

Ensuring Operational Integrity of Web Services and Services-Oriented-Architecture (SOA) Applications

Know Where Your Business Stands
– Anytime, Anyplace



Keeping Your Promises.

Ensuring Operational Integrity of Web Services and SOA Applications

Wrapping existing applications as web services and building new web services is the key trend in applications development and deployment today. The benefits of building new applications to this new architecture:

- Creating standard interfaces to a variety of heterogeneous applications
- Having the ability to reuse these applications via their standard interfaces without having to do point-to-point integration

However, with advances in how applications are built and deployed, the ability to create applications to a new paradigm has arrived ahead of the ability to effectively manage these applications in production. For example, J2EE has achieved critical mass as a deployment platform for custom, mission-critical, web-facing applications, but there is an emerging need for new types of products to enable enterprises to effectively manage these new types of applications *in production*.

The New Challenges of Managing Web Services and SOA in Production

There are significant differences between applications implemented as Web Services (or to SOA) and previous more monolithic applications:

- These new applications are much more likely to be compound applications – there are, in fact, a number of legacy applications combined and integrated via SOA into a Web Service.
- Compound applications are much more complex than even the sum of their parts, since a single transaction now has to flow through multiple applications (and their supporting infrastructure).
- At the time that you build a new SOA application, or Web Service, you may not know what other applications will use the new Web Service, nor will know what the other applications will be using your new Web Service for (the precise use cases).
- You may not have control of who will use your web service, both in terms of internal and external consumers. Therefore the convenience of the

standardized interface to your Web Service creates an exponentially greater opportunity for abuse of your new Web Service.

- Just as some of the consumers of your Web Service may in fact be external to your group or your company, you may find yourself relying upon Web Services that are external to your group or your company to make your Web Service able to do its job.
- Web Services are by their nature loosely coupled applications, so the notions of “keep-alive” mechanisms that have been used in tightly coupled applications for years to allow one component to ensure the availability of the other are simply not applicable.

Bottom-line – with the dramatic increase in flexibility that Web Services and SOA brings to the task of application integration, with it is brought a dramatic reduction in anyone’s ability to control a compound application from end-to-end, as well as a dramatic increase in the possibility for misuse or abuse of web services by “outsiders.”

The Need For Continuous, Production Verification of Web Services

Across the spectrum of systems management offerings available today, there is still the critical, unmet need for the ***continuous verification of web services, IN PRODUCTION.***

Specifically, the need consists of the ability to:

- Know that one’s own Web Services are available at the start of the day before customers and partners applications start invoking transactions against those Web Services
- Know that a partner’s Web Service is available before ones own production Web Services start issuing transactions against them
- Test and validate the performance of ones own and partner Web Services throughout the day to ensure that business transactions are being processed quickly
- Alert and report on outages and brownouts (performance issues)

Use Case Analysis

Most of the major vendors have put features into their mainline Application Performance Management (APM) products to monitor the operation of SOA applications in production. These features, modules and products target their traditional audiences that keep these applications running in production. However, the largely unmet need is to verify end-to-end business transactions for SOA-based applications.

What is meant by 'end-to-end business transactions' for SOA-based applications?

Use case – A mortgage loan application has an SOA interface for a credit scoring agency to retrieve credit scores when the mortgage application passes the credit scoring application the Social Security number and the other data required for authentication. APM solutions from the major monitoring vendors can tell if the mortgage loan application is running, what resources it is using, if DB queries are slow, and what the response time to the transactions are. They can also, for the most part, tell if the interface to the credit scoring application is where the slowdown in the response time is occurring. If all you have is one application and one SOA interface, then traditional APM products will simply tell you that the SOA interface to the credit scoring application is the problem when it becomes a problem. This allows the IT team to go fix the right part of the problem AFTER the problem has occurred.

Where this model falls down is in the case of many compound applications with many SOA interfaces. Let's assume a case of 60 SOA interfaces, with a total transaction flow of 120 transactions through the system in a hour. So, on average, there is a transaction through an interface once every 30 seconds, or 1440 every 8-hour day. These transactions are not distributed uniformly across the interfaces at any moment in time, nor is the distribution consistent across time. In other words, the load is variable and maybe even random on a per interface basis. Now, let's assume that a small percentage of these transactions fail every day due to issues with the systems behind an SOA interface. One percent is 14 failures a day. Two percent is 29 failures a

day. Reacting to this number of failures after they have occurred just does not work either for the IT organization or the business users.

What is needed in this case is continuous verification of each individual SOA-based component of the compound application via synthetic transactions. For compound applications with this many interfaces, this can only be accomplished utilizing a 'synthetic-transactions model'.

The reason is that the only way for the IT team (and the business stakeholders) to stay on top of such a complex compound application is to verify it on a per component basis with regularity. This allows issues to be addressed before they impact actual production transactions, and is the most effective way to be able to find out about these kinds of problems early enough so that something can be done about them (given that many of the applications that live behind an SOA interface might in fact be third party applications outside of the direct control of the IT team supporting the application).

SOA-based compound applications have different end-to-end monitoring requirements than do single function applications with known execution paths for all business transactions. Due to the complexity of SOA-based applications, and the numerous (exponentially growing) paths that users and business transactions can take through the compound application system, an approach that proactively monitors each of the external interfaces for such a system is required. Such an approach must also be able to rapidly add support for new interfaces, incorporate changes to existing interfaces, and deal with the inherently distributed (across organizational boundaries in many cases) nature of compound applications systems. Reveille EPM uniquely meets these requirements due to its ability to easily and proactively monitor the distributed and rapidly changing interfaces that comprise these systems.

About Reveille

Reveille Software™ is the provider of the most widely used experience and performance management (EPM) solution for enterprise content management (ECM) applications. Used by more than 270 companies, including GSK, T-Mobile and Chase, Reveille EPM helps companies ensure business-critical applications perform at peak efficiency and availability – without application babysitting. Reveille EPM proactively monitors user experience, application processes and business metrics to diagnose and repair failures before end-users experience delay.



Keeping Your Promises.

3820 Mansell Road, Suite T-20
Alpharetta, GA 30022
t. 770-642-2727
f. 770-642-2726
reveille.sales@reveillesoftware.com
www.reveillesoftware.com