

Availability and the Rise of the Mashup



Lori MacVittie, 2008-04-01

[Google](#). [Amazon](#). [Facebook](#). [LinkedIn](#). [Salesforce.com](#). While certainly not an all inclusive list, these very recognizable web monsters all offer access to their "platforms" via a web-based API, a.k.a. services. With the notable exception of [Salesforce.com](#), most have implemented these services as a REST (Representational State Transfer) or REST-like set of interfaces, but in general these APIs meet the criteria necessary to be referred to as **services**. They're SOA as surely as any other service out there.

These services are being incorporated at a rapid pace into other web-based (dare I say Web 2.0) applications, and a plethora of others are emerging. No doubt we'll see even more services and applications (a.k.a. *mashups*) built from existing and emergent services in 2008.

Inter-organizational integration - because that's really what these mashups are, don't kid yourself - has always been a scary proposition for the enterprise primarily because of the reliability and availability of said services. After all, it's hard for an organization to meet agreed upon service level agreements when the reliability of third-party services is questionable.

Ron Schmelzer of [ZapThink](#) sums it up nicely in an [SD Times article](#) regarding SOA Insecurity:

"Can you imagine what would happen if Google Maps went down? How many applications would I kill?" In the past, that would have been a problem for only Google, he noted, but with SOA, the impact is so much wider. "The greatest benefit of SOA—[the ability to share services]—is also the greatest problem of SOA."

Indeed, what *would* happen? I imagine the sound of beepers and Blackberries going off across the world would deafen us all.

Though Ron raises an excellent question, he doesn't dig into any solutions. That could be because SOA inherently has *no* solution for this problem; it's a matter of architecting a scalable, reliable delivery infrastructure that can ensure high availability and that's often outside the realm of the SOA experts.

It's well within the realm of application delivery network experts, however. Isn't that, after all, one of the core competencies of an **application delivery controller** like [BIG-IP](#)? To ensure availability of services - whether they're pure SOAP, REST, or simple resources?

(Hint: The answer is yes, yes it is)

An even bigger question, however, is what happens when the data center serving those services is unavailable? It happens - routers fail, packets get lost, power outages occur. No amount of internal data center preparation can account for these scenarios - you need a secondary data center from which services can continue to be served. But without an application delivery solution that can globally manage multiple data centers, like [BIG-IP Global Traffic Manager \(GTM\)](#), consumers of such services will not automatically discover and reroute requests to that secondary site.

Mashups and the inter-organizational integration of data is not a thing of the future, it's here today - and it's growing. The reliability of your own mashups and applications may now or may soon rely upon services provided by large - or small - external providers. Ensuring that both your own applications and the services upon which those applications are dependent are available is paramount to your success. The old saying used to go "there's no such thing as bad press", but ask a few organizations that have suffered major outages in the past year whether they agree with that saying...I'm guessing their answer would be "no, no we don't."

So what can you do to ensure that availability issues don't destroy your chances of being the next big thing?

1. Ask "service" providers to elucidate their availability strategy. What delivery infrastructure do they have in place to ensure the services you'll be using are always available? Do they have a secondary data center *and* a mechanism for failing over - quickly - to that site in the event of a failure at the primary data center?
2. Architect your own delivery infrastructure such that *your* mashup/services are highly available. Implement at least a

local application delivery network that is both intelligent and can adapt to changing conditions within your data center to ensure the highest availability of your services possible.

You may not have the volume today to make you think an application delivery network is a requirement to ensure the success of your applications and mashups, but **availability** isn't about volume, it's about reliability and consistency of service. And that most certainly requires a well-architected application delivery network.

Technorati tags: [MacVittie](#), [F5](#), [BIG-IP](#), [availability](#), [mashups](#), [SOA](#), [Web 2.0](#), [application delivery](#)

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](#)

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com

©2016 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at [f5.com](#). Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113