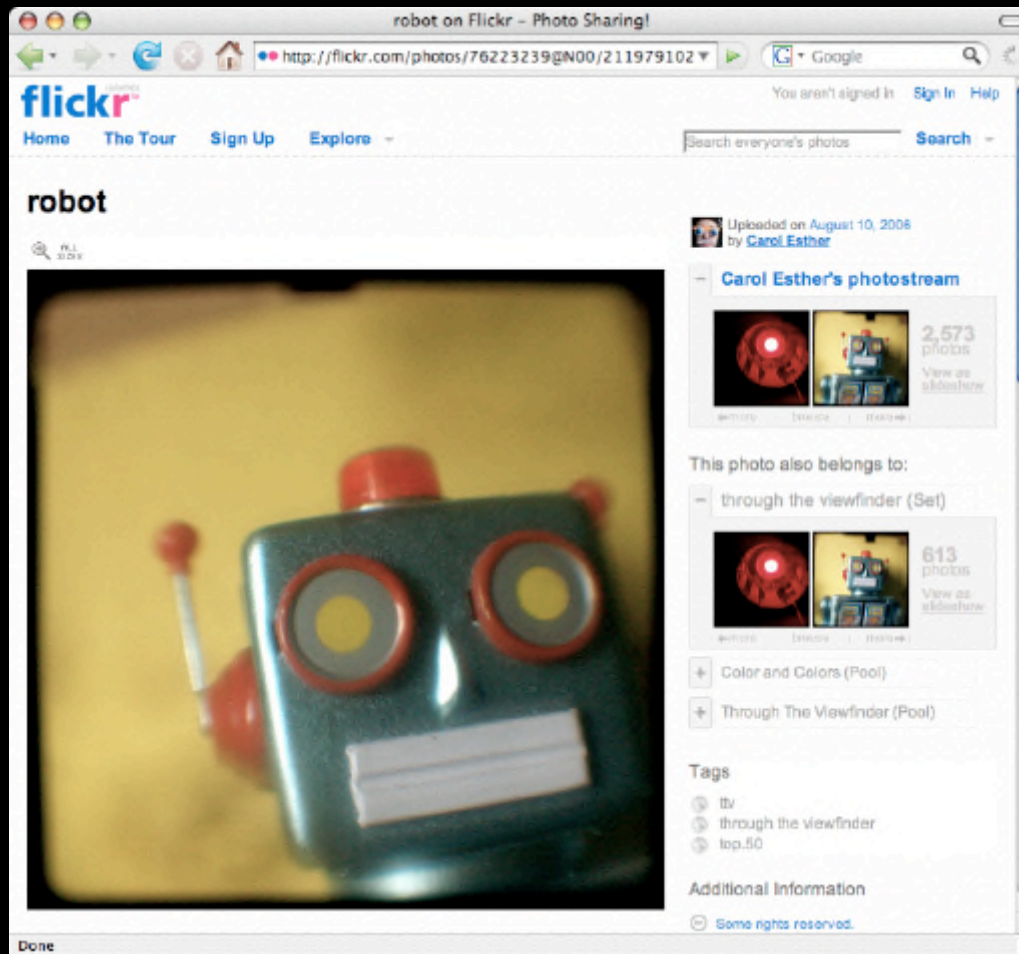


# Social Software for Robots

XTech 2007



**Kellan Elliott-McCrea**



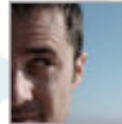
Flickr



**Blaine Cook**

**Thinking someone should adopt the tagline:  
"We put the point in web 2-point-oh."**

about 16 hours ago from txt ☆



Evan Williams

# Twitter

**For Robots?**

# Social Software

# Social Software is

- social software is made of people
- people talking to people?
- people talking to computers, talking to people.

flickr™

facebook



myspace.com®  
a place for friends



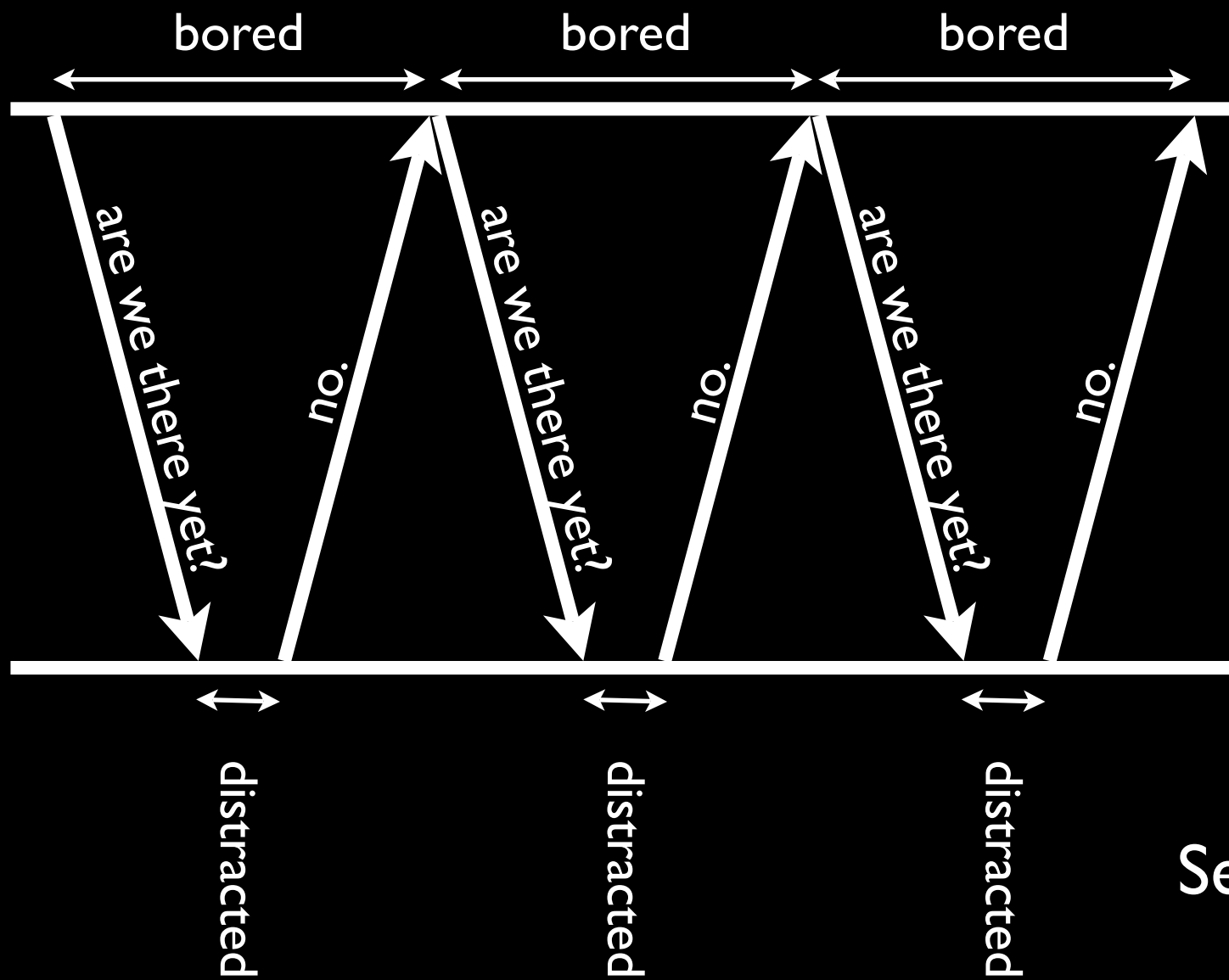
**Software to  
collect and share  
individuals' actions  
in a  
social context.**





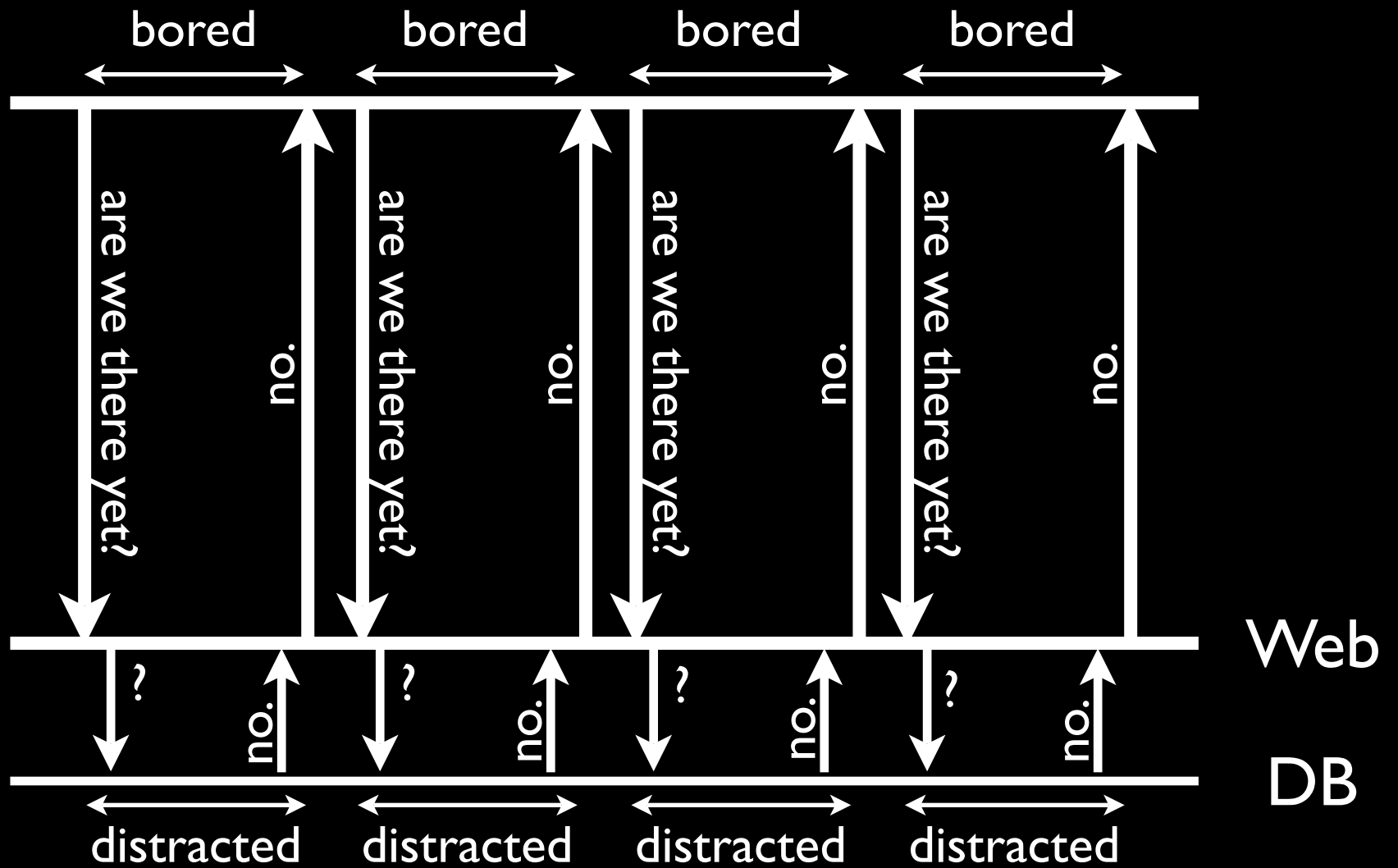
web apis

Client



Server

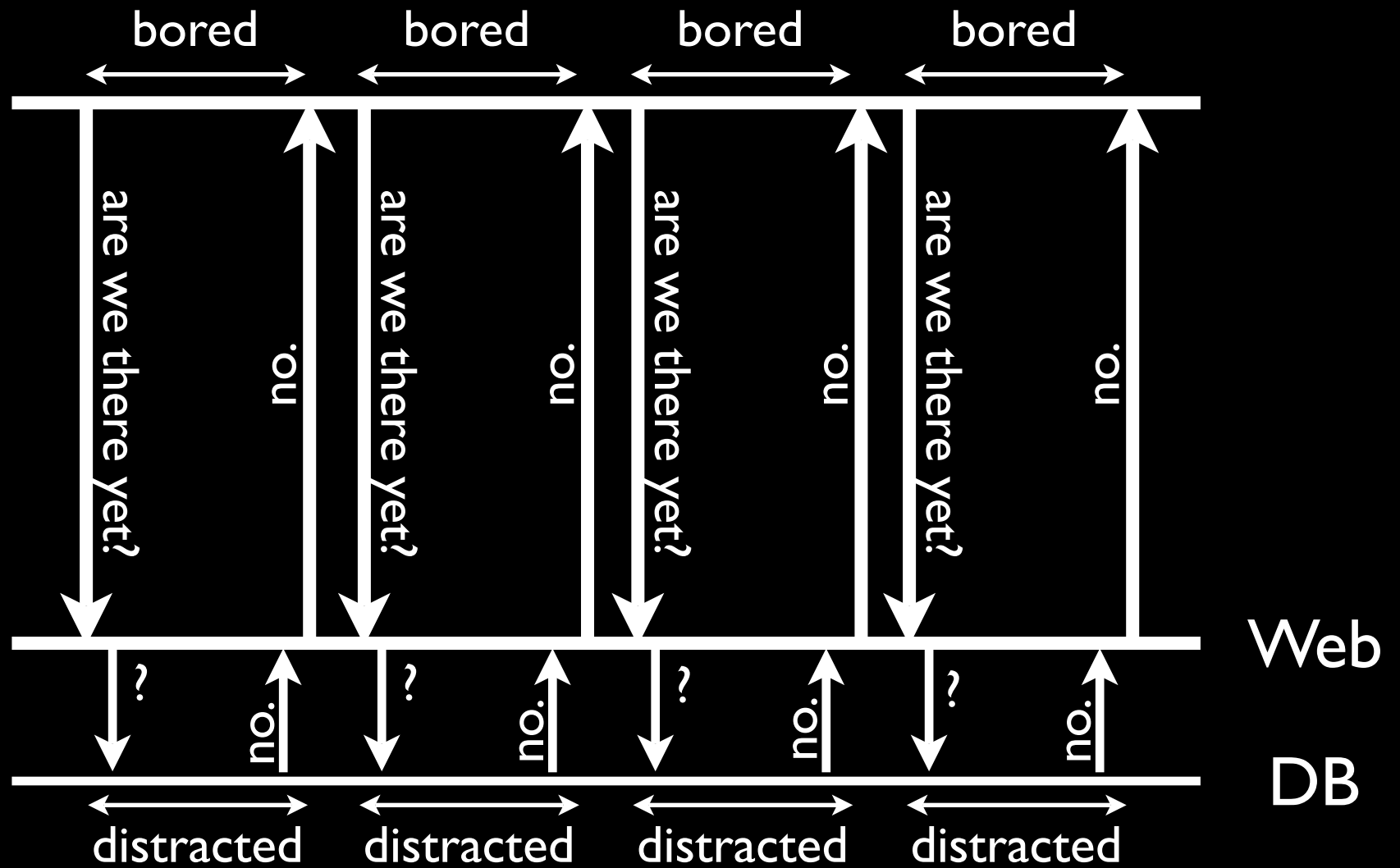
Client



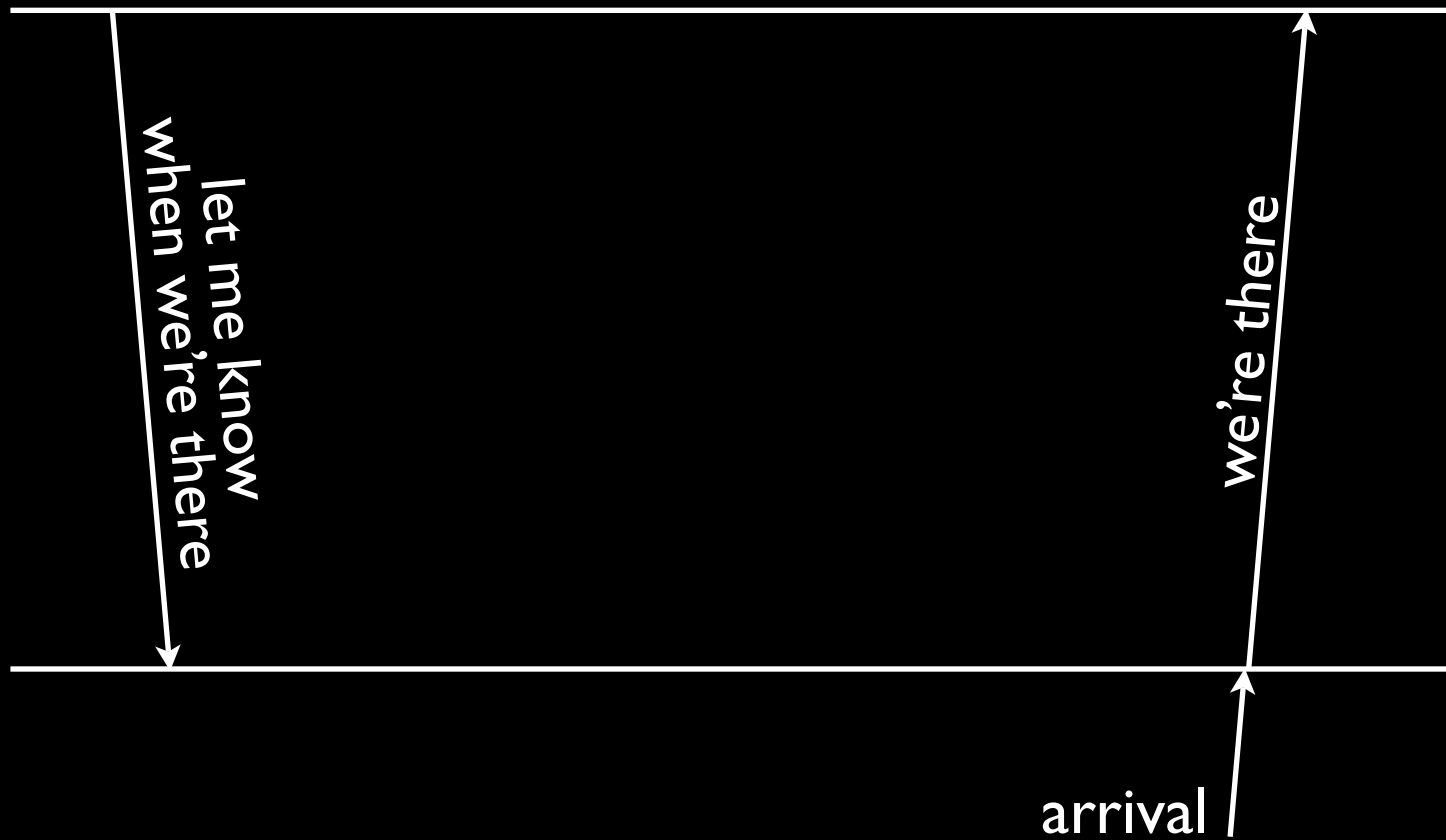




Client



Client



Server

PASSING NOTES  
YOU DID THIS WHEN YOU WERE EIGHT.  
IT'LL BE OKAY.

# Social Software For Robots

**Robots do it with  
Message Passing**

# Where it Works.

- real time data
- wire-level data  
(no db interaction)
- shift from access-past  
to notify-future
- very long  
running queries
- historical and static  
content - not so much.
- interactive searches -  
not so much

# New Patterns

(how else can you use it?)

time



Realtime

# Spontaneity

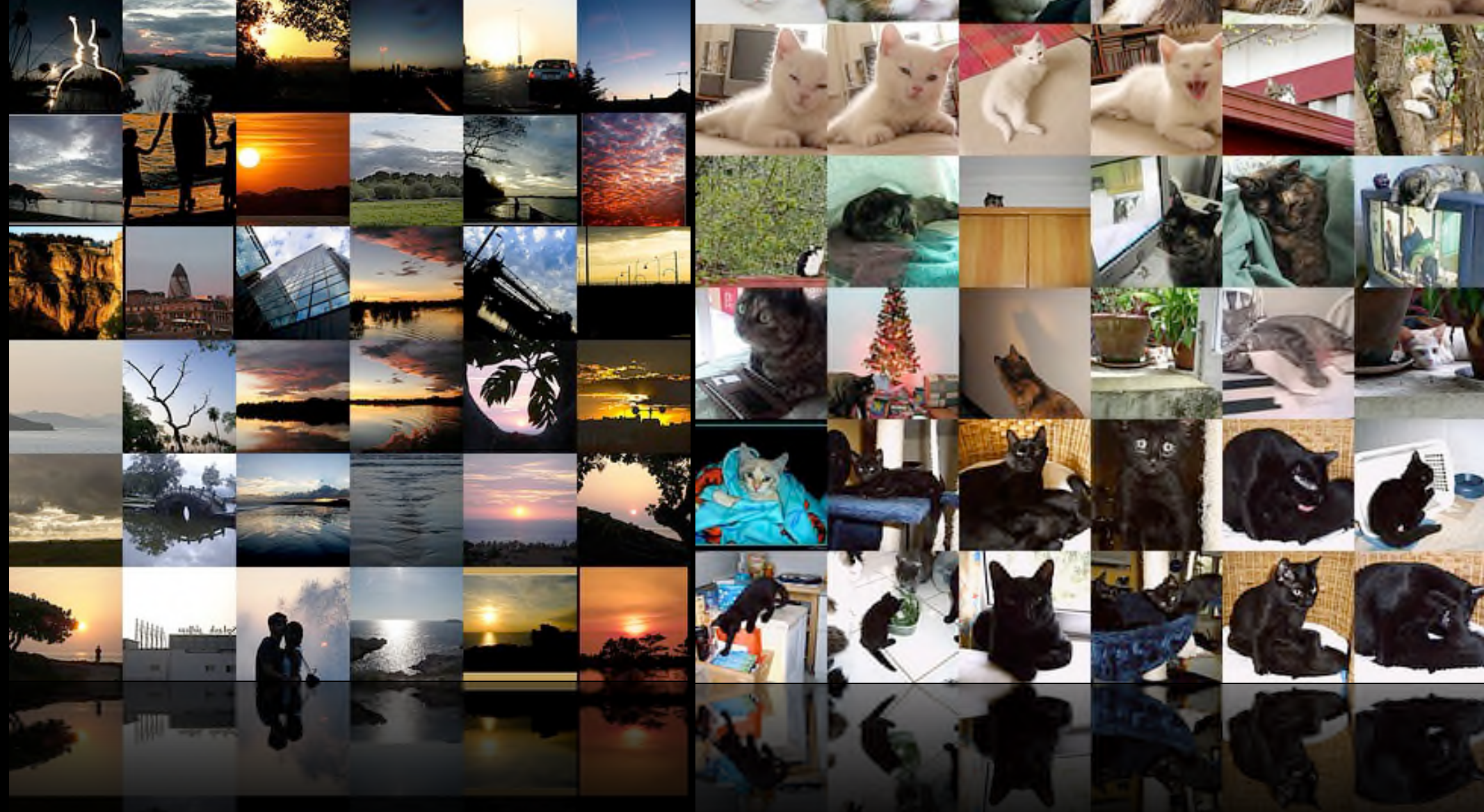
**The Manu Chao tickets arrived, but we can't go because i'm going to be in kenya and gaba's gonna be in boston.. duh!**

01:31 PM May 10, 2007 from im ☆



rabble





# Flickr TMI Addiction

# Fishing



- Put a net out, and get notified when you catch something.
- When real-time data is more important than knowing in advance or researching the future.
- Who's going for lunch **NOW?**

```
<message from="driver@local" to="car@local">  
PEN DOWN  
FORWARD 40  
RIGHT 65  
LEFT 25  
FORWARD 40  
</message>
```



Jabber



# What's in it for me?

- flexible & extensible
- asynchronous, low latency (realtime)
- delegated
- security & identity baked in
- presence
- standardized

# It's Just XML

```
<message  
  to="horatio@denmark"  
  from="hamlet@denmark">  
  <body>Angels and Ministers  
    of Grace, defend us!</body>  
  <foo  
    xmlns="http://www.foo.org/">  
    <bar>ab<fb/>cd</bar>  
  </foo>  
</message>
```

# Asynchronous

```
<message from="xena@gmail.com"  
to="sparql@w3c.org">meow!</message>
```



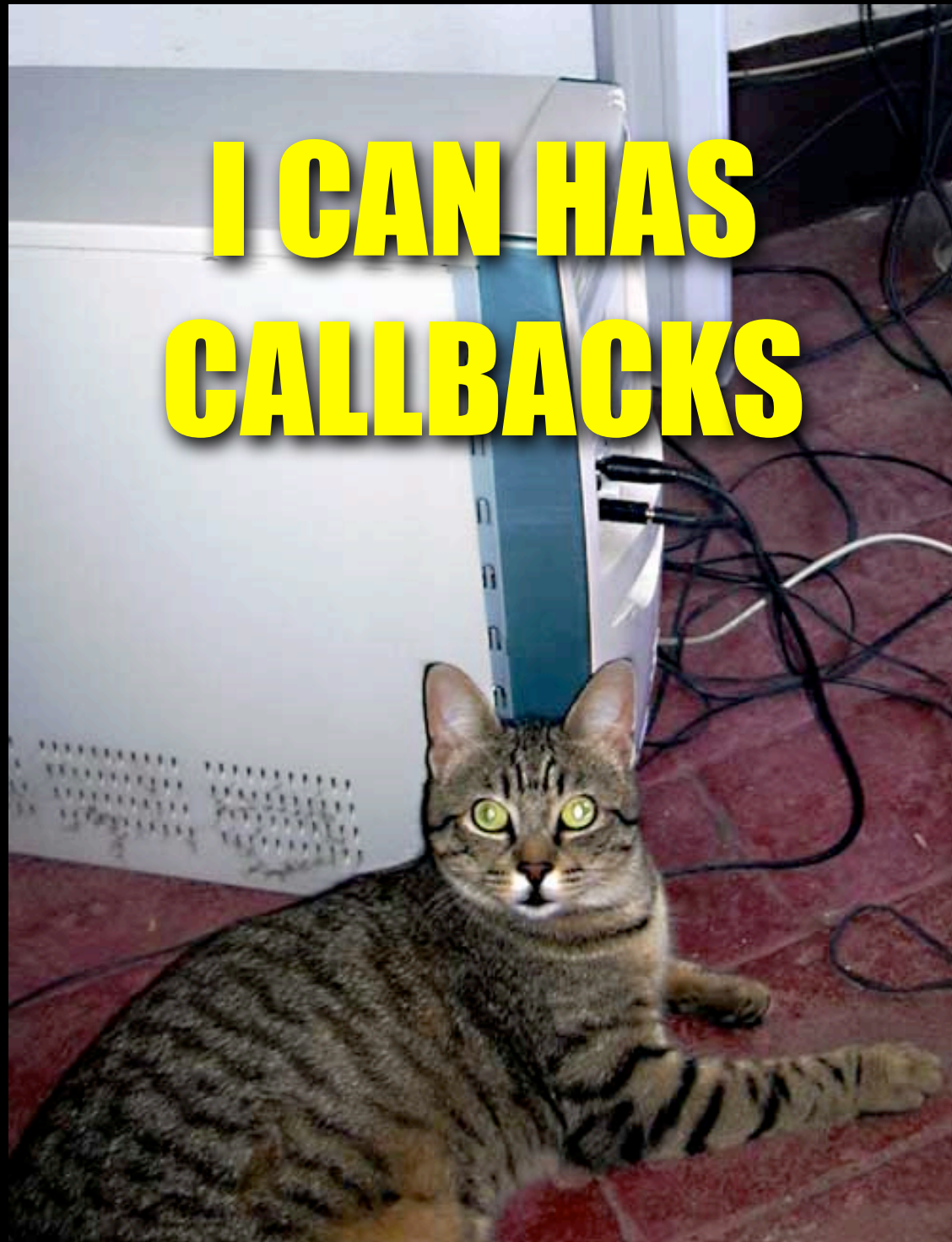
```
<message from="sparql@w3c.org"  
to="xena@gmail.com">meow!</message>
```

# Secure + Identity

- dialback server authentication (dns)
- s2s encrypted
- c2s encrypted
- from address set by server (no spoofing!)
- globally unique jids (like email)

# Delegated Delivery

**I CAN HAS  
CALLBACKS**



**Standardised**

# Open Sourcing

- ejabberd
- djabberd
- wildfire
- Imnop-jabberd
- twisted suite
- ruby
- php
- java
- perl
- python
- firefox

# Presence!

- Ralph is going to talk about this in the next talk. Y'all stick around now, y'hear?

# Decentralized, But not all the way.

- Smarts on the Edge-1 is what you want for ubicomp / device saturated world.

# that was a lot.

- async, real-time.
- extensible
- secure w/identity
- delegated
- i can has callbacks!
- standardized
- presence
- decentralized, but not p2p\*

**Robots**

# Twitter.

You use it every day. It's a Social Robot.

Use Jabber::Simple

```
require 'xmpp4r-simple'
```

```
jabber = Jabber::Simple.new  
        ("flickrit@gmail.com", "password")
```

send your tags to [flickrit@gmail.com](mailto:flickrit@gmail.com)!  
wifi is “icanhasinternet”

```
jabber.deliver("flickrviz@laughingmeme.org",  
               "hey there.")
```

send your tags to [flickrit@gmail.com](mailto:flickrit@gmail.com)!  
wifi is "icanhasinternet"

```
jabber.received_messages do |message|  
  puts message.body  
end
```

send your tags to [flickrit@gmail.com](mailto:flickrit@gmail.com)!  
wifi is “icanhasinternet”

```
require 'xmpp4r-simple'

CLIENT = Jabber::Simple.new("flickrit@gmail.com", "password")

def post_photo_update(message)
  CLIENT.deliver("flickrviz@laughingmeme.org", message)
end

def request_flickr_photos(message)
  CLIENT.deliver("flickrapi@gmail.com", "#{message.from} wants to see
photos of #{message.body}")
end

loop do
  CLIENT.received_messages do |message|
    case message.from.strip.to_s
    when 'flickrapi@gmail.com': post_photo_update(message.body)
    else request_flickr_photos(message)
    end
  end
end
```

pretty

send your tags to [flickrit@gmail.com](mailto:flickrit@gmail.com)!  
wifi is “icanhasinternet”

# Photo Credits

<http://www.flickr.com/photos/jayjizzle/205165885/>

<http://www.flickr.com/photos/heather/384134566/>

<http://flickr.com/photos/kevinsteele/473590084/>

<http://www.flickr.com/photos/exfordy/123370328/>

<http://flickr.com/photos/pingnews/426262117/>

**thanks!**

especially to tom from stamen. he totally rocks.