**facebook**

# Facebook Technologies

11/18/10
Wayne Chang (wayne@fb.com)
Software Engineer

facebook 　 　 　 | Search 🔍

**Wayne Chang**

**Wall** | Info | Photos | Questions | +

Share: 📝 **Status**　🖥 **Question**　🖼 **Photo**　🔗 **Link**　🎥 **Video**

Wayne + Friends　**Wayne**　Just Friends　　　　　　　⚙ Settings

RECENT ACTIVITY

💬 Wayne commented on Sonia Nagala's photo.
💬 Wayne commented on Sonia Nagala's photo.
👥 Wayne is now friends with Priyanka Goel and 2 other people.

View Photos of Me (3,996)
View Videos of Me (48)
Edit My Profile

can i have this dance?　　✏

**Information**　✏

Networks:
Facebook
Stanford Alum '05
Google

Relationship Status:
**Married to**
Sonia Nagala

Birthday:
**June 22, 1983**

Current City:
**Seattle, WA**

Hometown:
**Sunnyvale, CA**

Political Views:
**Save the planet**

Website:
http://www.facebook.com/wayne

**Friends**　✏

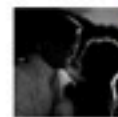1,045 friends　　　See All

**Wayne Chang** was tagged in Chris Pan's album.

**Mobile Team Wine Tasting – Nov 2010**
9 new photos
🖼 November 12 at 4:30pm

RECENT ACTIVITY

👥 Wayne is now friends with Joshua Harder and Anu Codaty.
💬 Wayne commented on Julie Zhuo's link.
👥 Wayne and Herbert Duarte are now friends. · Like · Comment

**Wayne Chang** was tagged in his own album.

# Agenda

- Facebook's scale
- Open Source Technologies
  - Memcached
  - Thrift
  - XHP
- Facebook Ads
- Questions?
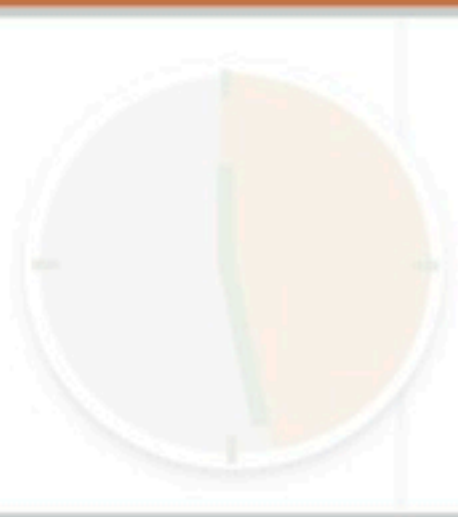
# 500 million unique experiences

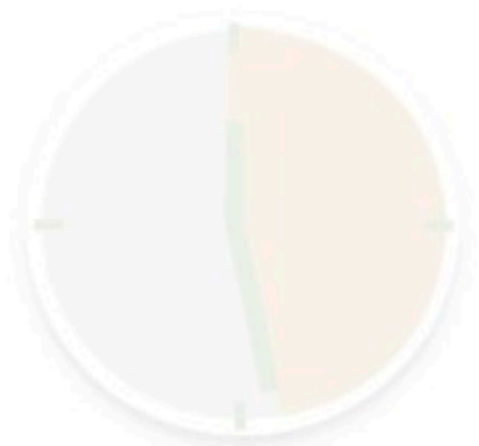Over **20 billion** minutes spent every day

Jane Richie wrote on Jim Morgan's wall.
Did you see the game tonight!?

Johnny Chen Two thumbs up for Zombieland.
9 hours ago · Comment · Like

Mark Zuckerber commented on Dave Morin's Photo.
...rce be with you.

Kate Losse is pleased by the intermittent singing of T-Pain lyrics by Eric Zamore in the office today. Might start intermittently singing Akon songs
Yesterday at 3:20pm · Comment · Like

Brandon McCormick I hate anthropomorphic dogs
Yesterday at 3:18pm · Comment · Like

Mike Ferrier fly high Facebook Falcons. Game tonight and Nuthouse after.
Yesterday at 7:36pm · Comment · Like

1 your friends are attending Family Picnic.
It's hosted by Jill Sparks. So far 18 people have been invited.
RSVP to this event

Lee Byron Snack of the day: Roast pumpkin seeds in reserved bacon fat; sprinkle with cayenne, sea salt. David Byron would be proud.
about an hour ago · Comment · Like

Jim Morr commented on Jack Dean's Photo.
Wow what a view. The Grand Canyon is just so amazing!

Viola Alvares 4-1 well done!
Yesterday at 8:52pm via Mobile Web · Comment · Like

Garry Kindle commented on Sara Lima's Photo.
Oooh Yum. Can I have some of that goodness? It's making my mouth water.

Dan Jones wrote on Katie Lee's wall.
Great to see you last night. Catch you late

4 of your friends are attending after work drinks.
It's hosted by Jim Leman. So far 8 people have been invited.
RSVP to this event

Jill Sparks commented on Frank Sparks' Video.
Grandma's Rose Garden.
0:45 Recorded 2 days ago

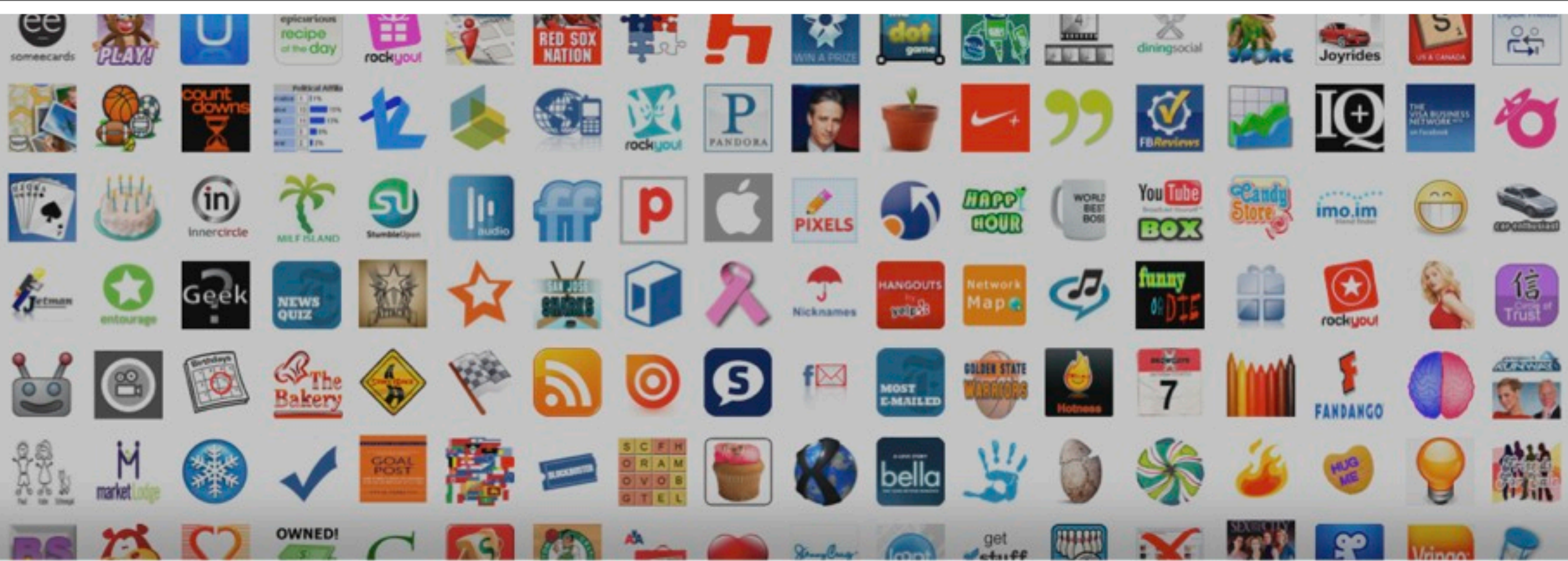# Over 7 billion pieces of content shared every week

31
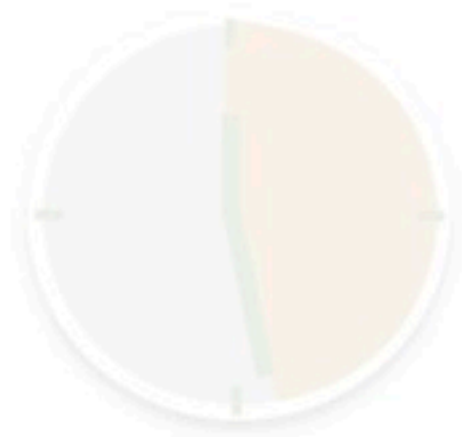
Over **3 billion** photos uploaded each month

500,000 Facebook Connect implementations
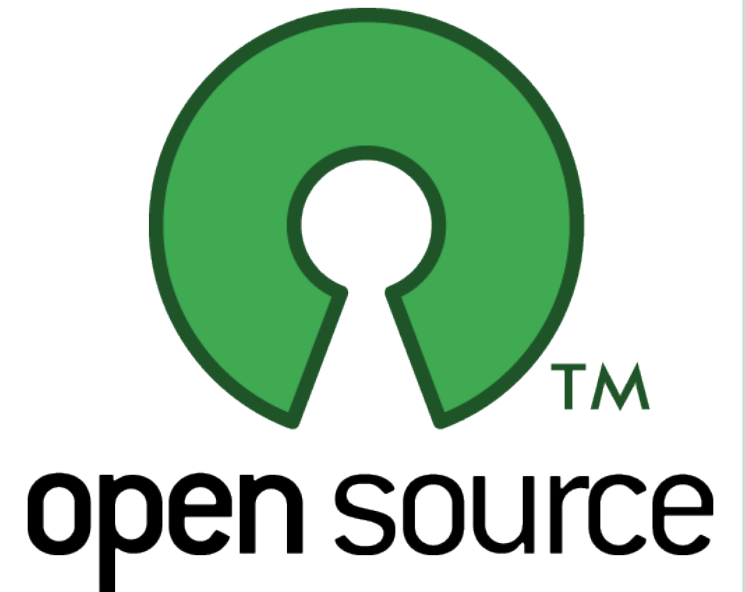
# Open Source

# Open Source

http://developers.facebook.com/opensource/

http://github/facebook
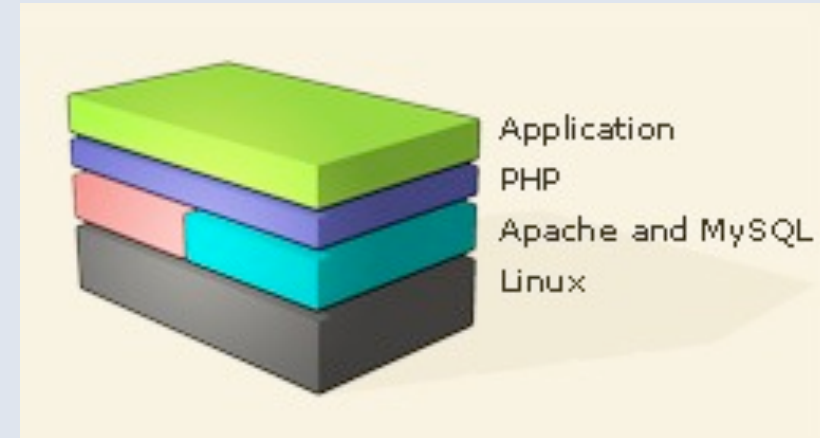
# Memcached

# Memcached
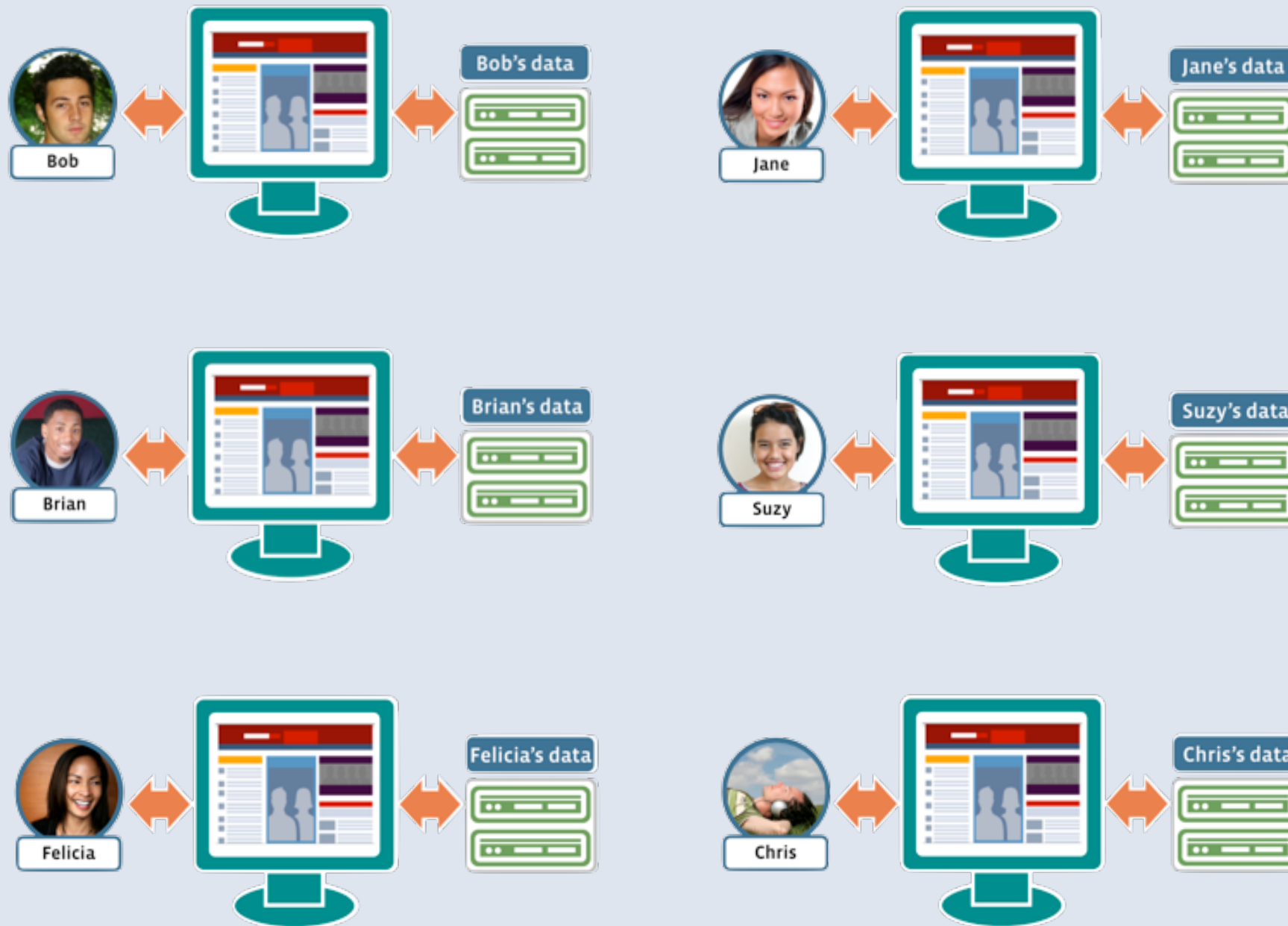## Scaling Traditional Websites

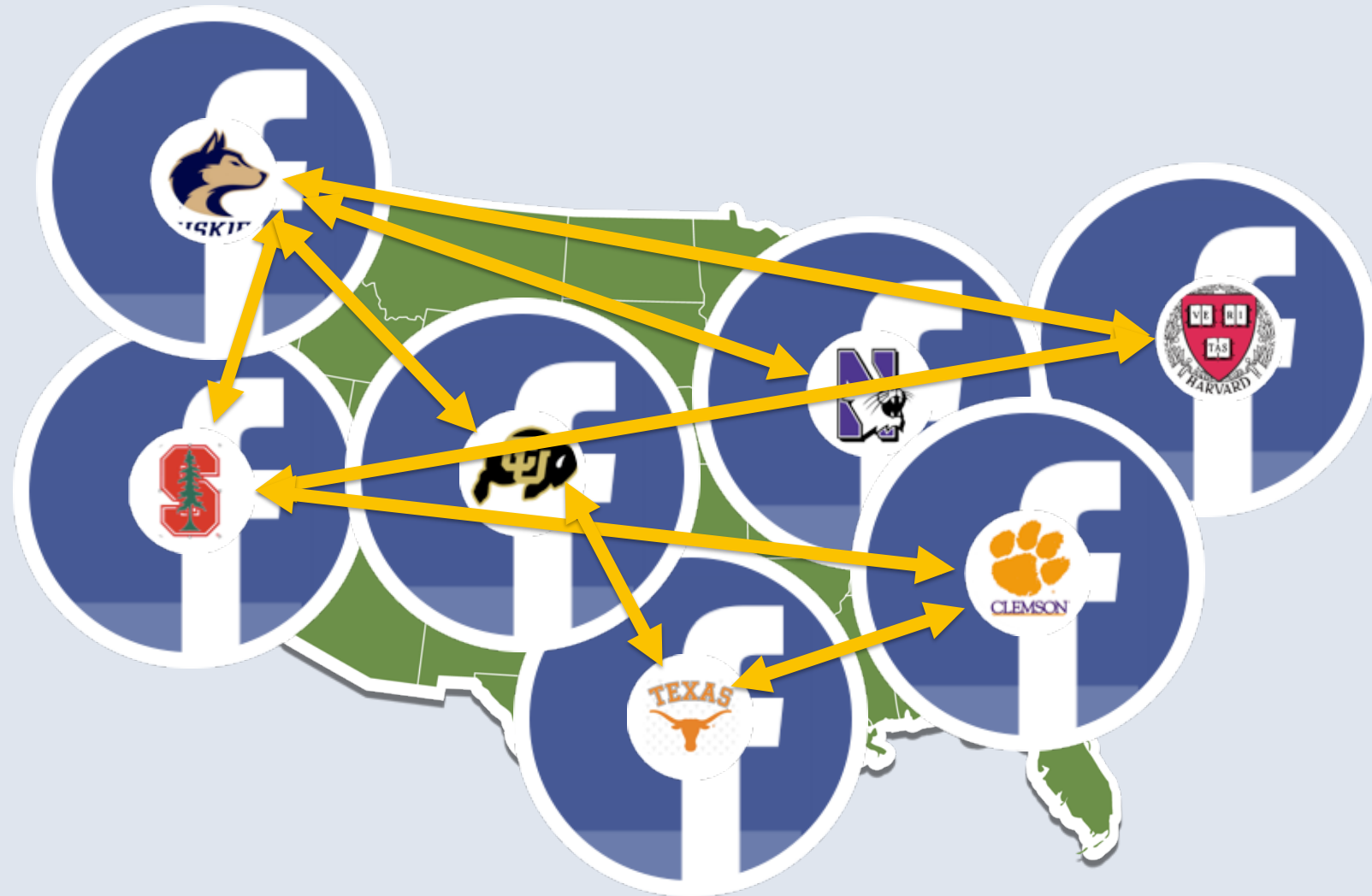# Memcached

## Scaling Traditional Websites
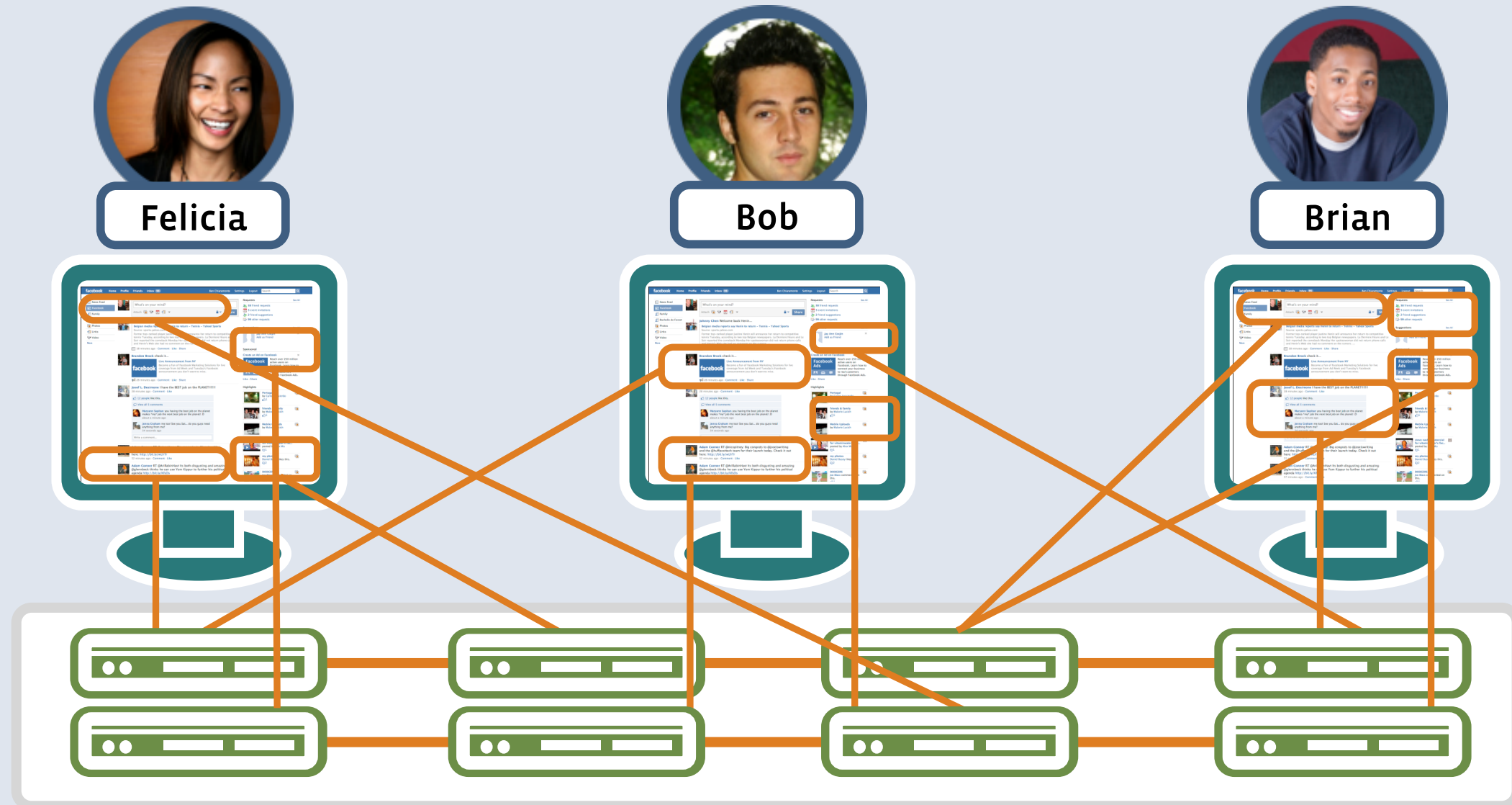
# Memcached

## Scaling Traditional Websites

# Memcached

## The problem with social applications

# Memcached

## Scaling Facebook: Interconnected Data

**500 million** unique home pages

Memcache

Memcache

**100 million** operations per second

# Our Answer: Memcached

- Developed @ Livejournal 2003

- First used at Facebook in 2005

- Contributed back to open source in 2007 (became root repository)

- Memcached is a general-purpose distributed in-memory caching system
  - Used to speed up dynamic database-driven websites.
  - Caches data and objects in RAM to reduce the number of times an external data source (like MySQL) gets hit.
  - Operations include get, set, incr, decr, multiget, multiset.

- Pros: Reliable, low-latency, extremely high service rates

- Cons: Easy to corrupt, limited data model, inefficient for small items

# Memcached

## Scaling Facebook

**Load Balancer (assigns a web server)**

**Web Server (PHP, assembles data)**

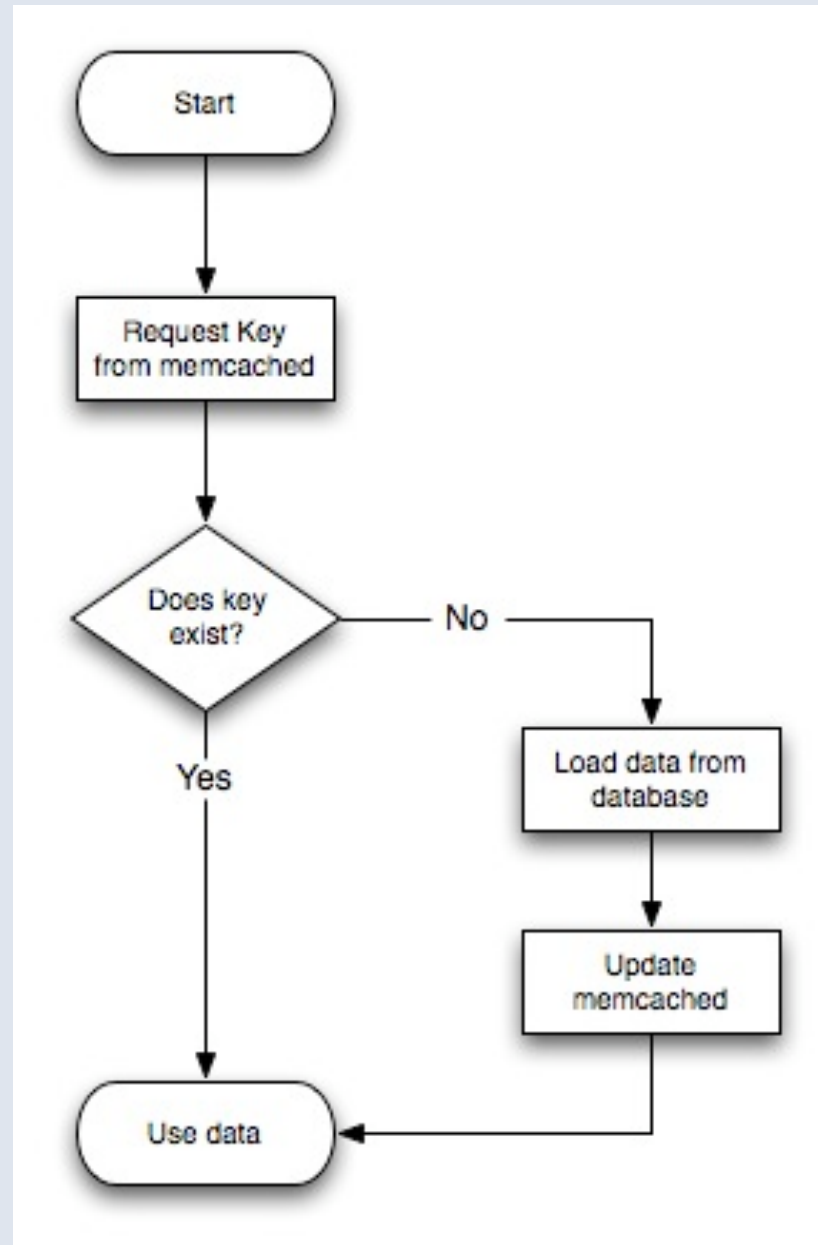**Services (specialized)**

**Memcached (fast)**

**Databases (persistent)**
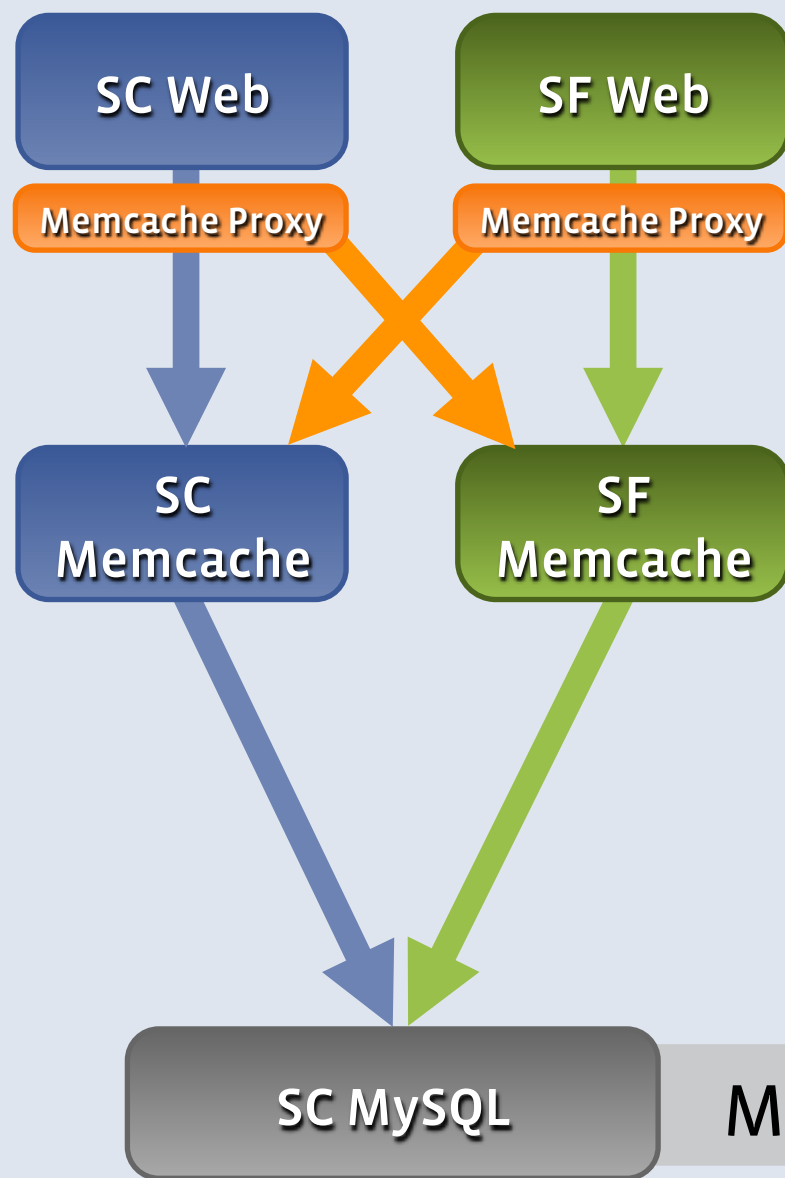
# Memcached

## Flow Chart

# Memcached Improvements

- 64-bit port
- More efficient serialization
- Multithreading
- Improved protocol
- New network drivers
- Compression
- Network stack optimizations
- UDP
- Memache Proxy
- Many others...

# Scaling across multiple datacenters

**West Coast**

**East Coast**

SC Web

SF Web

VA Web

Memcache Proxy

Memcache Proxy

SC Memcache

SF Memcache

VA Memcache

Memcache Proxy

SC MySQL

MySql replication

VA MySQL
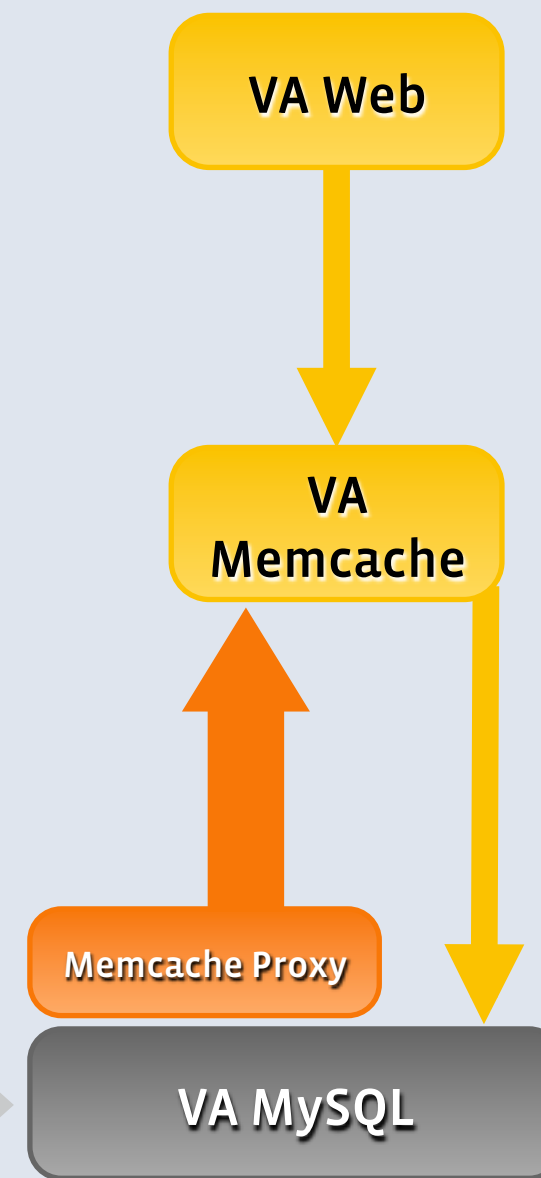
# Memcached

- Facebook's main trick to scaling

- http://github.com/facebook
  - Try it!
  - Improve it!
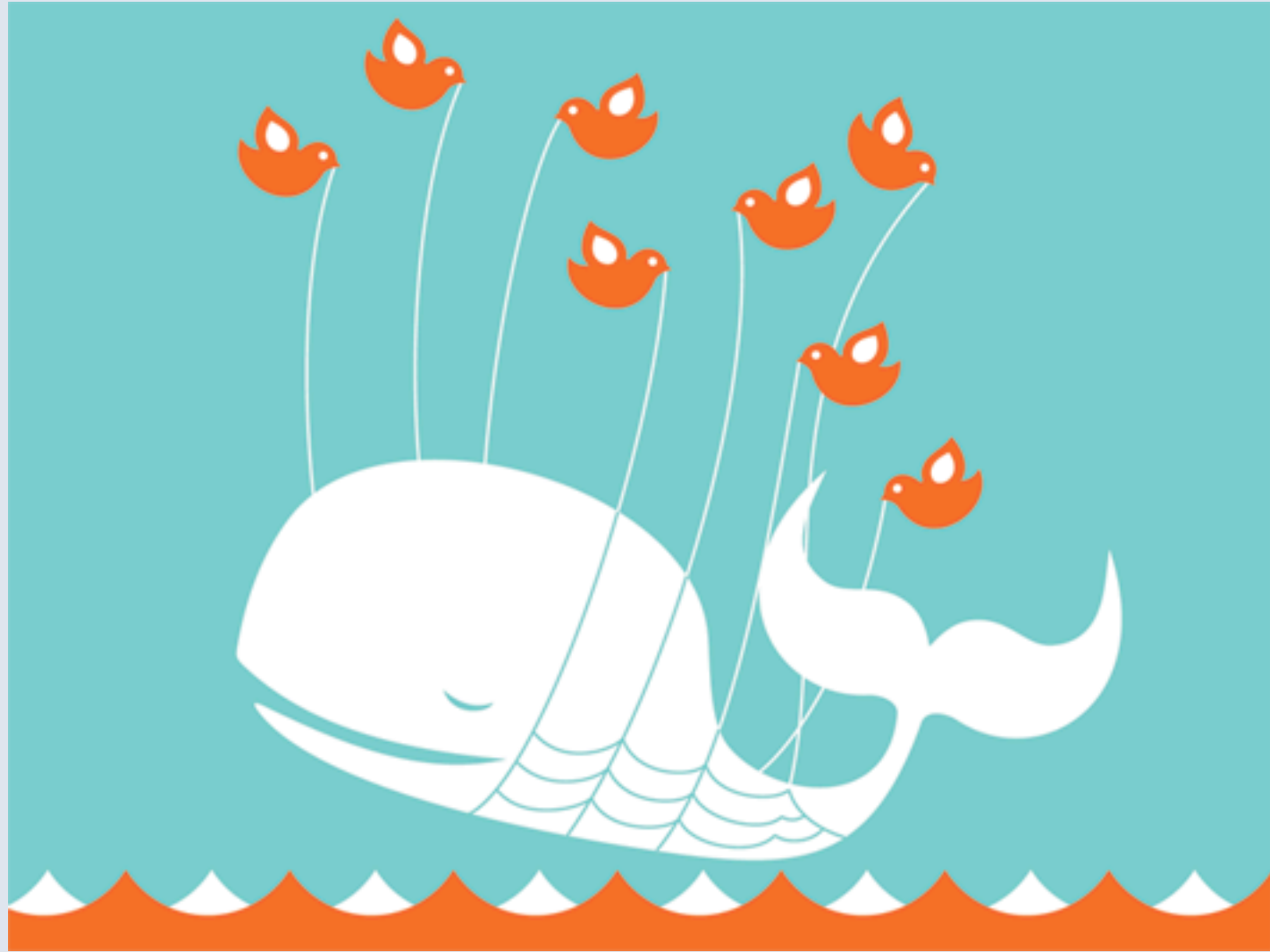
# Thrift

# The Problem

- The website scripting language of your choice can do the following well:

  - Handle large amounts of web requests

  - Fetch data (from mySQL, memcached, etc…)

  - Manipulate small amounts of data

  - Construct markup

  - Easier to debug, no compile time

- But it cannot do the following very well:

  - Handle large amounts of complex computation

  - Manipulate huge amounts of data MBs / GBs

  - Store / build a large index

  - Search a large index of data

# The Problem

- Your homegrown service (written in C/C++/Java/etc) can:

  - Handle large amounts of complex computation

  - Manipulate huge amounts of data MBs / GBs

  - Store / build a large index

  - Search a large index of data

- But it cannot do the following very well:

  - Handle web requests

  - Construct markup

# Typical Solution

- Divvy up the work!

- Write a C/C++/Java/etc server to do the computationally intensive stuff...

- Pass the result back to your web script to do the rest.
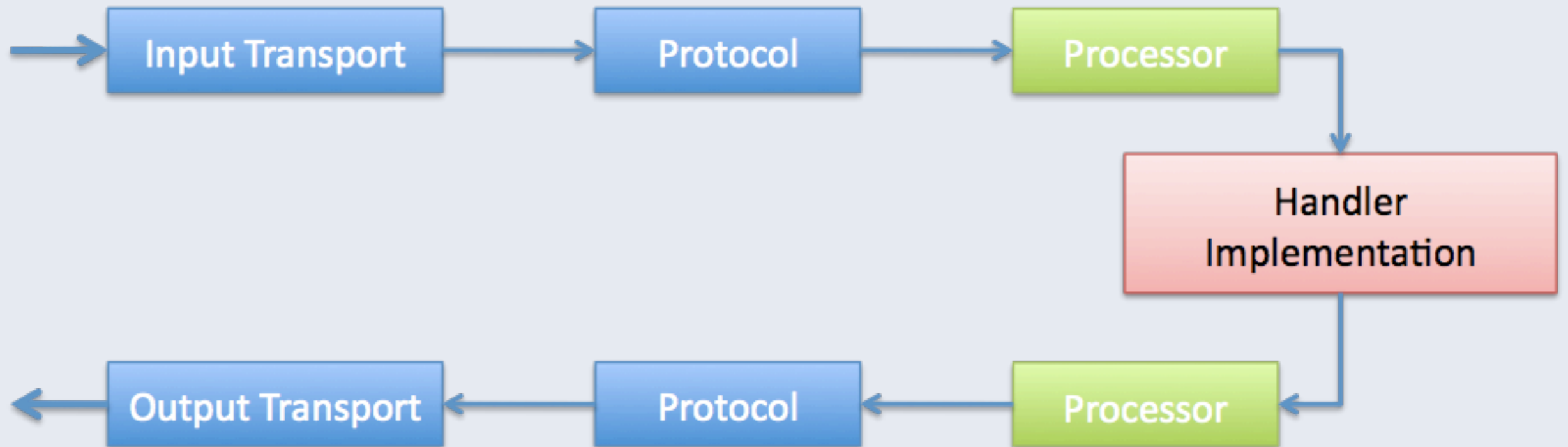
# Challenges

- Web and Service tiers are in different languages!

  - How to pass data back to web tier?

  - Code duality is very fragile.

- Challenging to make service tier handle a large qps from web tier.

  - Concurrency / multi-threading headaches

# Our Answer: Thrift

- Developed @ Facebook late-2005

- Open-sourced April 1st 2007

- Thrift is an remote procedure call (RPC) framework.
  - Makes executing RPC calls across languages easily.
  - Advanced serialization for many data types. (MC, DB blobs, etc.)
- High-performance server framework for C++ (100k+ queries per sec)
- Thrift writes its own code!

# Thrift Data Flow (Server)

Input Transport → Protocol → Processor → Handler Implementation

Handler Implementation → Processor → Protocol → Output Transport

Key: Thrift Lib | Generated | You Write This

# Thrift

- You: Define a .thrift file
  - Service interface
    - Functions
    - Parameters
    - Return types
  - Data structures
- Thrift compiler: Generates client AND server stubs
  - Up to 13(+!) languages
- You: Code actual server logic (in C++ / Java / C#)
- You: Code client result handling (in PHP, Python, Ruby, etc...)

# Thrift
## .thrift Interface File

```thrift
// math.thrift
struct DivideRequest {
  1: i64 numerator,
  2: i64 denominator,
}

struct DivideResponse {
  1: i64 quotient,
  2: i64 remainder,
}

exception DivideException {
  1: string reason,
}

service MathService extends fb303.FacebookService {
  DivideResponse divide(1: DivideRequest request)
                      throws (DivideException e)

  i64 add(1: list<i64> numbers)
  i64 multiply(1: list<i64> numbers)
}
```

# Thrift

## Magic

"thrift math.thrift -cpp -php"

# Thrift

## Server Code

```cpp
void MathServiceHandler::divide(
                    DivideResponse& response,
                    const DivideRequest& request) {
  if (request.denominator == 0) {
    DivideException e;
    e.reason = "Division by zero";
    throw e;
  }

  response.quotient = request.numerator /
                      request.denominator;
  response.remainder = request.numerator %
                       request.denominator;
}
```

# Thrift
## Server Code

```cpp
....
....
// set up our thrift stack
shared_ptr<MathServiceHandler> handler(
            new MathServiceHandler());
shared_ptr<TProcessor> processor(
            new MathServiceProcessor(handler));
shared_ptr<TServerTransport> transport(
            new TServerSocket(FLAGS_port));
shared_ptr<TTransportFactory> transportFactory(
            new TBufferedTransportFactory());
shared_ptr<TProtocolFactory> protocolFactory(
            new TBinaryProtocolFactory());

TSimpleServer server(processor, transport,
                    transportFactory, protocolFactory);

// start serving
server.serve();
....
....
```

# Thrift
## Client Code

```php
$socket = new TSocket(                    , 44444);
$transport = new TBufferedTransport($socket);
$protocol = new TBinaryProtocol($transport);
$math_client = new MathServiceClient($protocol);
try {

    $request = new DivideRequest(120, 4);
    $response = $math_client->divide($request);

} catch (DivideException $e) {
    debug_rlog('Caught DivideException: ' . $e->reason);
} catch (Exception $e) {
    debug_rlog('Caught other exception: ' . $e);
}
```

# Thrift

- Takes care of all the tedious stuff.

  - Handles low-level network operations and data serialization

  - Many multi-threading features are not in standard C++ lib

  - Thrift simplifies usage of threads & pools, mutexes, etc.

- Significantly simplifies writing your server code.

- Seamlessly gives your front-end code the power it needs.


- http://github.com/facebook

  - Try it!

  - Improve it!

# XHP

# HTML

# HTML

# HTML

facebook

<input />

Home   Profile   Account

Stefan <h2 /> Parker

<button />

Facebook · Trinity U

<img />

**Suggestions**

Ruchi Sanghvi
39 mutual friends
Add as friend

Carol Pai
30 m<div />
Add as friend

Current Activity

What's on your mind?

Attach:

Share

**Sponsored**          Create an Ad

theKbuzz | Social Media and Word o...

Become a Fan

Become a fan and learn how we put companies and governments on this site, the best platform ever, or call us at 718.416.2899.

Scott Chun-Yang Chen likes theKbuzz | Social Media and Word of Mouth Marketing.

Like

Wall

<a /> Back in Texas and missing y'all already. Thanks to you and Christina for everything, see you in June, I hope! Give Squish a hug for me!

Yesterday at 6:5 <span /> ebook for iPhone · Comment · Like

See Wall-to-Wall

**Shea Parker** you should pre-order this.

Hulu – Onion News Network: Apple Introduces Revolutionary New Laptop With No Keyboard

www.hulu.com

Video description: The MacBook

**The Art Institutes**

Make us the first stop on the path to the filmmaking career you really want. Learn more about courses offered in Sunnyvale.

Like

Profile
Stream
Friends
Photos
Notes

<ul />

In a Relationship with

Christina Gray

# HTML
## Problems

## Semantics

- What's wrong with this?

```
<span id="test1" class="uiTextTitle">
  <img src="/images/image.png" />
  <div id="inner">
    Text content
  </div>
</span>
```

# HTML

## HTML
### Problems

```
<span id="test1" class="uiTextTitle">
  <img src="/images/image.png" />
  <div id="inner">
    Text content
  </div>
</span>


<span>
  <div />
</span>
```

✗

# CSS

# HTML
## Problems

▪ Rule of 1.  If a browser has over 1% market share, we officially support it.

  ▪ IE 6/7/8, Firefox 3, Chrome 4, Safari 4



IE 7/8              Firefox              IE 6              Safari

# HTML
## Browser Hacks

- How to target CSS to specific browsers

  - IE 6 "underscore rule": `.component { _width: 100%; }`

  - IE 6 & 7 "star rule": `.component { *width: 100%; }`

  - IE 8, 7, & 6 "slash 9 rule": `.component { width: 100%\9; }`

  - Firefox 3: `.ff3 .component { width: 100%; }`

  - Safari 4/Chrome 4: `.safari4 .component { width: 100%; }`

Wednesday, May 19, 2010

# HTML
## Problems



...

# Our Solution: XHP

- XHP is a PHP extension, which augments the syntax of the language such that XML document fragments become valid PHP expressions.
  - Makes front-end code more readable.
  - Prevents invalid / erroneous markup.
  - Prevents against cross-site scripting (XSS) attacks.
  - XML document fragments are like objects.
    - You can define your own elements, making XHP a templating engine.
    - Inheritance can be used for more reusable code.
      - Allows you to define complex components in simple XHP tags.
      - Encourages cleaner code, more readable code.
    - Attribute validation

# XHP
## Simple PHP Example

```php
<?php
if ($_POST['name']) {
  echo '<span>Hello, '.$_POST['name'].'.</span>';
} else {
  echo
    '<form method="post">
      What is your name?<br />
      <input type="text" name="name" />
      <input type="submit" />
    </form>';
}
```

# XHP

## Simple XHP Example

```php
<?php
if ($_POST['name']) {
  echo <span>Hello, {$_POST['name']}</span>;
} else {
  echo
    <form method="post">
      What is your name?<br />
      <input type="text" name="name" />
      <input type="submit" />
    </form>;
}
```

# XHP
## Preventing markup errors

```php
<?php
if ($_POST['name']) {
  echo '<span>Hello, '.$_POST['name'].'.</span>';
} else {
  echo
    '<form method="post">
      What is your name?<br />
      <input type="text" name="name" />
      <input type="submit" />
    <form>';
}
```

# XHP
## Preventing markup errors

# XHP
## Preventing markup errors

```php
<?php
if ($_POST['name']) {
  echo <span>Hello, {$_POST['name']}</span>;
} else {
  echo
    <form method="post">
      What is your name?<br />
      <input type="text" name="name" />
      <input type="submit" />
    <form>;
}
```

# XHP

## Preventing markup errors

```php
<?php
if ($_POST['name']) {
  echo <span>Hello, {$_POST['name']}</span>;
} else {
  echo
    <form method="post">
      What is your name?<br />
      <input type="text" name="name" />
      <input type="submit" />
    <form>;
}
```
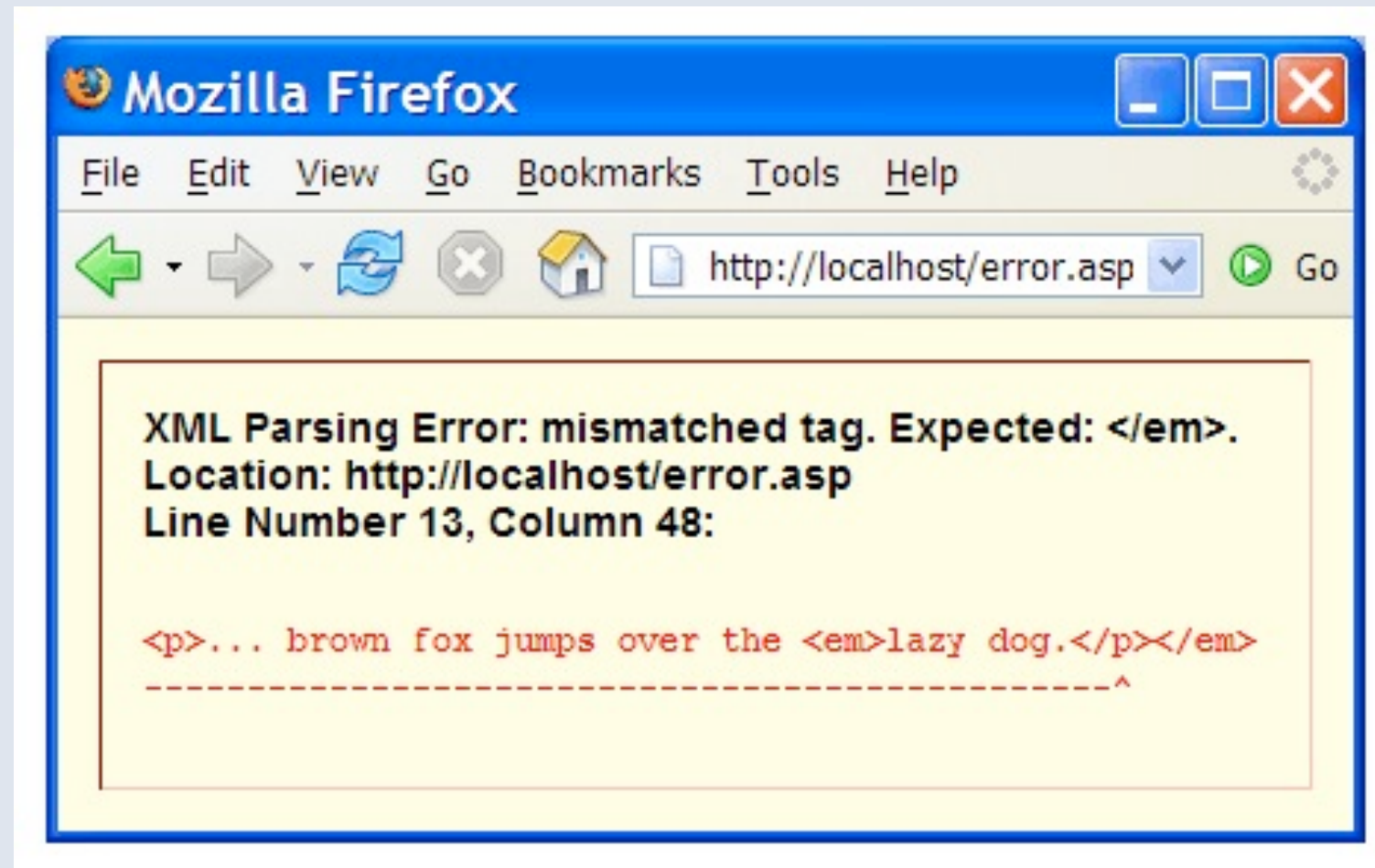
# XHP
## Preventing markup errors

```php
<?php
  echo
    '<span>'.
      'Hello, '.
      '<div>'.
        $_POST['name'].
      '</div>'.
    '</span>';
```

# XHP
## Preventing markup errors

```php
<?php
  echo
    <span>
      Hello
      <div>
        {$_POST['name']}
      </div>
    </span>;
```

# XHP
## Preventing markup errors

```php
<?php
echo
    <span>
        Hello
        <div>
            {$_POST['name']}
        </div>
    </span>;
```

# XHP
## Cross Site Scripting Prevention

```php
<?php
if ($_POST['name']) {
  echo '<span>Hello, '.$_POST['name'].'.</span>';
} else {
  echo
    '<form method="post">
      What is your name?<br />
      <input type="text" name="name" />
      <input type="submit" />
    </form>';
}
```

**What happens if $_POST['name'] = '<script src="http://ha.ckers.org/xss.js"></script>'?**

# XHP
## Cross Site Scripting Prevention

```php
<?php
if ($_POST['name']) {
  echo '<span>Hello, '.htmlspecialchars($_POST['name']).'.</span>';
} else {
  echo
    '<form method="post">
      What is your name?<br />
      <input type="text" name="name" />
      <input type="submit" />
    </form>';
}
```

# XHP
## Cross Site Scripting Prevention

```php
<?php
if ($_POST['name']) {
  echo <span>Hello, {$_POST['name']}</span>;
} else {
  echo
    <form method="post">
      What is your name?<br />
      <input type="text" name="name" />
      <input type="submit" />
    </form>;
}
```

# XHP

## Context-aware markup



IE 7/8    Firefox    IE 6    Safari

Wednesday, May 19, 2010

```
class :fb:wide-div extends :fb:div {
  attribute
    int width = 100;

  category %div;

  protected function compose() {
    return <div width={$this->getWidth()} />;
  }

  private function getWidth() {
    if ($this->width != 100) {
      return $this->width;
    }

    return $this->browser->isIE6() ?
      '*100%' : '100%';
  }
}
```

# Templating Engines

- Templating Engine acts as the "glue" between your data and your markup.

- Pros:

  - Data logic is completely separate from display.

    - Makes it much easier to read both code and markup.

    - Simplicity prevents errors.

  - Markup can be easily changed w/o code change.

- Cons:

  - Intermediary template engine step has inherent limitations.

  - Extensibility is limited.

  - Markup reuse is limited.
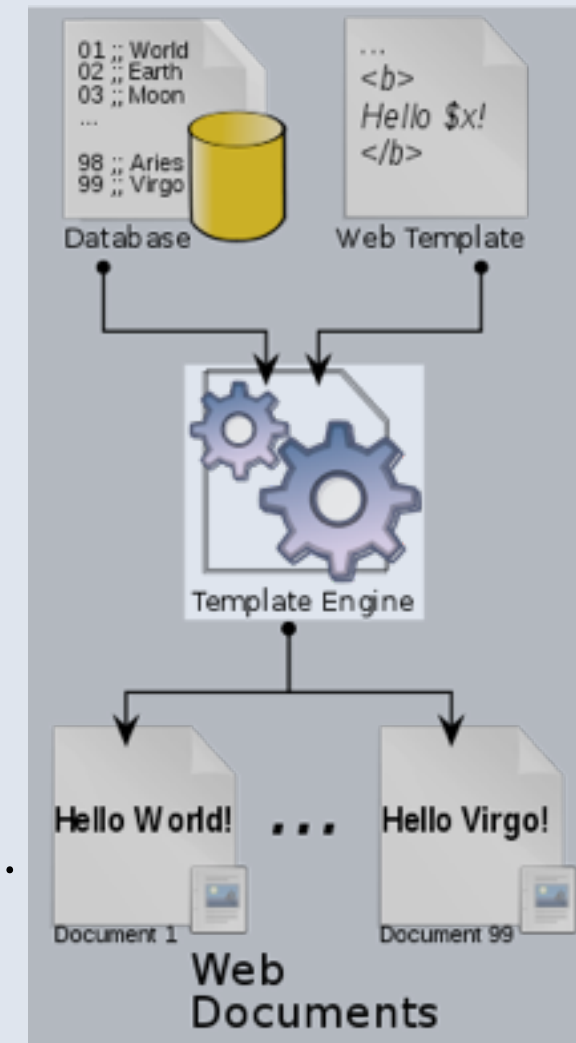
# XHP as a pseudo-Templating Engine

- XHP itself looks just like markup, making it easy to edit.

- Markup render code lives within XHP elements, separating from data code.

- XHP elements can be built from other XHP elements.

  - They can render very powerful / complex HTML.

  - But also exist simply as an element.

- XHP elements can subclass + extend other XHP elements too.

  - This is also extremely powerful.

  - Encourages code reuse.

- No intermediary template engine required!

# XHP
## As a pseudo-templating engine

```
class :fb:like-link extends :fb:link {
  attribute
    int objectid @required,
    enum color {'blue', 'dark blue'} = 'blue';

  category %link;

  protected function compose() {
    return
      <fb:link href={$this->getHref()}
               color={$this->color}>
        Like
      </fb:link>
  }

  private function getHref() {
    return '/like.php?id='.$objectid;
  }
}
```

Like

# XHP

## As a pseudo-templating engine



```
class :fb:like extends :x:element {
  attribute
    int objectid @required;

  protected function compose() {
    return
      <fb:hovercard id={$this->objectid}>
        <fb:image img="likethumb" />
        <fb:like-link id={$this->objectid}>
        <fb:like-count id={$this->objectid}>
      </fb:hovercard>;
  }

  ...
}
```

# XHP
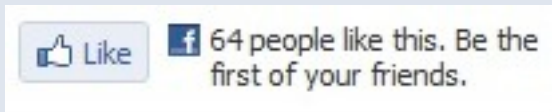## As a pseudo-templating engine



```
<fb:frame>
  <fb:nav />
  <fb:panel pos="left">
    <fb:quicklinks />
    <fb:chatbar />
  </fb:panel>
  <fb:panel pos="center">
    <fb:composer />
    <fb:newsfeed />
  <fb:panel pos="right">
    <fb:events />
    <fb:pymk />
    <fb:ad size="large" />
    <fb:questions />
    <fb:connected />
  </fb:panel>
  <fb:footer />
</fb:frame>;
```

# XHP

- Better templating system for webpage development.

- Promotes object-oriented markup design.

- Extremely versatile.


- http://github.com/facebook

  - Try it!

  - Improve it!

# Ads
## Purchase Funnel

# Ads
## Purchase Funnel

Television, Radio, Newspaper →

Product Reviews, Consumer Reports, Talk to Friends →

Amazon, Mom & Pop shop, Toyota Dealer →

Initial Brand awareness

Brand engagement / interaction

Product discovery

Consideration / options

Product selection

Convert to sale

# Ads
## Purchase Funnel

Television, Radio, Newspaper → 

→ Initial Brand awareness

Brand engagement / interaction

Product discovery

Product Reviews, Consumer Reports, Talk to Friends →

Consideration / options

Product selection

Amazon, Mom & Pop shop, Toyota Dealer →

Convert to sale

← ~$100s Billions a year

← ~$10s Billions a year

# Ads
## Purchase Funnel

Television, Radio, Newspaper →

← facebook

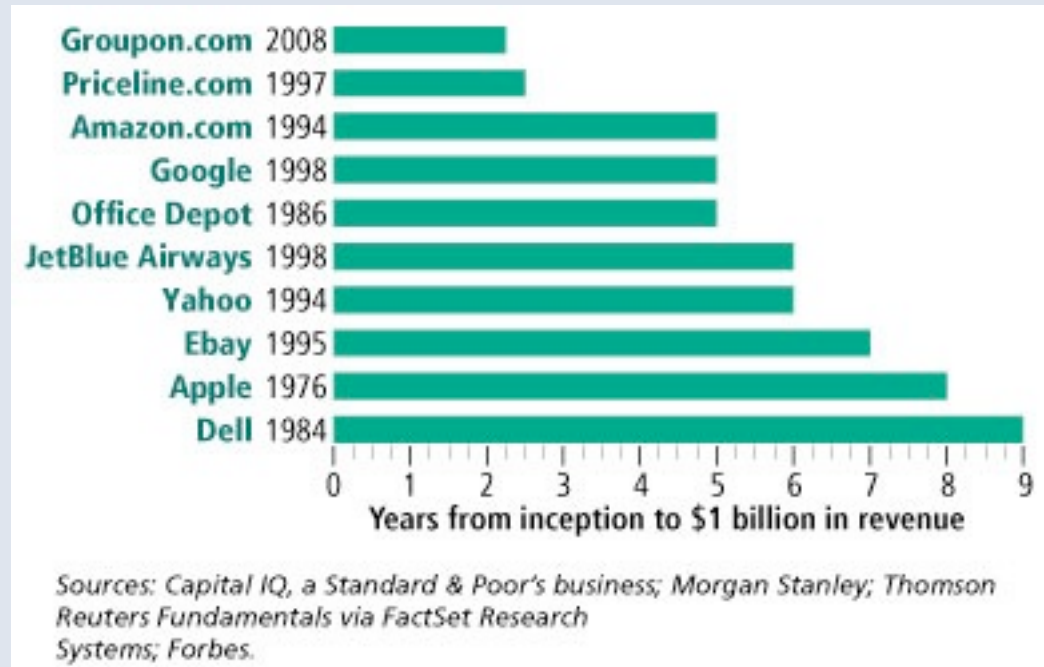Product Reviews, Consumer Reports, Talk to Friends →

**Initial Brand awareness**

**Brand engagement / interaction**

**Product discovery**

**Consideration / options**

**Product selection**

**Convert to sale**

Amazon, Mom & Pop shop, Toyota Dealer →

← Google

# Ads
## Purchase Funnel

What is the
fastest growing
company EVER?
(to $1B in revenue)



Groupon.com 2008
Priceline.com 1997
Amazon.com 1994
Google 1998
Office Depot 1986
JetBlue Airways 1998
Yahoo 1994
Ebay 1995
Apple 1976
Dell 1984

Years from inception to $1 billion in revenue

Sources: Capital IQ, a Standard & Poor's business; Morgan Stanley; Thomson Reuters Fundamentals via FactSet Research Systems; Forbes.

# Ads
## Purchase Funnel

# Facebook Ads

## Features

Powerful demographic targeting

Bid estimation/suggestion

Campaign optimization

www.facebook.com/ads/

# Facebook Ads

Emerging Demographic Segments

- US 18+ eCPM ~$0.27
- Weddings
  - Engaged females age 22-28
    - eCPM ~$0.40
- Recruiting
  - People who work at Google
    - eCPM ~$0.42
- You can play around with this yourself:
  - http://www.facebook.com/ads/create/



Great Engineers Wanted
**Linked in**
...ting for great ...engineers, ...those with

Web Engineer
**mystrands**

Engineers wanted ...
Internet personali...
company offering...
services and
marketplaces over...
AI middleware for
mobile–fixed arch...

Web App Engineer

Join a smart, collegial, web engineering team using Ruby, Rails, css, Javascript and html. Excellent pay and amazing location.

# Facebook Ads
## Auction Model

- 2nd Price Auction "Vickrey Auction"
  - Google, Bing, etc… use this auction model

- Optimal for seller when there are multiple quantities of an item available.
- Bids are "blind".
- Bidders have much less incentive to bid low.
- Auction "winners" pay the 2nd highest price for an auction.

# Facebook Ads
## Auction Model

| Advertiser | Max CPC |
|------------|---------|
| Mallory | $0.40 |
| Mark | $0.50 |
| Laura | $0.20 |
| Cameron | $2.00 |
| Alison | $0.05 |
| Will | $0.10 |

# Facebook Ads
## Auction Model

| Advertiser | Max CPC | Quality Score |
|------------|---------|---------------|
| Mallory | $0.40 | 4 |
| Mark | $0.50 | 3 |
| Laura | $0.20 | 6 |
| Cameron | $2.00 | 0.5 |
| Alison | $0.05 | 16 |
| Will | $0.10 | 2 |

# Facebook Ads
## Auction Model

| Advertiser | Max CPC | Quality Score | AdRank |
|---|---|---|---|
| Mallory | $0.40 | 4 | $0.4 x 4 = **1.6** |
| Mark | $0.50 | 3 | $0.50 x 3 = **1.5** |
| Laura | $0.20 | 6 | $0.20 x 6 = **1.2** |
| Cameron | $2.00 | 0.5 | $2.00 x .5 = **1** |
| Alison | $0.05 | 16 | $.05 x 16 = **.8** |
| Will | $0.10 | 2 | $0.10 x 2 = **.2** |

# Facebook Ads
## Auction Model

| Advertiser | Max CPC | Quality Score | AdRank | Position | Actual CPC |
|---|---|---|---|---|---|
| Mallory | $0.40 | 4 | $0.4 x 4 = **1.6** | 1 | (1.5 / 4) + $.01 = $.39 |
| Mark | $0.50 | 3 | $0.50 x 3 = **1.5** | 2 | (1.2 / 3) + $.01 = $.41 |
| Laura | $0.20 | 6 | $0.20 x 6 = **1.2** | 3 | (1 / 6) + $.01 = $.17 |
| Cameron | $2.00 | 0.5 | $2.00 x .5 = **1** | 4 | (.8 / .5) + $.01 = $1.61 |
| Alison | $0.05 | 16 | $.05 x 16 = **.8** | 5 | (.2 / 2) + $.01 = $.11 |
| Will | $0.10 | 2 | $0.10 x 2 = **.2** | 6 | Minimum Bid |

**Actual CPC = (AdRank to beat / Quality Score of Advertiser) + $.01**

# Facebook Ads



## Social Ads

- What's a social ad?
- Do they work?
  – Ads that have social data attached have 3-5x CTR compared to ads that do not have social data attached
- > 20% of all ads are eligible for social actions
- Very few user complaints

# Facebook Ads
## Alternative Ads

**Friends' Photo Albums**

**Winter 2010**
by Sonia Nagala

**Find More Friends**

**Wayne, More Friends Are Waiting**

These 7 friends found their friends using the friend finder. Have you found all of your friends? Give it a try.

Your Email

Email Password

**Find Friends**

**People You May Know**     See All

**Jessica Gronski**
16 mutual friends
Add as friend

**Recommended Pages**     See All

**Mountain Dew**
Chris Pan and 21 other friends like this.
Like

http://www.facebook.com/video/video.php?v=10150331790590484

# facebook