

The Secret to Amazon's Success--Internal APIs

12 Jan 2012

Last year there was an [accidental post from a Google employee](#) about Google+. The internal rant was accidentally shared publicly and provides some insight into how Google approached APIs for their new Google + platform, as well as insight how Amazon adopted an internal service oriented architecture (SOA).



The insight about how Google approached the API for Google+ is interesting, but what is far more interesting is the insight the Google engineer who posted the rant, Steve Yegge, provides about his time working at Amazon, before he was an engineer with Google.

During 6 years at Amazon he witnessed the transformation of the company from a bookseller to the almost \$1B, Infrastructure as a Service (IaaS) API, cloud computing leader. As Yegge's recalls that one day Jeff Bezos issued a mandate, sometime back around 2002 (give or take a year):

- All teams will henceforth expose their data and functionality through service interfaces.
- Teams must communicate with each other through these interfaces.
- There will be no other form of inter-process communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
- It doesn't matter what technology they use.
- All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.

The mandate closed with:

Anyone who doesn't do this will be fired. Thank you; have a nice day!

Everyone got to work and over the next couple of years, Amazon transformed itself, internally into a service-oriented architecture (SOA), learning a tremendous amount along the way.

Think about what Bezos was asking! Every team within Amazon had to interact using web services. If you were human resources and you needed some numbers from marketing, you had to get them using an API. He was asking every team to decouple, define what resources they had, and make them available through an API. Every team within your company essential becomes a partner of the other.



Some of the lessons Amazon learned along the way:

- **Support** - Support for your teams interface becomes critical
- **Security** - Every teams becomes a potential DOS attacker requiring service levels, quotas and throttling
- **Monitoring / QA** - Monitoring and QA are interconnected, you will need smart tools for not just telling if something is up and running, but actually delivering the expected results
- **Discovery** - Service discovery becomes important. You will need to know what APIs there are, if they are available and where to find them.
- **Testing** - Sandbox and debugging is essential for all APIs

Of course this is a very small sampling from a rant of a former employee. There are dozens, maybe hundreds of individual lessons like these that Amazon had to discover organically. Yegge's points out that, "Organizing into services taught teams not to trust each other in most of the same ways they're not supposed to trust external developers."

This makes deploying internal APIs a great exercise for preparing your company for the coming API economy where, you will have to have expose self-service, partner and public APIs to stay competitive in your industry.

When Amazon started, it was difficult to see how the bookseller would become the e-commerce powerhouse it is today, let alone to see that Amazon would transform culturally into a company that thinks and operates as a service oriented architecture, delivering Amazon Web Services, that not only changed how the company operates, but created an entire platform that would change how the Internet works.