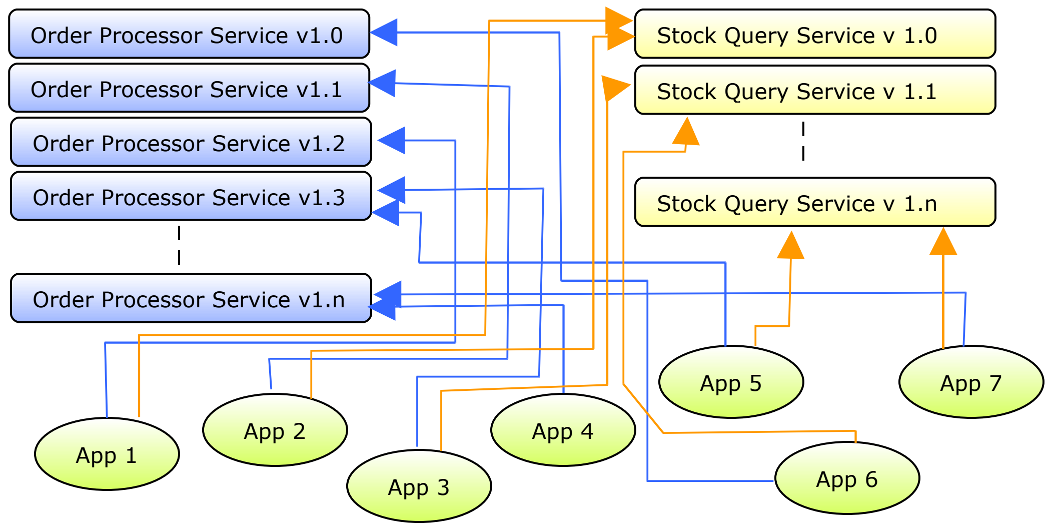
2. Why Versioning?

Service Versioning Overview

Service Versioning is the approach followed by Service developers to allow multiple versions of the same service to be operational at the same time. To give an analogy, any re-usable software API library has multiple versions used by different applications. The same analogy applies to services.

Service Versioning Example

Web Services are the ideal way to implement SOA services. The following diagram illustrates the concept of multiple active versions with a view of the services and the dependency with its consumer applications. In the example, there are two services - Order Processor Service and Stock Query Service. Multiple active versions exist for both these services. The diagram shows the dependency path of 7 different applications using several versions of these two services.



Order Processor Service v1.0
Order Processor Service v1.1
Order Processor Service v1.2
Order Processor Service v1.3
Order Processor Service v1.n
Stock Query Service v 1.0
Stock Query Service v 1.1
Stock Query Service v 1.n
App 1
App 2
App 3
App 4
App 5
App 6
App 7

Pros and Cons

Pros	Cons
Changes and enhancements can be made to individual services and released as new versions without causing an impact on the existing consumer applications	As shown in the diagram above, multiple versions can quickly lead to lot of dependency management hassles, thereby increasing complexity in managing and tracking them
In an enterprise, consumer applications would typically be developed and maintained by different teams. Multiple versioning gives the flexibility to the team to prioritize and migrate to the latest version according to their convenient schedule and budget.	Any bug-fixes/errors discovered at a later point of time in services would need appropriate fixing in all applicable versions of the service. This eventually leads to poor maintainability of the service code
There is always an easy rollback plan in place when an application faces an issue with the new version. It can just continue interacting with an earlier stable version.	Custom solution would need to be followed in most cases, requiring to maintenance of several versioned copies of the WSDL and schema.
	Additional tool / runtime registry is required to fetch the appropriate endpoint URL based on the version number
	Source code of the services would need to be carefully controlled using branching so that multiple versions of binaries are generated and maintained appropriately.

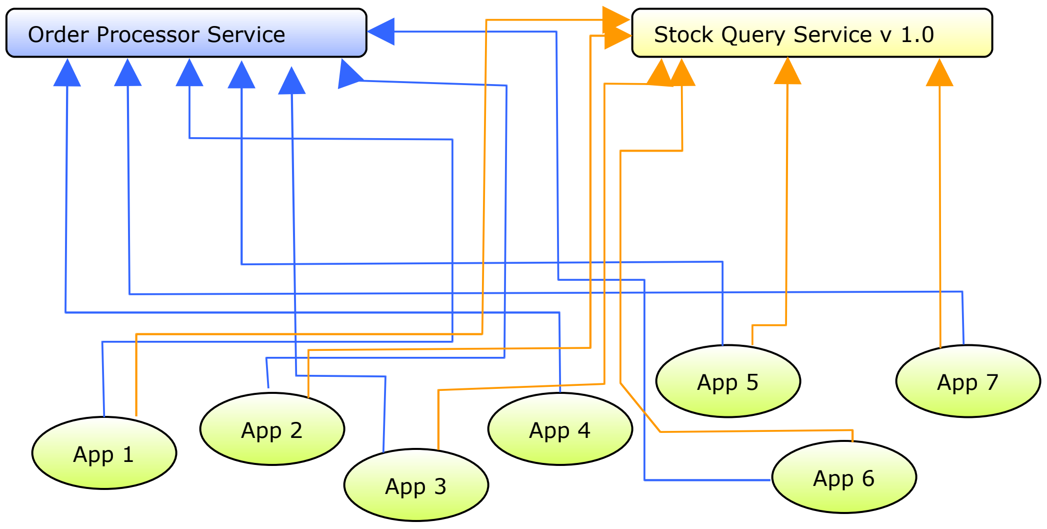
3. Baselined Services – No Versioning

Baselined Services Overview

The concept of Baselined Services discourages the use of versioning. Only one finalized version of Service is kept active at any point of time, and all consumer applications points to one and only version of the service, which is the baselined version.

Baselined Services Example

The following diagram (adapted from the previous) example illustrates the concept of baselined services.



Pros and Cons

Pros	Cons
The Service Platform represents one single view of the enterprise service portfolio, thereby ensuring "re-usability" in its true sense.	This policy is "too rigid" on several application consumer teams, where in every change/enhancement to the service would require a bit of migration work within the application
Maintenance is greatly simplified	Design of the service would need to be considered with great care, ensuring forward and backward compatibility. This could be a limiting factor in some cases for the business teams to plan for major service enhancements
A runtime registry is not needed as	Impact analysis procedure should be

consumer applications maintain a fixed endpoint URL	strengthened so that changes are implemented with great accuracy
Source code management of the services is greatly simplified as there is only one version to maintain	Live deployment Rollbacks should be planned carefully so that there is minimal impact on the consumer application.

4. Best Practices

Having described both approaches, and their pros/cons listed, it becomes very difficult for enterprises to choose a particular approach. The pros of the versioned services approach seems to be ideal compared to the cons of baselined services approach. Hence, the best practice recommendations for versioning governance strategy are

- Use a mix-and-match of both worlds by following versioned services approach, yet having control on the versioning nightmares by limiting the maximum active versions to 3

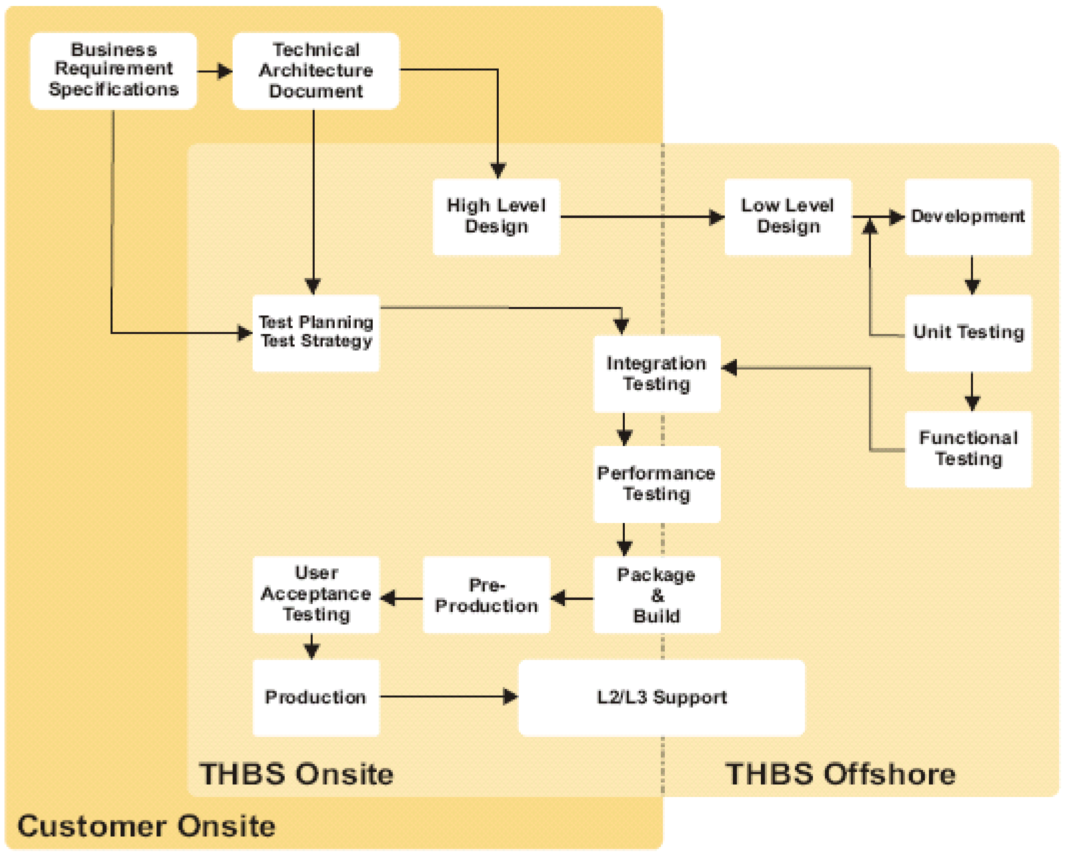
- Upon launching a new version of the service, only the last two continue to remain active. All the lower versions should be deprecated and de-commissioned.

- This means, no more than three active versions of a service exist at any point of time. Policies needs to be established and communicated to the consumer application teams to ensure that migration is done on time

- The services team will not be responsible for impacts to the consumer application if the application continues to use a deprecated version of the service.

Torry Harris SOA engagement

Onsite Offshore Project Flow



Torry Harris Business Solutions (THBS) is a US based IT service provider with development facilities in India and China. The company, started in 1998, has for several years delivered a large variety of middleware services to enterprise clients around the world. Now, with a large pool of highly skilled technologists and rapidly growing, the company remains focused on the middleware and integration space, implementing large projects across the US, Europe, the Middle East and the Far East. The company is committed to Service-oriented Architecture(SOA), which it sees as the logical movement to follow the phenomenon of distributed computing in the late nineties, where THBS was clearly the market leader in implementing the offshore/onsite delivery model.

For more information, write to us at soa@thbs.com.