Service Evolution

Software development does not end for a custom software system until it is replaced and/or decommissioned. A successful software system will be used for many years, and will constantly be enhanced with new features and functionality during those years. In other words, there really is no such thing as "maintenance mode" for a software system, it is really "ongoing software development mode" with an occasional bug fix added on the side.

System evolution is the practice of designing a system's architecture and code to make the constant addition of enhancements to a system as fast, easy, reliable and inexpensive as possible. The DGP architecture enables continuous system evolution, in which all components of the system can be frequently enhanced without ever breaking backward compatibility for any client applications or integrated systems. This is made possible by the modularity of the architecture, the immutable append-only conventions used for those modules/tiers, and the isolation/loose coupling between the tiers and components of the architecture. In particular, the single argument front controller of the message-based API's enables the loose coupling by encapsulating all of the variability between different request and response messages within the body of the messages themselves.

Evolution Verification

1. *Almost all aspects of a DGP system follow an immutable append-only convention, and this includes all API methods as well as the test harness files that test those API methods. In practice this means that API methods cannot be changed once they have been deployed to production environments (excluding bug fixes), and that also means that the test files which test those API methods never need to be maintained or changed from that point forward either. As a result, a DGP system becomes a growing collection of immutable API methods and test files that require no maintenance work. All development work is focused on building new API methods and test files to add to the collection.*
2. *Since all possible logic in a system is consolidated in the middle tier as web service API methods, the full regression which runs all test files for every API method as part of each deployment is the mechanism used to verify that no breaking changes to any existing API methods have occurred as part of each new deployment.*
3. *The immutable append-only conventions also apply to all of the database schemas and in some cases the data itself.*