

]

Nifty web apps on an OpenResty

Nifty web apps on an OpenResty

☺ *agentzh@gmail.com* ☺

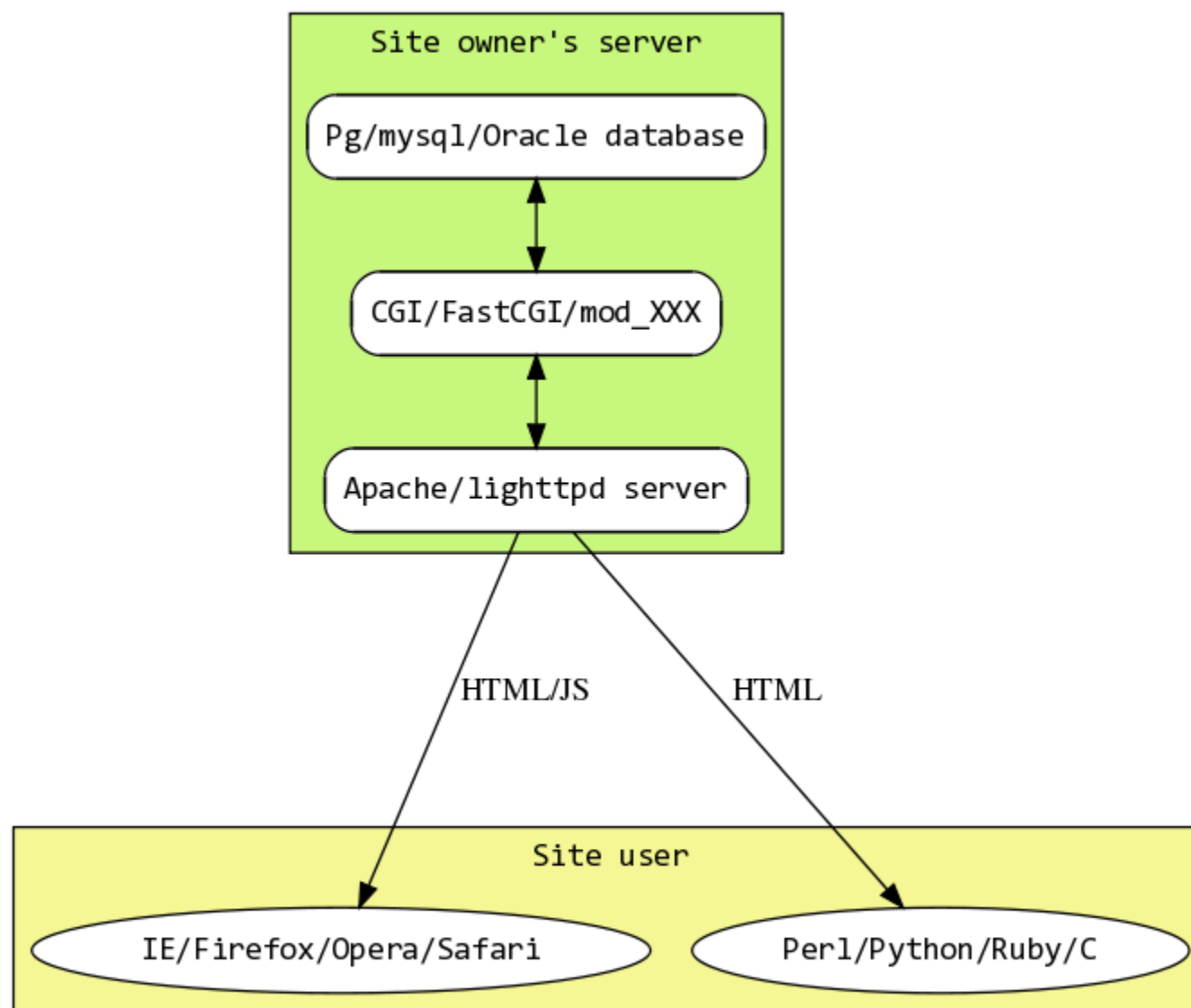
章亦春

2008.4

The *history* of the web...

The *history* of the web...

✓ **Web 1.0**



The good old Web 1.0

Benefits

- ☺ Server code is easy to write.
- ☺ Browsers are easy to build as well.
- ☺ Web 1.0 Spiders are happy.

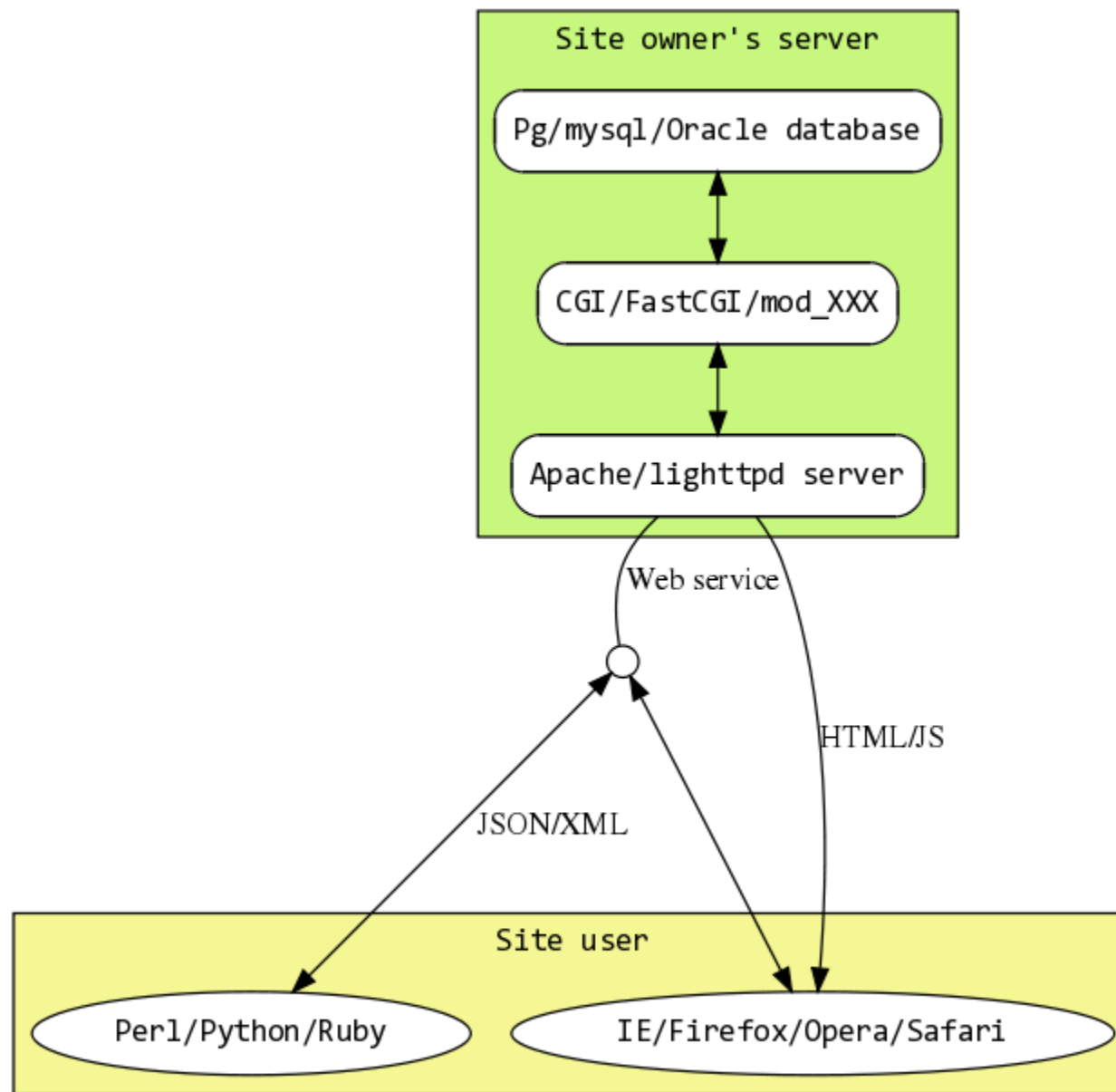
Drawbacks

- ☺ Pages *refresh* everytime the user takes an action.
- ☺ Advanced spiders try very hard to extract data from HTML source.
- ☺ Blog owner's server is *fat* and hard to scale

The *history* of the web...

✓ **Web 1.0**

✓ **Web 2.0**



The modern Web 2.0 era

Benefits

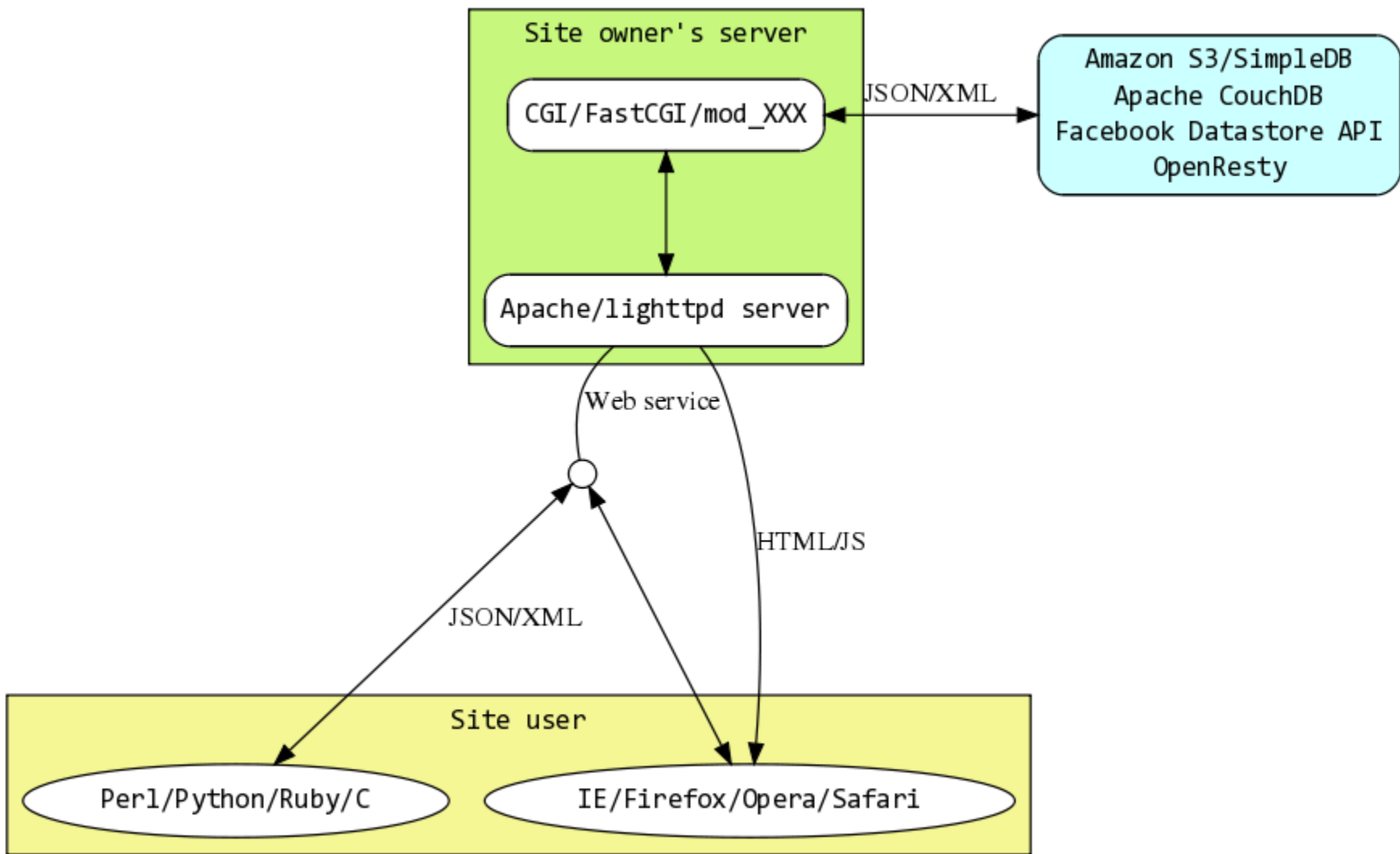
- ☺ Data can be applied to view templates on client side.
- ☺ Pages can be highly *responsive* via updating just page regions.

Drawbacks

- ☹ Server of the site owner is *very fat*, which is hard to deploy and hard to maintain.

The *history* of the web...

- ✓ **Web 1.0**
- ✓ **Web 2.0**
- ✓ **Post-mordern Web 2.0**



The post-modern Web 2.0 era

Benefits

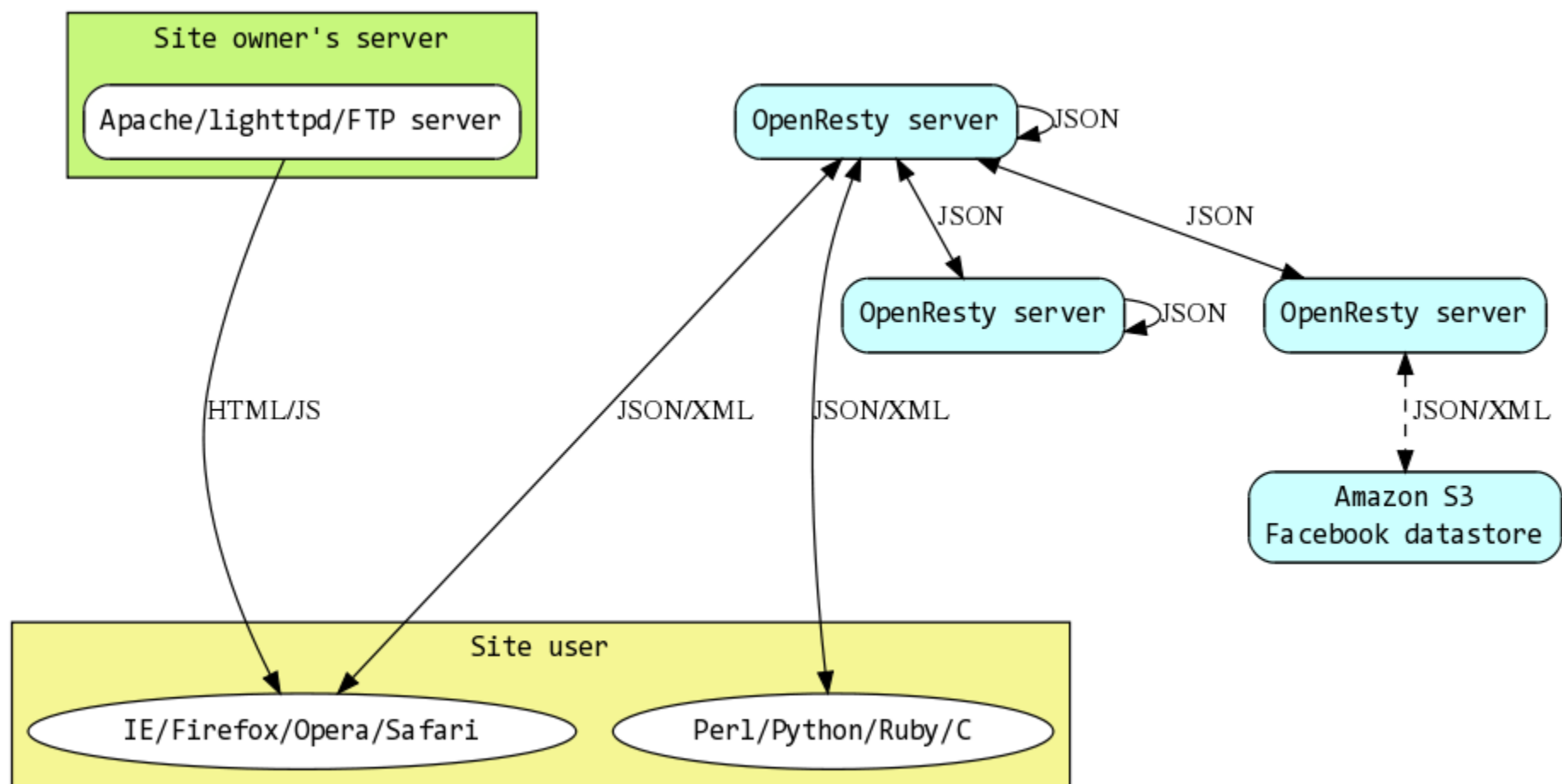
- ☺ **No** database required on the site owner's server
- ☺ Database has the full possibility to *scale*

Drawbacks

- ☹ **Two** HTTP round-trips are required
- ☹ Site owner's server is still *fat*.

The *history* of the web...

- ✓ **Web 1.0**
- ✓ **Web 2.0**
- ✓ **Post-mordern Web 2.0**
- ✓ **Web X**



Our Web X vision

Benefits

- ☺ **Clients** access third-party service sites *directly*.
- ☺ **Service** is *general, composable*,
and even *recursive*.
- ☺ All the Web 2.0 *goodies*.
- ☺ Site owner's server is *extremely thin*.
- ☺ *Deployment* of web sites is **trivial**.

Drawbacks (which may be *benefits* at the same time)

- ☹ Web *1.0* spiders **cannot** find anything interesting
at all in the HTML.
- ☹ Web sites are too easy to *steal* and **re-distributed**
by others

☺ We have *already* produced such web sites
belonging to the **Web X** era!

- ☆ They're *powered* by the **OpenResty** platform.
- ☆ They're written *completely* in **JavaScript**.
- ☆ They consist of *static* files **only**.

☀ My *blog* site

<http://blog.agentzh.org>

Human & Machine - Mozilla Firefox 3 Beta 5

File Edit View History Bookmarks Tools Help



http://blog.agentzh.org/#post-73

Google

Human & Machine

The mad house for a Perl guy, agentzh

* April 2008

<<	<					>	>>
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
		1	2	3	4	5	
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
20	21	22	23	24	25	26	
27	28	29	30				

* Recent Posts

» [A \(potential\) marriage of OpenResty and Prophet](#)

» [OpenResty versus Google App Engine](#)

» [Overview document for OpenResty](#)

» [The upcoming OpenResty talk on D2 Forum](#)

» [OpenAPI is now officially renamed to OpenResty and is now on CPAN](#)

» [SearchAll 0.4.15 is coming soon](#)

» [Overview document for OpenResty](#) | [Main](#) | [OpenAPI is now officially renamed to OpenResty and is now on CPAN](#) »

2008-04-04 00:00:00+08

The upcoming OpenResty talk on D2 Forum

I'm going to talk about OpenResty and pure JavaScript web applications in the "D2 Forum" conference which will be held on 27 of this month:

<http://d2forum.cn>

(Two Yahoo US colleagues will give presentations there as well :))

I'd like to see the RSS feed and custom Action support landed onto OpenResty before this event. Let's wait and see :)

Posted by 董亦春 at 2008-04-04 00:00:00+08 in [Articles](#) | [Permalink](#) | [Comments \(0\)](#)

* Comments

* Post a comment

Name:

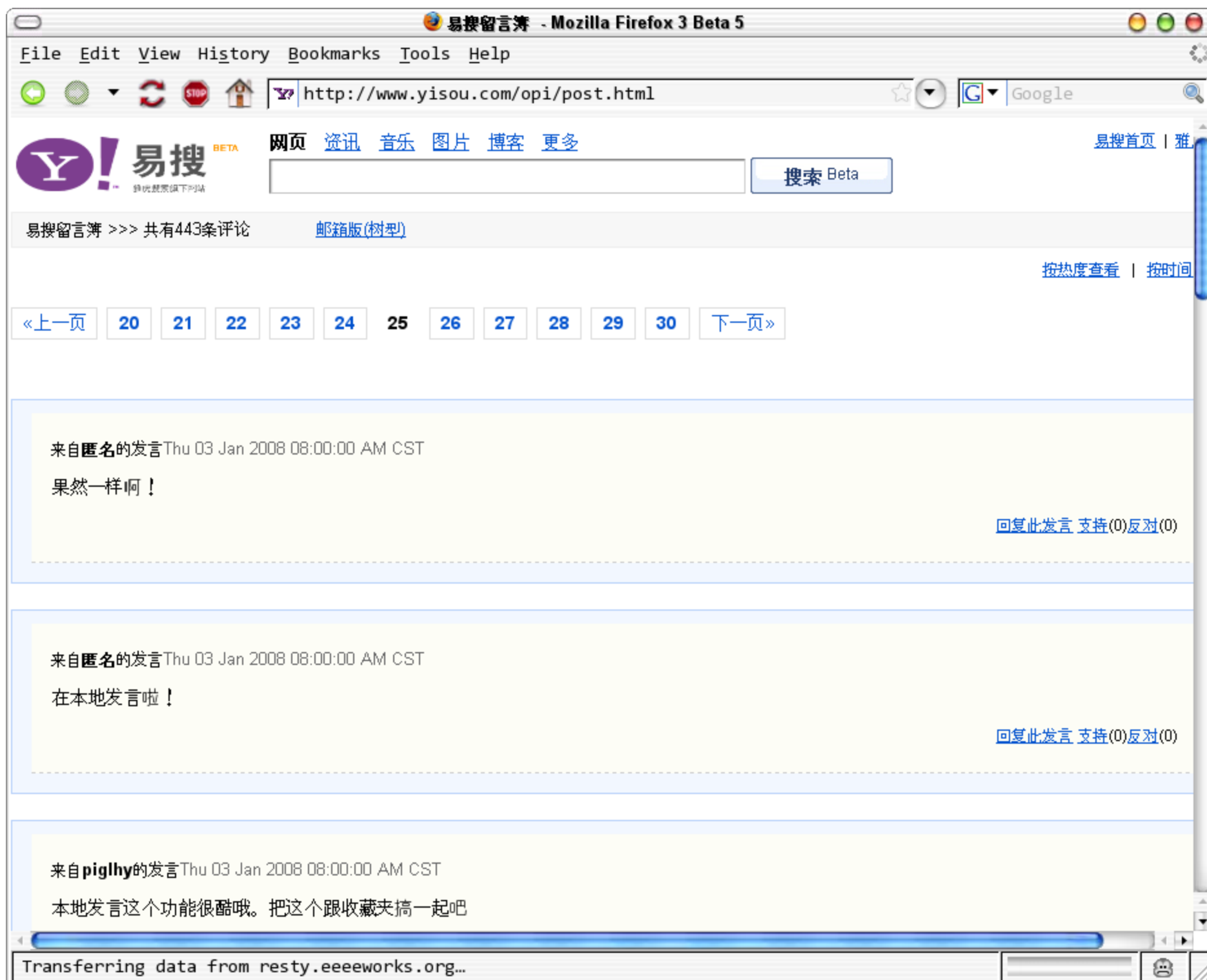
Email Address: (Not displayed with comment.)

URL:

Done

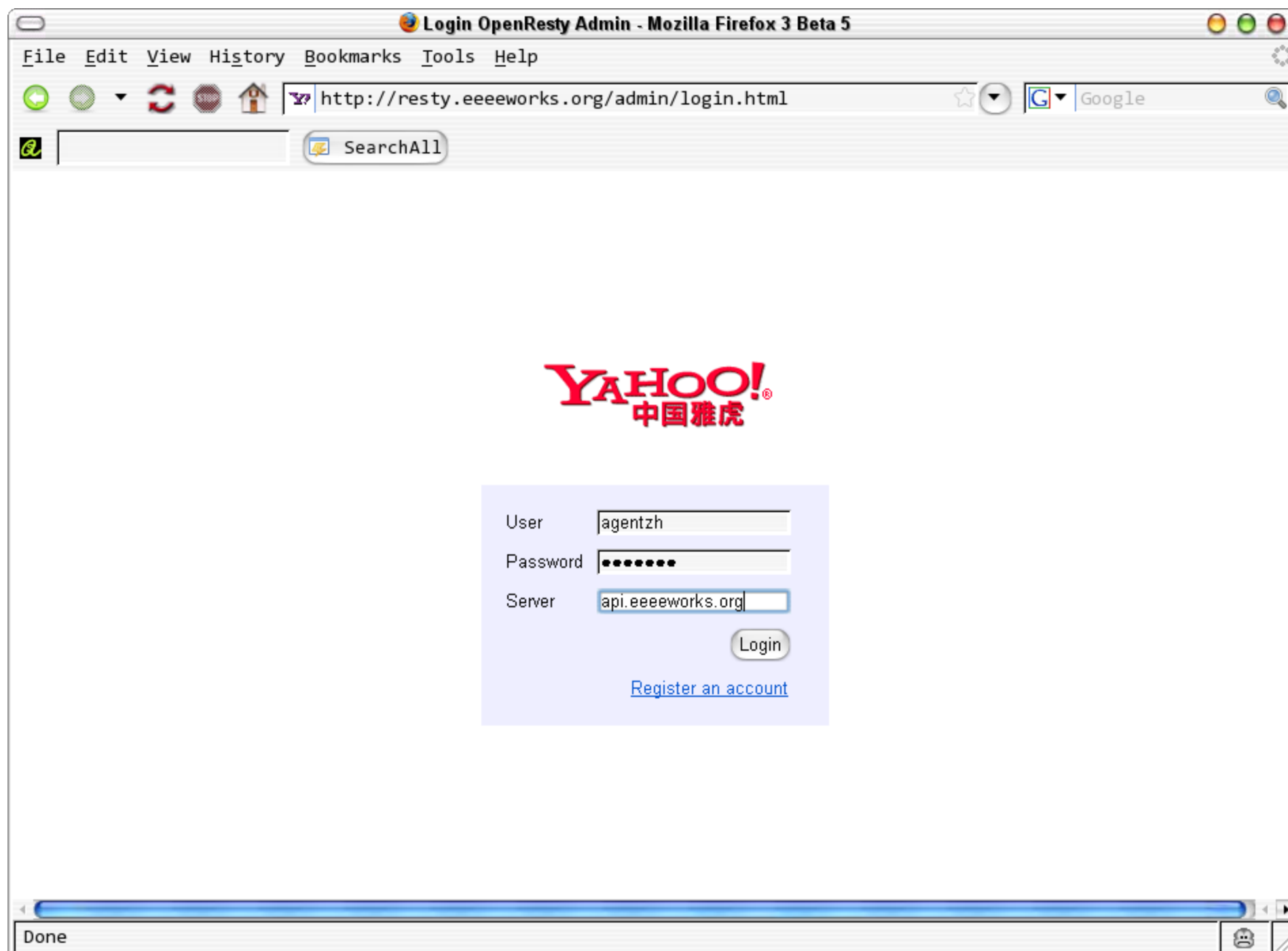
☀ Yisou *BBS*

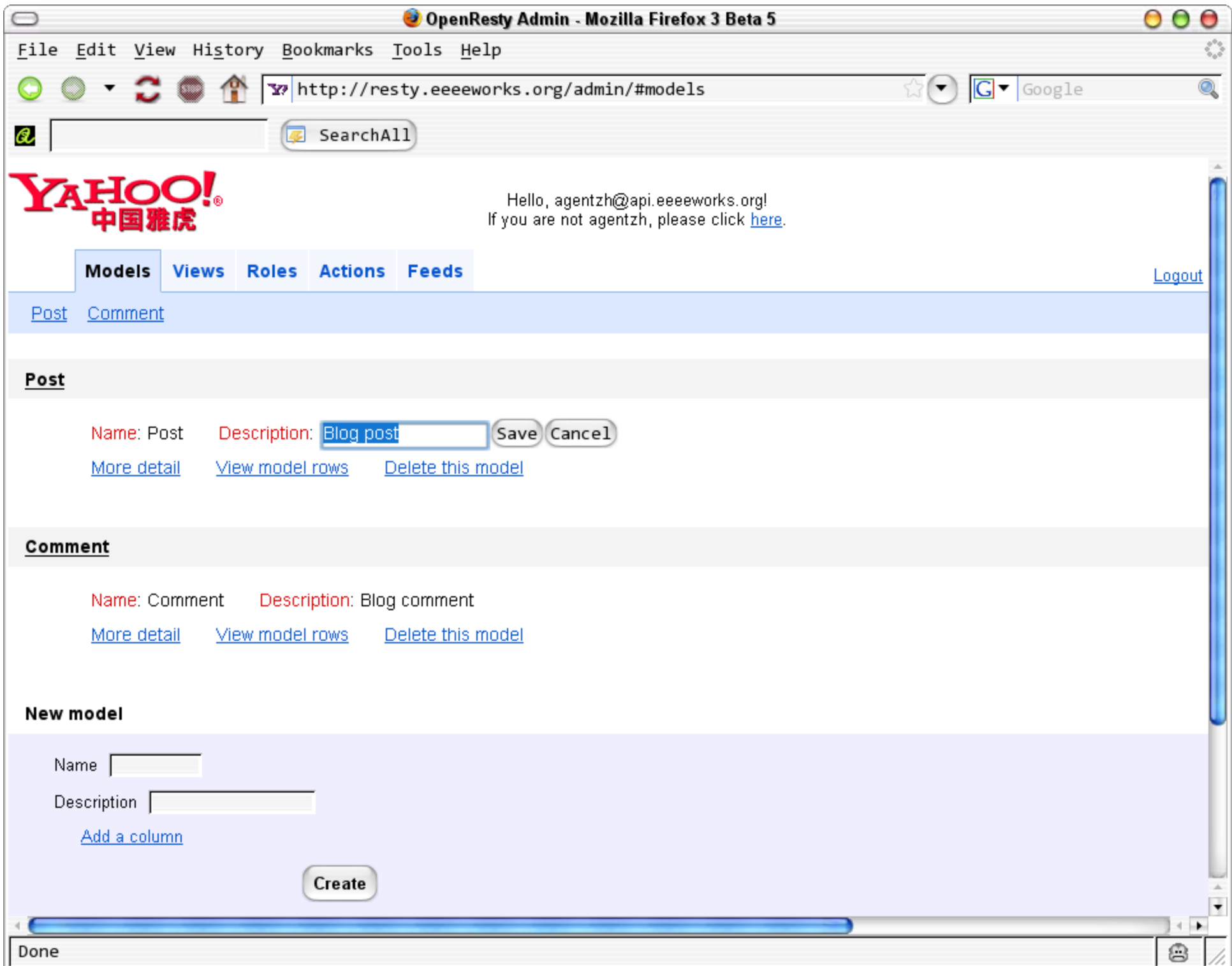
<http://www.yisou.com/opi/post.html>



☀ OpenResty *Admin* site

<http://resty.eeeeworks.org/admin/>





My blog site = index.html

+ openresty.js

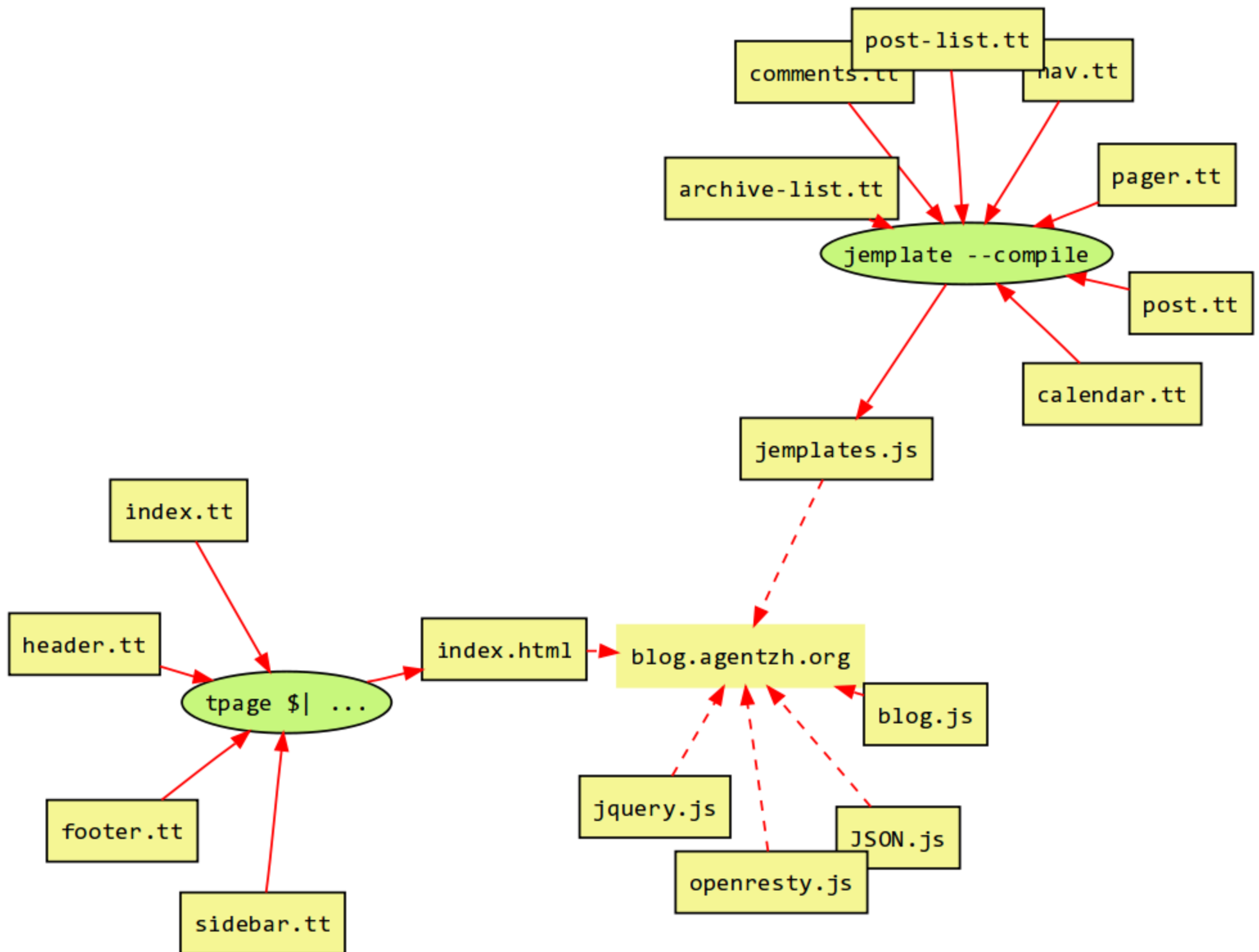
+ blog.js

+ jemplates.js

+ jquery.js

+ JSON.js

+ CSS/image



Hey, my *readers* can write a **new** blog
site for my articles atop
the **OpenResty** API
exposed by my "*agentzh*" account
without my permission!

<!-- index.html -->

<html>

<head>

<meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />

<script src="JSON.js"></script>

<script src="openresty.js"></script>

<script src="blog.js"></script>

<title>My blog</title>

</head>

<body onload="init()">

<div id="main"></div>

</body>

</html>

// blog.js

var openresty = null;

function init () {

openresty = new OpenResty.Client(

{

server: 'http://api.eeeeworks.org',

user: 'agentzh.Public'

}

);

openresty.callback = renderPosts;

openresty.get(

'/=model/Post/~/~',

{ offset: 0, count: 10, order_by: 'id:desc' }

);

}

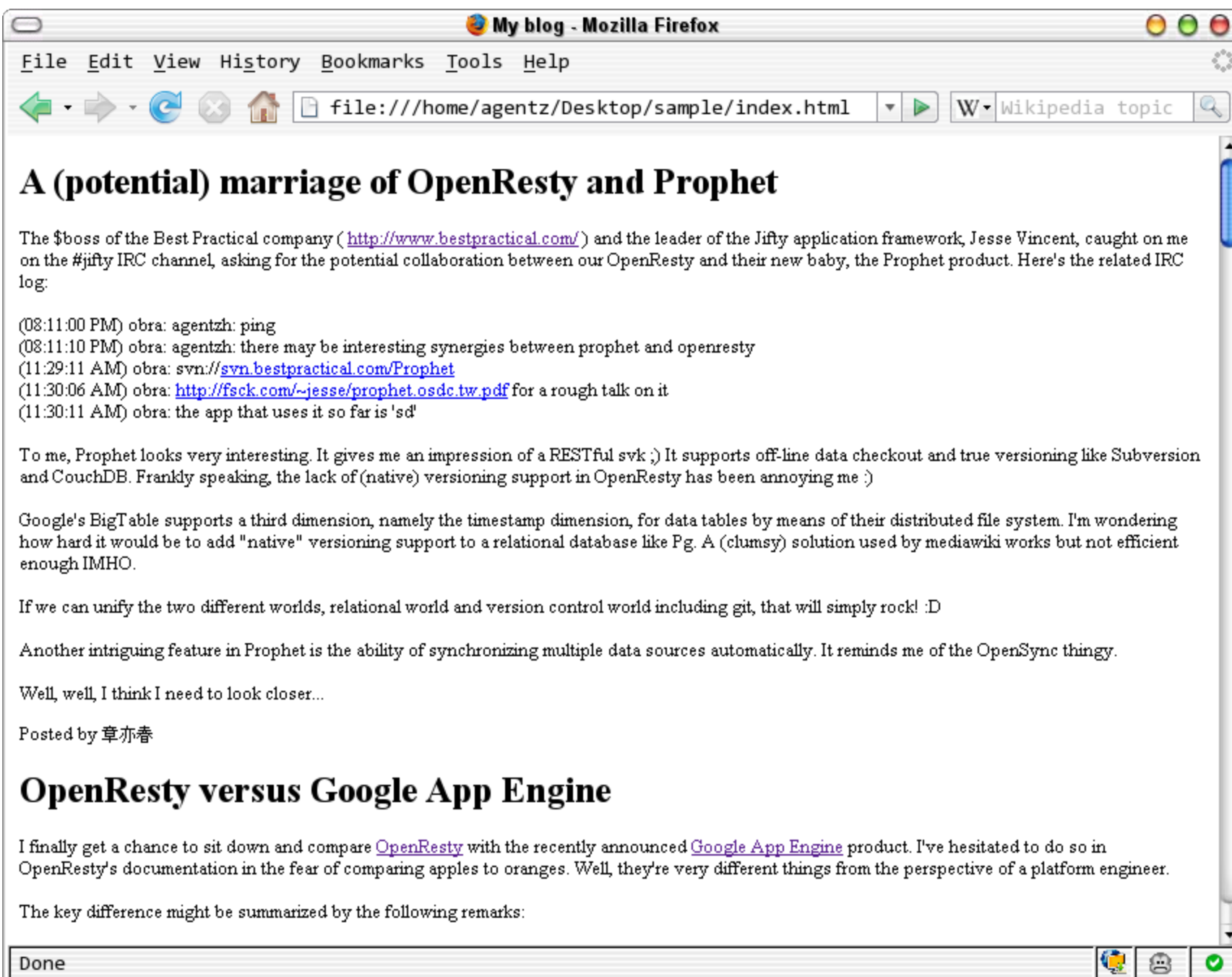
// blog.js (continued)

```
function renderPosts (res) {  
  if (openresty.isSuccess(res)) {  
    var html = '';  
    for (var i = 0; i < res.length; i++) {  
      var post = res[i];  
      html += "<h1>" + post.title + "</h1>" +  
        post.content +  
        "<p>Posted by " + post.author + "</p>";  
    }  
    document.getElementById("main").innerHTML = html;  
  } else {  
    alert("Failed to fetch posts: " + res.error);  
  }  
}
```

♥ Hey, it *runs* now!

♡ Hey, it *runs* now!

```
$ firefox ~/Desktop/sample/index.html
```



Concatenating HTML strings is boring and no fun :(

Concatenating HTML strings is boring and no fun :(

→ **Some Jemplate love**

```
<!-- post-list.tt -->
```

```
[% FOREACH post = posts %]
```

```
  <h1>[% post.title %]</h1>
```

```
  [% post.content %]
```

```
  <p>Posted by [% post.author %]</p>
```

```
[% END %]
```

```
$ jemplate --runtime > jemplates.js
```

```
$ jemplate --compile post-list.tt >> jemplates.js
```

<!-- index.html -->

<html>

<head>

**<meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />**

<script src="JSON.js"></script>

<script src="openresty.js"></script>

<script src="blog.js"></script>

<script src="jemplates.js"></script>

<title>My blog</title>

</head>

<body onload="init()">

<div id="main"></div>

</body>

</html>

// blog.js (continued)

```
function renderPosts (res) {  
  if (openresty.isSuccess(res)) {  
    var html = Jemplate.process(  
      'post-list.tt', { posts: res }  
    );  
    document.getElementById("main").innerHTML = html;  
  } else {  
    alert("Failed to fetch posts: " + res.error);  
  }  
}
```

♥ Hey, it *runs* again!

♡ Hey, it *runs* again!

\$ firefox ~/Desktop/sample/index.html

So **what** is *OpenResty* then?

OpenResty is a

- ✓ *general-purpose* **RESTful** web service platform
- ✓ REST **wrapper** for *relational* databases
- ✓ Virtual web *runtime* for **RIA**
- ✓ "*meta* web site" supporting **other sites**
via web services
- ✓ handy *web database* which can be accessed
from **anywhere**

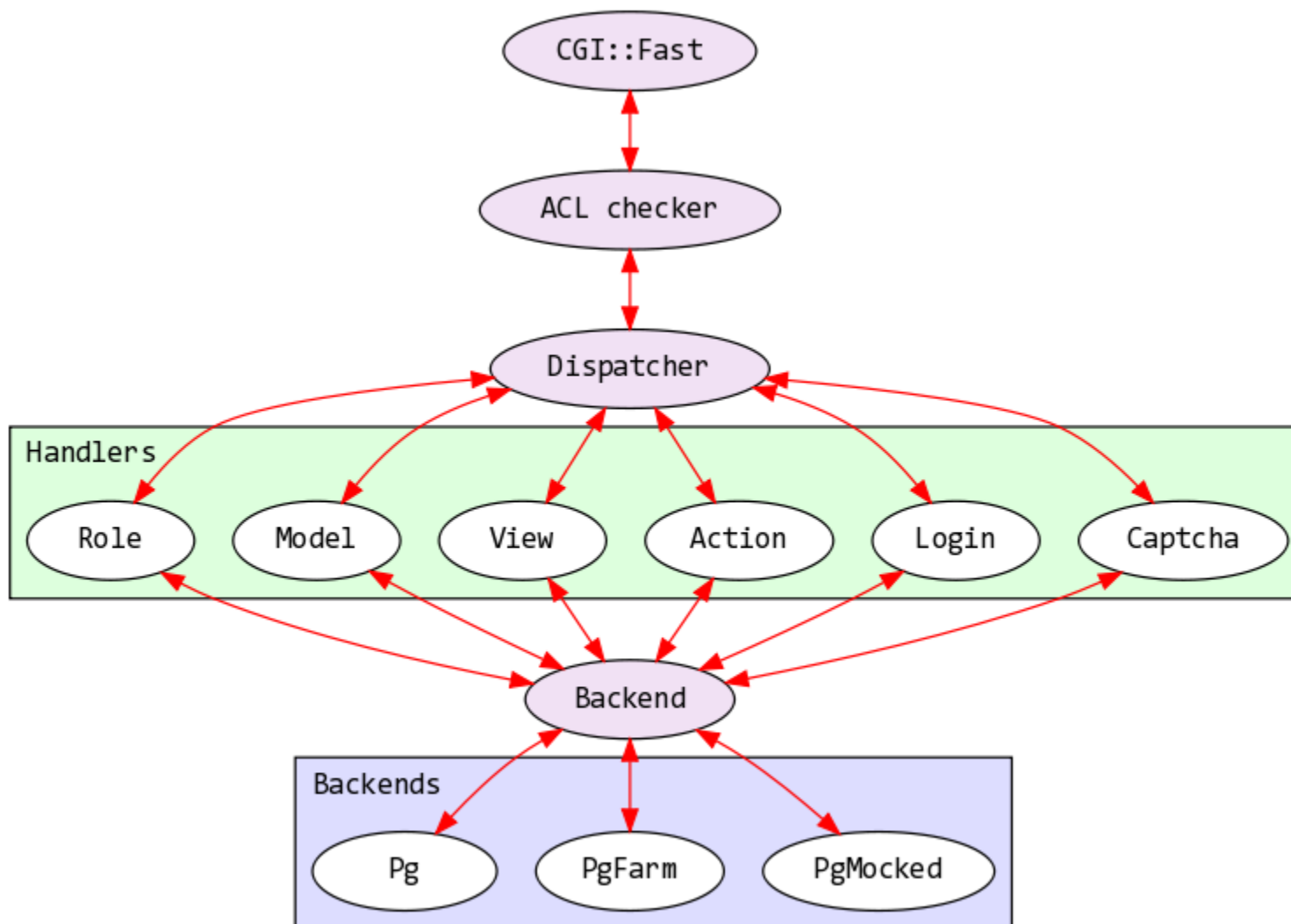
OpenResty is **NOT** a

- × ***server***-side web application **framework**.
- × replacement for highly scalable
semi-structured data storage solutions
like Amazon SimpleDB or CouchDB.

OpenResty offers

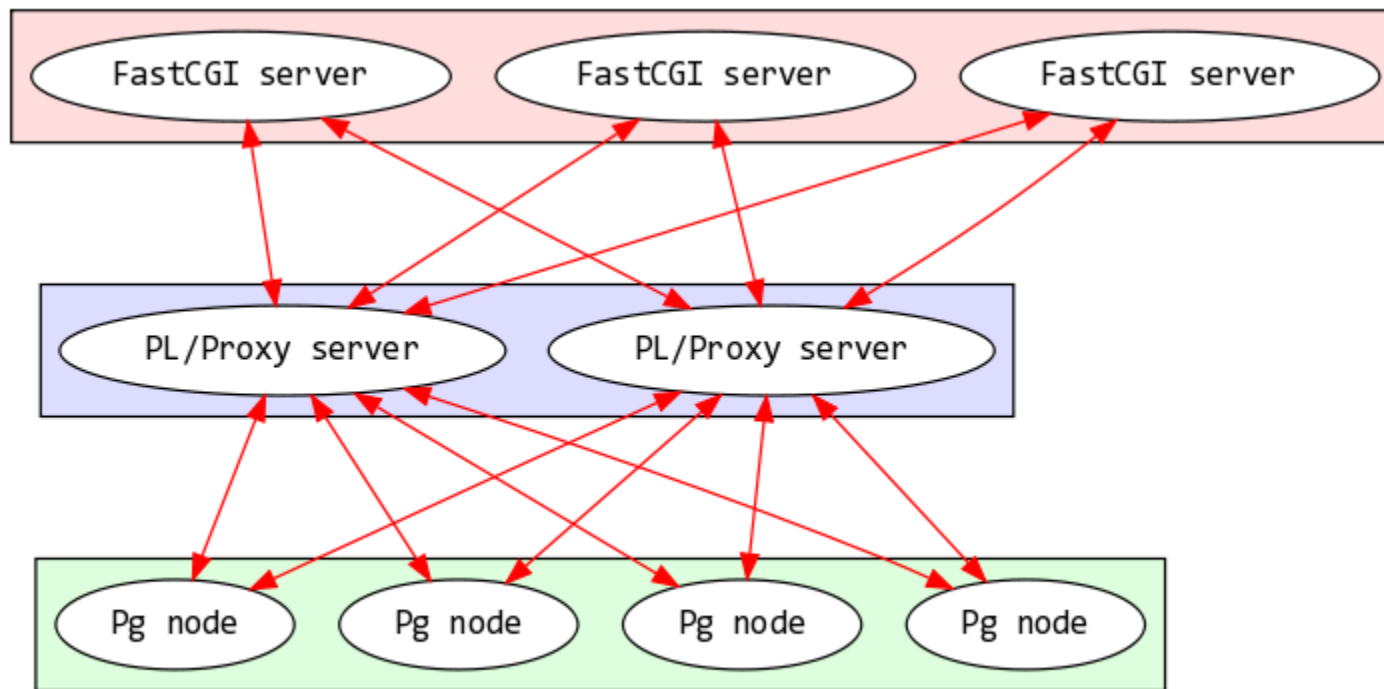
- ✓ (scalable) *relational* data **storage**
- ✓ *truely* **RESty** interface
- ✓ *JSON/YAML* data transfer format
- ✓ **SQL**-based reusable *views*
- ✓ a REST-oriented *role system* for **ACL**
- ✓ *view*-based **RSS feeds**
- ✓ user-defined *actions* in **RestyScript**
- ✓ native *captchas*
- ✓ *cross-site* **AJAX** support

♥ The OpenResty *FastCGI* server
is currently written in Perl 5.



The Internal Structure of the OpenResty Server

♥ The **PgFarm** backend of OpenResty
is designed to be *scalable*.



The Architecture Diagram for OpenResty PgFarm

"How can I get *started*?"

- ➔ Write to *agentzh@gmail.com* to request an OpenResty **account**
- ➔ OpenResty is on *CPAN* already!
<http://search.cpan.org/dist/OpenResty>

"Where can I learn *more* about OpenResty?"

→ See the **OpenResty::Spec::Overview** document

<http://search.cpan.org/perldoc?OpenResty::Spec::Overview>

"*How* can I get *involved*?"

➔ Write to *agentzh@gmail.com* to ask a commit bit!
<http://svn.openfoundry.org/openapi/trunk>

😊 *Any questions?* 😊

