

## Google: A MIRV from the Cloud

Do you remember the MIRV? It's a weapons system with a "gotcha". One rocket delivers a war head that deploys multiple payloads. MIRVs can be tough to intercept with conventional anti-warhead techniques. If you don't have a super, whiz-band missile counter measure, you're toast.

Google Apps engine is a digital MIRV. Most of the Google watchers point to this new service as a direct threat to Microsoft's enterprise initiatives. Others assert that Google wants to respond to Amazon's cloud-initiatives: EC2 (elastic compute cloud) SimpleDB, S3 (Simple Storage Solution), and other bits of wizardry you can read about here:

<http://www.amazon.com/gp/browse.html?node=3435361>. If you can't locate the service via Amazon's search engine A9.com, navigate to Google.com and enter the phrase "Amazon developer program". A handful of Web 2.0 wizards see the Apps engine as a response to Facebook.com's platform pretensions.

Everyone is right? Google Apps is a digital MIRV. One delivery vehicle—Google Apps. Whoosh! An attack on Redmond and Seattle. An attack on Facebook.com. The collateral damage from one hit would be significant. If all three warheads find their target, the Google has reshaped the landscape. In the rebuilding, Google has opportunities. The targets of the strike have their hands full determining what's been damage, calculating costs, and figuring out how to respond. Salesforce.com has been quick to announce its support for Google Apps. Google has been cheerleading Salesforce.com because Google likes "those guys". Salesforce.com is a petri dish in which Google can study without significant costs the cloud-computing phenomena.

The Google Apps engine is a small-scale test launch. Googlers have been quick to point out that any one lucky enough to be one of the 10,000 allowed to fiddle with the new service get access to Google's infrastructure. Poking through the sparse documentation, it's obvious that there are significant constraints on this test vehicle. There's a Google-imposed limit of 500 megabytes of storage. Any programming is permitted as long as it is Python. The service is free at this time (April 2008), but like Amazon's "free" wireless for Kindle owners, the policy can change with the flip of a bit. Google's data management system—BigTable—is part of the package, but Google has kept mum about its proprietary Sawzall language developed specifically for BigTable. GQL or Google Query Language, like perl, haven't been bolted on this first MIRV.

What can you do with Google Apps engine? If you navigate to Google's information page (<http://code.google.com/appengine/>), you learn "no assembly required". App Engine is "easy to scale". And—like most of Google's betas—"it's free to get started". The demos are modest applications; for example, a utility to display multiple To Do lists side by side (ToDo), a lightweight collaboration tool (MobWrite), and a basic online chat tool (YouTalk). You can see these and other mini-applications built using App Engine here: <http://appgallery.appspot.com/>.

Below is an example of the standard Python code used to build apps for the engine. This snippet receives text from a Web form, formats it appropriately, and creates a web server response to serve it to

a web browser. (Why? It's a quick-and-dirty way for programmers to exchange code without having to worry that sending it via Instant Messenger or IRC would mangle the whitespace and other formatting – a particular concern of Python coders.)

```
def post(self):
    user = users.get_current_user()

    #XXX: needs validation!

    paste = Paste()

    paste.uid = "p" + str(uuid.uuid4())[:8]
    if user:
        paste.userid = user.nickname()
    else:
        paste.userid = "anonymous"

    paste.title = self.request.get("title")
    paste.comment = self.request.get("comment")
    paste.code = self.request.get("code")
    paste.put()

    self.response.headers['Content-Type'] = 'text/html'
    if user:
        self.response.out.write(Site(self, user.nick()).getContent(uid=paste.uid))
    else:
        self.response.out.write(Site(self, 'anonymous').getContent(uid=paste.uid))
```

How does Google's approach differ from the more mature Amazon cloud services? On the surface, the Google App Engine looks quite a bit like an Amazon clone. Upon closer inspection, Google packs all of the separate Amazon functions into one service. At Amazon, you can choose to use storage, the database, or other discrete functions. Not with the App Engine. Google becomes your computer. Amazon provides at a modest cost access to separate functions. The Google App Engine is free, which may be significant for cheapskates like venture capitalists who want their investment dollars to generate revenue, not huge capital costs. Entrepreneurs may give App Engine a whirl. It's not just free (although price is a factor), the App Engine is Google's, and Messrs. Brin's and Page's “cool factor” is a tad higher than investment banker / bookseller / code wizard guru, Jeff Bezos' rating to math-loving 20-somethings.

The downside of the App Engine could consume several thousand words. Let me boil down its weaknesses to a few of the larger drawbacks. First, your code, your idea, and your data are in the Google's claws. Use App Engine and you cede control to the Google. I'm not worried, but some engineers are more paranoid than I. So, fear, security, control—these are deal breakers for some.

The App Engine is a work in progress. The demo applications come and go. Features and functions and policies can be just as fluid. Developers hands are on controls like those boardwalk games with little wheels and levers you turn to grab a prize. You can push and jerk the wheels, but most of the time, the game does what it does best: consumes your time and your money without yielding a prize. Without a track record, the App Engine could be just as tough to steer to your end. I feel confident in my ability to surf the Google-induced waves, but you might not have my degree of hubris.

STEPHEN E ARNOLD APRIL 10, 2008

Third, Google may not make BigTable, Sawzall, GQL, and other Googley services available to App Engine users. The result is that the service is a way for Google to get great ideas, more user data, and more information about how to “hook” developers. Microsoft's developer program is a money-making exercise. Google may be using its App Engine to find out how to convert developer interest into a money-making business. Like Salesforce.com, Google envisions an application exchange. A clever developer for Salesforce.com can make some money on his code. Google's reach and business goals are somewhat more expansive than Salesforce.com's. Keep in mind that in March 2006, Google's management offered a revenue target of \$100 billion. (No, that's not a typo; it's b as in billions of dollars. To give you some perspective, IBM and Microsoft are in the \$60 billion revenue zone. Google wants to best this healthy figure by \$40 billion. One analyst suggested the time line was in the 2011 or 2012 range.

One other downside is Google's ability to lose focus or drift away from certain initiatives. For example, when was the last time you heard about Google's programmable search engine, data spaces, or containers. Maybe Google is playing coy with these engineering marvels, but Google seems to invest in serious engineering, roll out some betas or test shots, and then move on to other, more interesting technologies. The fear is that Google could lose interest. Why waste cycles learning something that could go away in a three to six months. I don't think Google's cloud computing will go away, but it's important to be aware of historical Google starts, stops, and wanderings. As you read this (May 23, 2008) Google will unveil more demonstrations of services using the App Engine. Given Google's penchant for slip streaming enhancements into its betas, App Engine, like Topsy, will grow and grow fast. Google engineers will be stuffing digital Feltner Whatta-Burgers (Russellville, Arkansas, in case you're wondering) down the App Engine's gullet.

Google App Engine is a significant step for Google. Most of its competitors are unaware of technologies that Google can “snap into” its cloud platform. Let me conclude with one example. Google has a clever invention that makes it possible for Google to virtualize small applications wherever a connection to Google exists. You can see this in action if you navigate to [www.google.com/ig](http://www.google.com/ig) and customize a Web page and tabs. These functions are running in containers. Now, if developers create mini-applications that run in the ig or individualized Google session, we don't have cloud computing. We have a new way of interacting within, across, and through a Google space. Google Apps is an important step toward the company's “final frontier”. Beam me up, Sergey.

Stephen Arnold, April 10, 2008

Image: MIRV.tif

**Caption:** This MIRV image is from George Washington University in its National Security Agency archive here: [www.gwu.edu/~nsarchiv/nsa/NC/mirv/mirv.gif](http://www.gwu.edu/~nsarchiv/nsa/NC/mirv/mirv.gif)