

September 09, 2008

## What Language/Framework to use on a brand-new project?

In the next 24 hours I will be starting a new project - a "next generation LMS system" for the UM Medical School - this is a small prototype but I hope it shows the way for how LMS systems will work in the future.

My angst today is what to write it in - and frankly I am still not 100% sure - if you have any suggestions - hurry up - I need to know soon.

Should I use Java?

Java scales and when you have a lot of code - strict interfaces are great. But Java simply means that when you try to build an open source community - you will be ridiculed by hobby programmers for choosing Java because it is too \*hard\*.

PHP + MySql / LAMP?

Man this is tempting - it works so well - ISP's make this work for \$3 per month trivially. It is really really direct to slap stuff together. And it is nice and relational. It feels so hacky though. Maybe PHP5 makes it feel a little better.

Ruby on Rails?

I love Ruby on Rails - I have never felt so productive in an environment - it is the best way to write elegant code I know of. But it needs to find a hold in the market place. I do not like the feeling of living on the edge and having to become a zealot and sell folks on the framework before I sell them on my product. Last year this time - I put together a small LMS in Ruby - and it was great fun - <http://www.rubytoozday.org/>. I could even start with that code for my NG project.

Python and App Engine?

Google is behind this - I like that a \*lot\*. I like Python. I like App Engine's free accounts. I worry that I cannot scale my service in App Engine yet - there is no pay account and I cannot run App Engine on my own production infrastructure. Oh yeah - and that missing relational database might be a bit of a learning curve. At the same time - in the real world of big-time scalability - sharding is your friend - so why not bite the bullet and force myself to understand sharding.

So on the drive into work this morning - it seemed that even in 2008 - there was no good solution to this problem. DANG!

But as the miles rolled by - I realized that my new application was really about Javascript, JQuery, JSON, REST, and some back end. And I was fussing about what the back-end would be. I already had made much of the decisions of the project without realizing it.

So this makes it less painful to make a choice. Because it is only the headless back end that I am talking about. Even so the back end is an important bit.

So unless I change my mind in the next 24 hours - it will be Python with the following caveats: (a) the real good stuff is isolated into code that does not depend on the App Engine at all, (b) any APIs are defined in REST/JSON and not in Python itself.

The thing that scares me the most is learning sharding under pressure. This is the App Engine risk. I will need to hide my storage behind an abstraction. If I feel too much pain too quickly, I may fall back to PHP or Rails - that will be a harder decision - because Rails is so productive but I don't have it memorized yet - PHP is a hack but I do have more of PHP and MySql memorized.

Oh the pain of making this decision!

If someone wants to help and/or get involved in this - let me know. I don't have enough time to do this and I am pretty much a beginner in the technologies I am intending to use (REST, JSON, Javascript, JQuery, and Python App Engine). I guess that is why they call it "life long learning".

Seriously, if you want to help, I would much appreciate it - I need folks who can really \*help\* - I do need folks smarter than me who have more time than me. I may be even able to use Med School funds to pay the right person to work on this with me.

Posted by csev at September 9, 2008 11:07 AM