

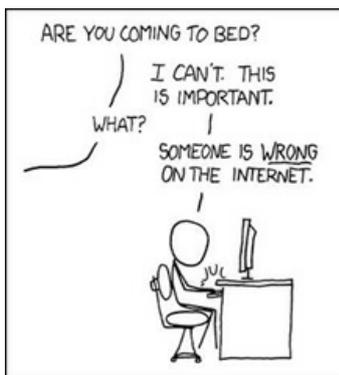
Duty Calls: Data Portability in The Cloud is an Application Integration Problem, Not a Cloud Problem



Lori MacVittie, 2009-12-10

Spectacular “cloud” failures over the past few weeks have raised the hue and cry for portability and interoperability across clouds for data. The problem is that the cry is based on the false assumption that a “cloud service” is the same as an “application service.”

Apparently [Microsoft](#) felt [Google](#) and [Amazon](#) were getting too much attention with their recent outages and decided to join the game. The absolute loss of data for ~~thousands~~ lots and lots of [T-Mobile Sidekick](#) users is regrettable and yes *someone* needs to address such issues but that *someone* is not a standards group or a committee or even Microsoft.



The problem here seems to be that people equate “cloud services” with “application services”. Sidekick, etc... is not a cloud service, it’s an application service and that’s an important distinction. Even if it were deployed in a cloud, which it is not, it would *still* be an application and not a cloud service. Yet folks continue to make this very basic and very important mistake despite the FUD that results from their inaccurate verbiage. For example, [Lauren Weinstein’s blog on Microsoft’s recent Sidekick-related data loss](#), has this to say on the subject:

Another important related risk is being “locked into” particular cloud services. Most cloud computing services make it as simple as possible to get your data into their universe. But getting your data out again can often be anything but trivial. If your data is “trapped in the cloud” and something goes wrong, it can be a very serious double whammy indeed.

Yes, users should be able to get their data out of a cloud with the same relative ease with which it went in, but we aren’t talking about cloud services we’re talking about application services. And interoperability and portability between applications has never, ever been a guarantee. E-mail is about the only exception to this rule and you can thank [RFC 822](#) for that. Unless we’re willing to sit down and write this level of detail for *every application* known to man and every *application* that does not yet exist, consider e-mail *data* interoperability a fluke of nature and thank the powers that be that we have that much. No, HTML and HTTP don’t count because they don’t actually deal with *data*; they just define the transport, access, and presentation of data. There is a difference, and it’s on the same level as the difference that separates cloud services from application services.

Any application deployed in a cloud and accessed by users IS NOT A CLOUD SERVICE. I repeat, it’s NOT A CLOUD SERVICE. It’s an APPLICATION.

AN IMPORTANT AND NOT ALL THAT SUBTLE DISTINCTION

The “cloud service” is Microsoft’s platform. A “cloud service” is Google’s platform, or Salesforce.com’s platform, or [BlueLock](#) or [GoGrid](#) or [Amazon](#). A “cloud service” is used by IT, by developers, by the technical community at large. What consumers access is an *application*, and nothing more. They aren’t the user of the cloud service, they are the consumer of an application deployed in a cloud environment. Google Docs is an application. Gmail is an application. [Twitter](#) is an application. None of these are “cloud” services, even when using APIs designed to integrate them with *other* applications; they are still, always and forever, *applications*.

Interoperability and portability, whichever Lauren (and others) is calling for, is not going to solve this problem. This problem can only be solved by the application provider, which in this case is T-Mobile. It is the responsibility of T-Mobile to provide a means by which the data stored in *its application* can be transferred to another application – cloud-based or otherwise – and its data is properly backed up (which is yet another piece of this supposed cloudtastrophe puzzle that isn't being addressed enough). And that's it. It is the responsibility of other application providers to offer a means by which *that* data can be imported and transferred to *their* application, thus providing the portability that is apparently demanded by consumers if we are to listen to the myriad cries erupting at every little hiccup in the cloudosphere.

Portability of *application services*, like Sidekick, is not a cloud problem. It is an ancient problem that goes back to the first attempt at sharing data across two applications that continues to plague enterprises and developers like some kind of immortal, invulnerable locust. An entire software industry focuses on making this process as simple as possible; you may have heard of it, it's called enterprise *application* integration (EAI). This industry exists because no two applications store their data in exactly the same way or in the same format or in the same database. Thus there are standards and tools provided to allow two applications to share, extract, transform, and otherwise access that data.

Lauren mentions this later in her post:

There are positive ways to proceed. Google, for example, a leader in cloud computing, has recently launched a specific project -- [The Data Liberation Front](#) -- explicitly including as a key facet the goal of making sure that users can quickly and easily export data from Google products. This ambitious and extremely important effort should be a model for the rest of the cloud computing industry.

Note the operative "from Google products." Google isn't going to provide interoperability or portability of its *cloud services*, it's providing access to *application* data that effectively promotes integration of its *application services* with third-party application services – wherever they may be hosted. This is no different than taking advantage of [Salesforce.com's](#) Web Services API to enable integration between its *application services* and some enterprise-hosted application.

But the cloud services, which are the applications, will be no more or less portable than they are today when at last you can get at the data. Because that data, surely in some kind of raw format (XML, JSON, etc...) will not be usable by consumers anyway. It will need to be interpreted by *yet another application*. Another application that is, likely, running as a cloud-based application.

Once people start recognizing the distinction between the two then perhaps we'll actually be able to make some headway toward resolving the *application integration* challenges in the cloud. Until then, we're just spewing so much pollution into the clouds with these calls for "cloud" portability and interoperability that no one can see what's really necessary.



- [Lauren Weinstein's Blog: Microsoft's "Cloudburst" – Spectacular Data Loss Drowns Sidekick Users](#)
- [The Sidekick Failure and Cloud Culpability](#)
- [Cloud: The Other White Meat...](#)
- [Infrastructure Integration: Metadata versus API](#)
- [If You Focus on Products You'll Miss the Cloud](#)
- [Putting the Cloud Before the Horse](#)
- [The Infrastructure 2.0 Trifecta](#)
- [Do you control your application network stack? You should.](#)
- [Dynamic Infrastructure: The Cloud within the Cloud](#)

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