

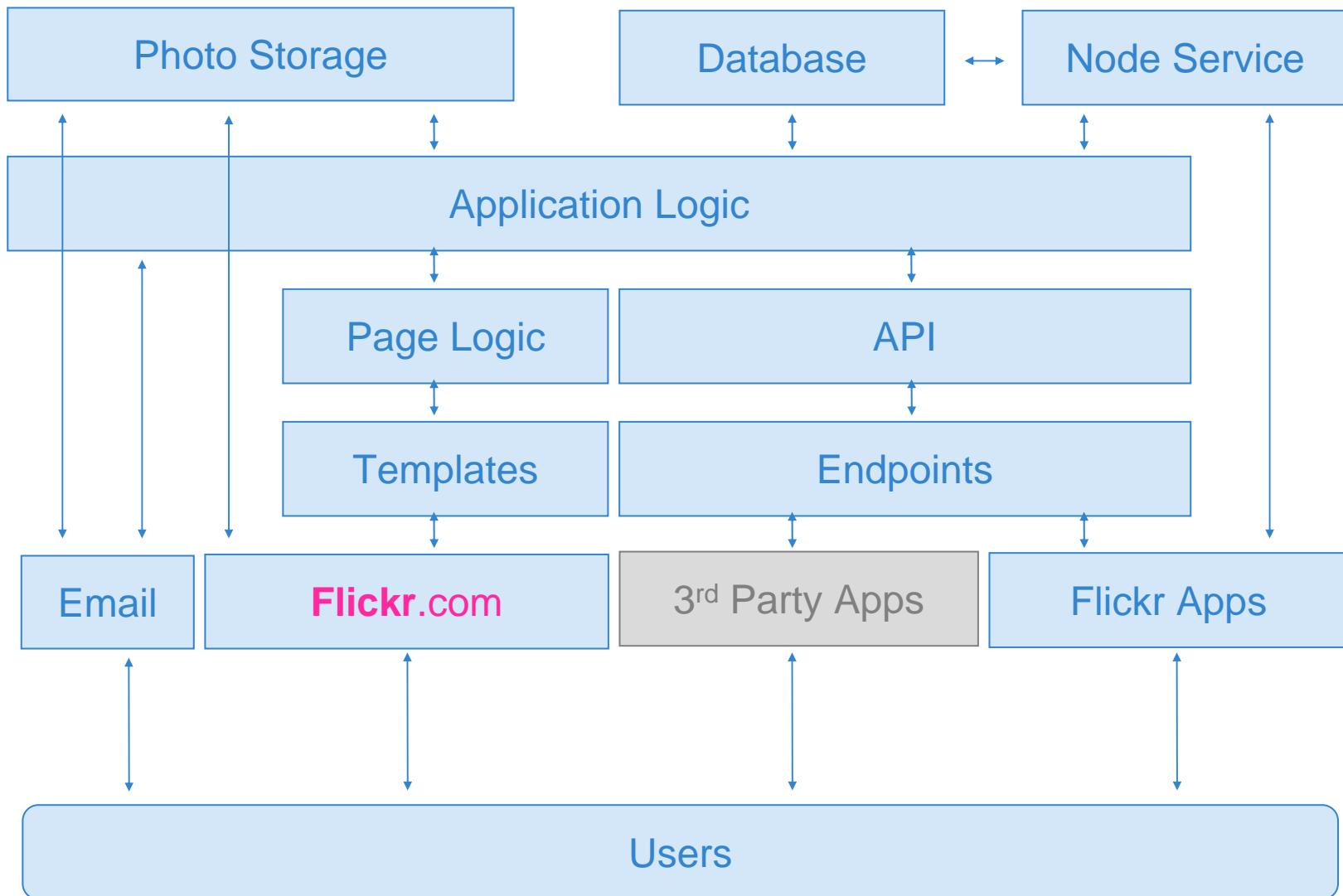
# Flickr and PHP

Cal Henderson

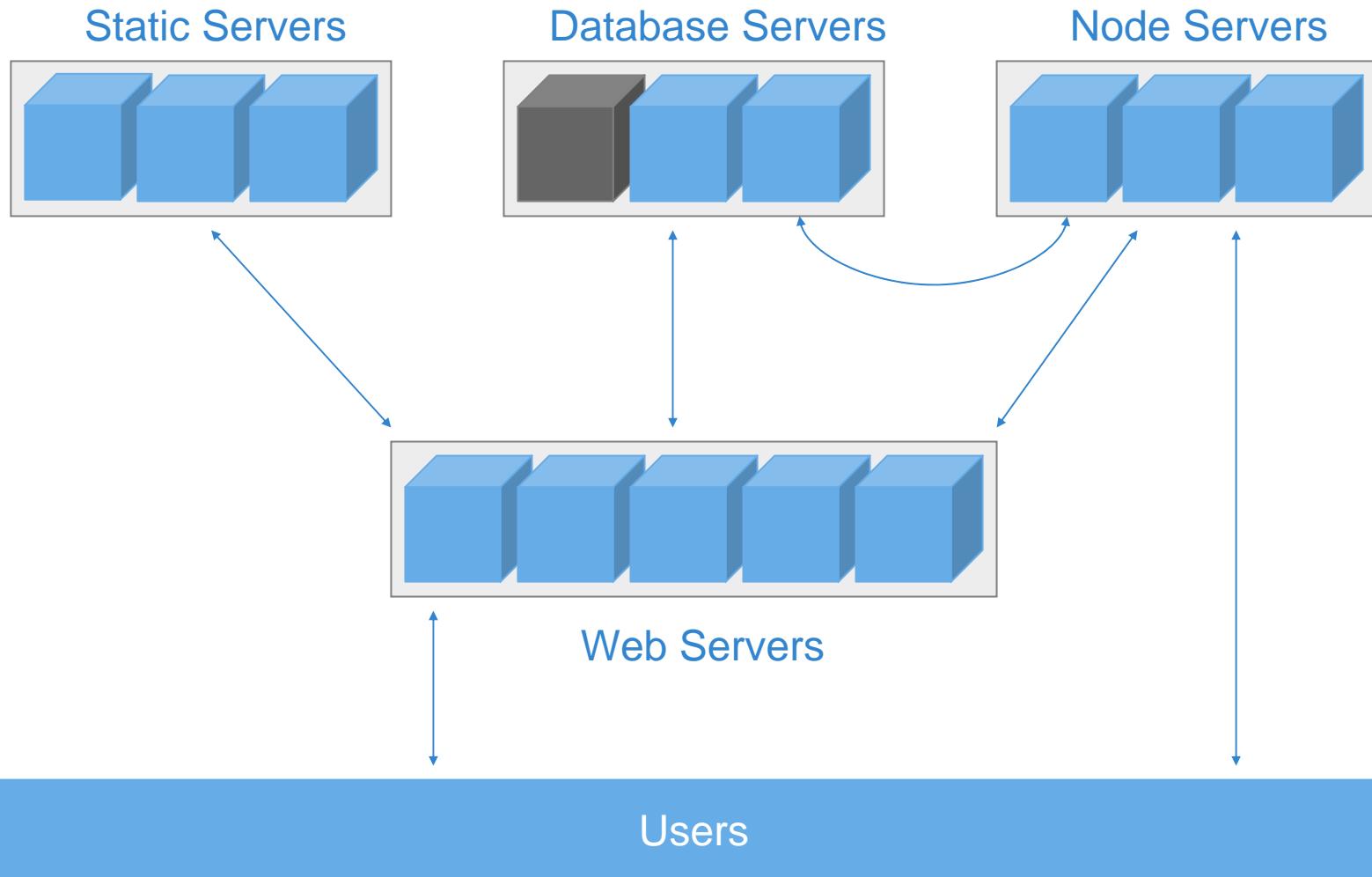
# What's Flickr

- Photo sharing
- Open APIs

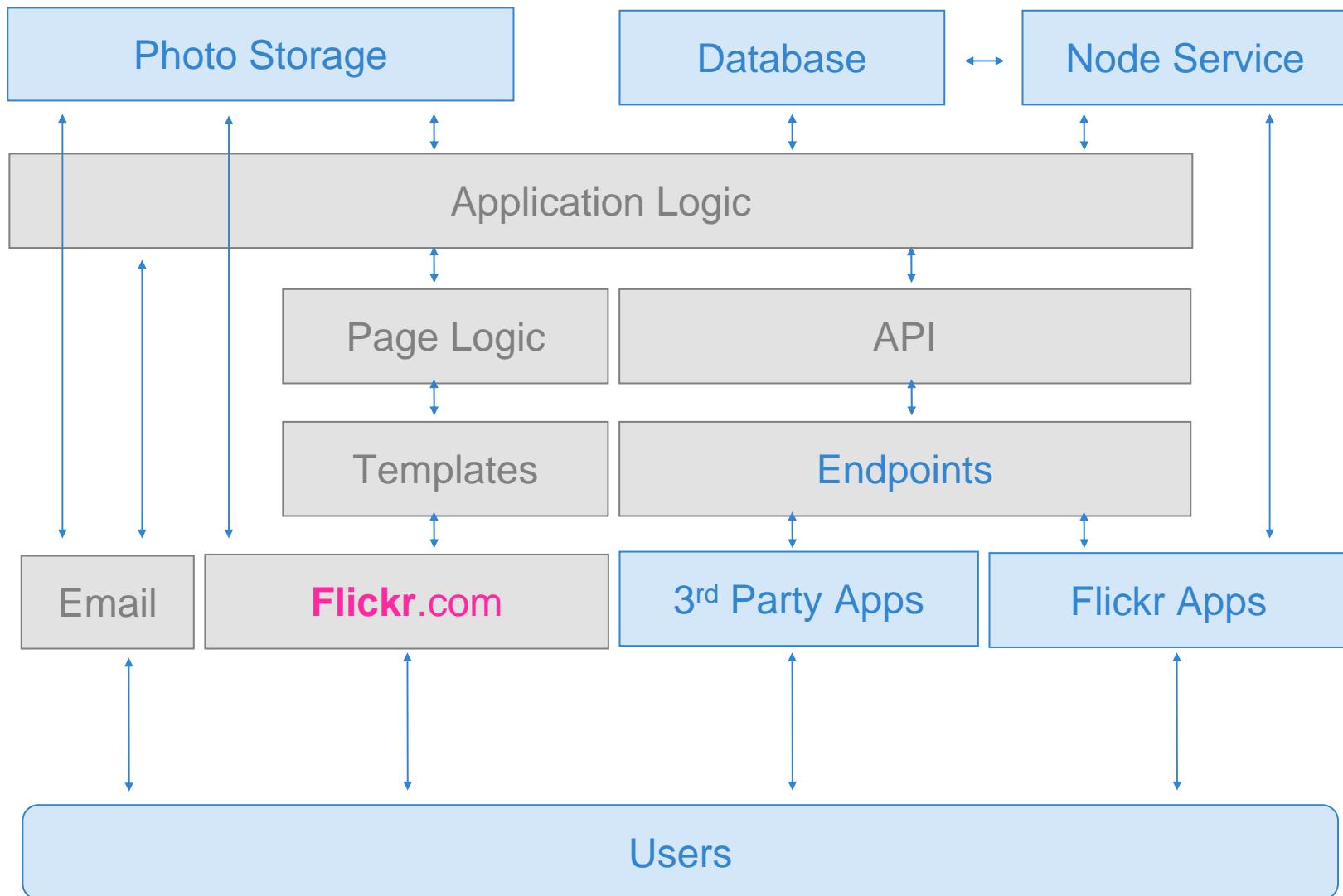
# Logical Architecture



# Physical Architecture



# Where is PHP?



# Other than PHP?

- Smarty for templating
- PEAR for XML and Email parsing
- Perl for controlling...
- ImageMagick, for image processing
- MySQL (4.0 / InnoDB)
- Java, for the node service
- Apache 2, Redhat, etc. etc.

# Big Application?

- One programmer, one designer, etc.
- ~60,000 lines of PHP code
- ~60,000 lines of templates
- ~70 custom smarty functions/modifiers
- ~25,000 DB transactions/second at peak
- ~1000 pages per second at peak

# Thinking outside the web app

- Services
  - Atom/RSS/RDF Feeds
  - APIs
    - SOAP
    - XML-RPC
    - REST
    - PEAR::XML::Tree

# More cool stuff

- Email interface
  - Postfix
  - PHP
  - PEAR::Mail::mimeDecode
- FTP
- Uploading API
- Authentication API
- Unicode

# Even more stuff

- Real time application
- Cool flash apps
- Blogging
  - Blogger API (1 & 2)
  - Metaweblog API
  - Atom
  - LiveJournal

# APIs are simple!

- Modeled on XML-RPC (Sort of)
- Method calls with XML responses
- SOAP, XML-RPC and REST are just transports
- PHP endpoints mean we can use the same application logic as the website

# XML isn't simple :(

- PHP 4 doesn't have good a XML parser
- Expat is cool though (PEAR::XML::Parser)
- Why doesn't PEAR have XPath?
  - Because PEAR is stupid!
  - PHP 4 sucks!

# I love XPath

```
if ($tree->root->name == 'methodResponse'){
    if (($tree->root->children[0]->name == 'params')
        && ($tree->root->children[0]->children[0]->name == 'param')
        && ($tree->root->children[0]->children[0]->children[0]->name == 'value')
        && ($tree->root->children[0]->children[0]->children[0]->children[0]->name == 'array')
        && ($tree->root->children[0]->children[0]->children[0]->children[0]->children[0]->name == 'data')){
            $rsp = $tree->root->children[0]->children[0]->children[0]->children[0];
        }
    if ($tree->root->children[0]->name == 'fault'){
        $fault = $tree->root->children[0];
        return $fault;
    }
}
```

```
$nodes = $tree->select_nodes('/methodResponse/params/param[1]/value[1]/array[1]/data[1]/text()');

if (count($nodes)){
    $rsp = array_pop($nodes);
}else{
    list($fault) = $tree->select_nodes('/methodResponse/fault');
    return $fault;
}
```

# Creating API methods

- Stateless method-call APIs are easy to extend
- Adding a method requires no knowledge of the transport
- Adding a method once makes it available to all the interfaces
- Self documenting

# Red Hot Unicode Action

- UTF-8 pages
- CJKV support
- It's *really* cool

# どんなカメラをお使いでしよう？

[日本語flick'r](#) / [Discuss](#)

フォスター・ペアレン  
ト  
交流しながら途上国の  
子どもを支援。月々  
3,000円から。詳細と資  
料請求

公共サービス広告 by  
Google

Current Discussion

[group photo pool](#)

31 Aug '04, 11.45pm PDT

[contactsってどうやって](#)



[kazenotayori says:](#)

現行機種:  
撮影後はどうしていますか:  
次に買うのは何?:

みたいな感じで…

Posted at 11:35pm, 18 August 2004 PDT ([Permalink](#))



[kazenotayori says:](#)

現行機種:  
[CASIO QV-2000UX 2.1Mega Pix](#)  
撮影後はどうしていますか:  
[Photoshop](#)してから、リサイズ  
次に買うのは何?:  
やっぱり、[D70](#)でしょうか。

Posted 2 weeks ago. ([Permalink](#))

# Unicode for all

- It's *really* easy
  - Don't need PHP support
  - Don't need MySQL support
  - Just need the right headers
  - UTF-8 is 7-bit transparent
  - (Just don't mess with high characters)
    - Don't use HtmlEntities()!
- But bear in mind...
  - JavaScript has patchy Unicode support
  - People using your APIs might be stupid

# Scaling the beast

- Why PHP is great
- MySQL scaling
- Search scaling
- Horizontal scaling

# Why PHP is great

- **Stateless**
  - We can bounce people around servers
  - Everything is stored in the database
  - Even the smarty cache
  - “Shared nothing”
  - (so long as we avoid PHP sessions)

# MySQL Scaling

- Our database server started to slow
- Load of 200
- Replication!

# MySQL Replication

- But it only gives you more SELECT's
- Else you need to partition vertically
- Re-architecting sucks :(

# Looking at usage

- Snapshot of db1.flickr.com
  - SELECT's 44,220,588
  - INSERT's 1,349,234
  - UPDATE's 1,755,503
  - DELETE's 318,439
  - 13 SELECT's per I/U/D

# Replication is really cool

- A bunch of slave servers handle all the SELECT's
- A single master just handles I/U/D's
- It can scale horizontally, at least for a while.

# Searching

- A simple text search
- We were using RLIKE
- Then switched to LIKE
- Then disabled it all together

# FULLTEXT Indexes

- MySQL saves the day!
- But they're only supported by MyISAM tables
- We use InnoDB, because it's a lot faster
- We're doomed

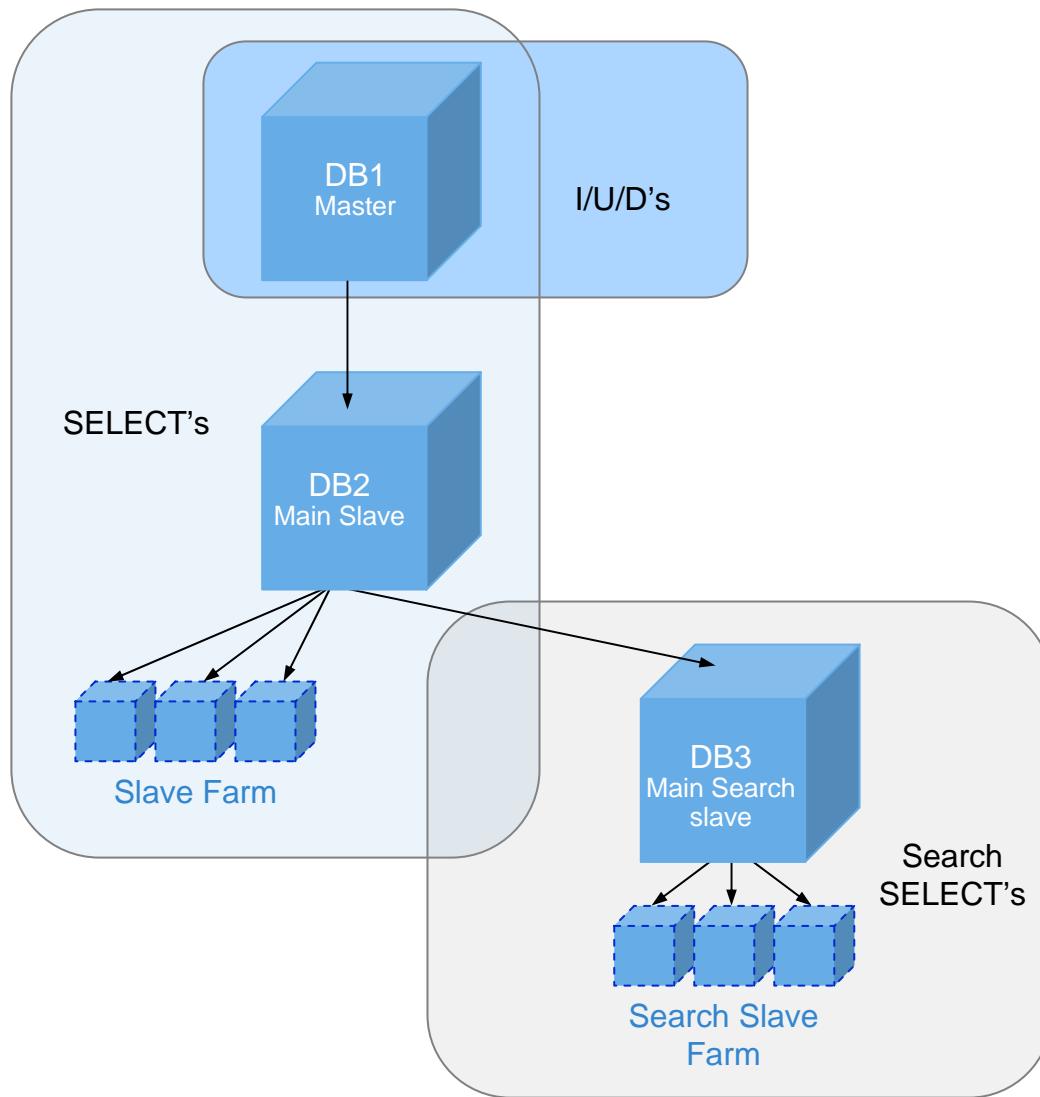
## But wait!

- Partial replication saves the day
- Replicate the portion of the database we want to search.
- But change the table types on the slave to MyISAM
- It can keep up because it's only handling I/U/D's on a couple of tables
- And we can reduce the I/U/D's with a little bit of vertical partitioning

# JOIN's are slow

- Normalised data is for sissies
- Keep multiple copies of data around
- Makes searching faster
- Have to ensure consistency in the application logic

# Our current setup



# Horizontal scaling

- At the core of our design
- Just add hardware!
- Inexpensive
- Not exponential
- Avoid redesigns

# Talking to the Node Service

- Everyone speaks XML (badly)
- Just TCP/IP - `fsockopen()`
- We're issuing commands, not requesting data, so we don't bother to parse the response
  - Just substring search for `state="ok"`
- Don't rely on it!

# RSS / Atom / RDF

- Different formats
- All quite bad
- We're generating a lot of different feeds
- Abstract the difference away using templates
- No good way to do private feeds. Why is nobody working on this? (WSSE maybe?)

# Receiving email

- Want users to be able to email photos to Flickr
- Just get postfix to pipe each mail to a PHP script
- Parse the mail and find any photos
- Cellular phone companies hate you
- Lots of mailers are retarded
  - Photos as text/plain attachments :/

# Upload via FTP

- PHP isn't so great at being a daemon
- Leaks memory like a sieve
- No threads
- Java to the rescue
- Java just acts as an FTPd and passes all uploaded files to PHP for processing
- (This isn't actually public)
- Bricolage does this I think. Maybe Zope?

# Blogs

- Why does everyone loves blogs so much?
- Only a few APIs really
  - Blogger
  - Metaweblog
  - Blogger2
  - Movable Type
  - Atom
  - Live Journal

# It's all broken

- Lots of blog software has broken interfaces
- It's a support nightmare
- Manila is tricky
- But it all works, more or less
- Abstracted in the application logic
- We just call `blogs_post_message()` ;

# Back to those APIs

- We opened up the Flickr APIs a few weeks ago
- Programmers mainly build tools for other programmers
- We have Perl, python, PHP, ActionScript, XMLHTTP and .NET interface libraries
- But also a few actual applications

# Flickr Rainbow



# Tag Wallpaper



# So what next?

- Much more scaling
- PHP 5?
- MySQL 5?
- Taking over the world

# Flickr and PHP

Cal Henderson

**Any Questions?**