

The Erlang Web

an Open Source Fault Tolerant and
Scalable Web Framework

Erlang Factory, Palo Alto
May 1, 2009



Erlang Training and Consulting Ltd
www.erlang-consulting.com

Michał Ptaszek, Michał Slaski
michal@erlang-consulting.com

Contents

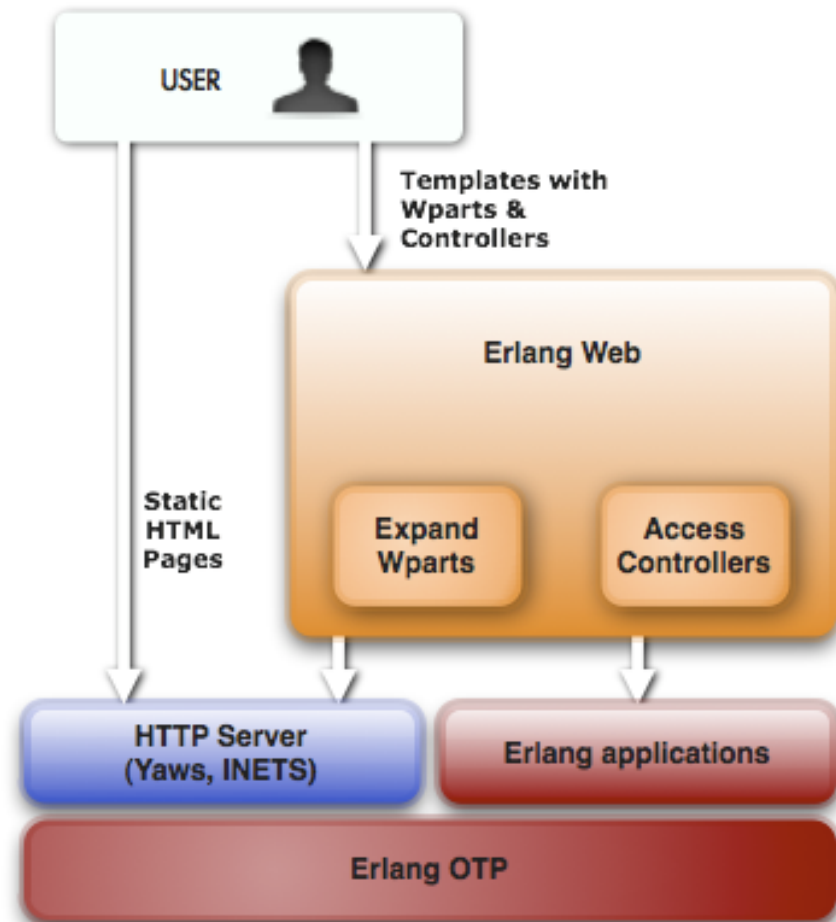
- The reason to have the Erlang Web and how it works
- Erlang Web 1.3 features
 - *Function annotations*
 - *Form builder*
 - *Cache*
 - *Support for two-tier architectures*
 - *Plugins*
- License

The reason to have the Erlang Web

- Concurrent Erlang good for web applications
- HTTP servers: Yaws and INETS
- Databases: Mnesia and CouchDB
- Same platform and share same memory space
- MVC paradigm
- No Erlang code in HTML templates
- Reusable components (plugins)
- Cache

How it works

- Built on top of HTTP server
- Calls controller functions
- Expands templates with dynamic parts



Features of Erlang Web 1.2

- Dispatcher
 - *regular expressions used to match on URL*
- Controllers
 - *functions accessed through dispatcher*

```
{dynamic, "^/index.html$", {main, home}}.
```

```
{dynamic, delegate, "^/user", "config/dispatcher/user.conf"}.
```

```
{static, "^/about$", "about.html"}.
```

```
{static, "^style.css$", enoent}.
```

Features of Erlang Web 1.2

- Templates
 - *XHTML files and wtpl engine with inheritance*
- Wparts
 - *tags put in templates*
- Wtypes
 - *data types for which validation and formatting can be done automatically*

```
<wpart:list select="map" as="user_age" list="all_users">  
  <p><wpart:lookup key="user_age" format="integer"/></p>  
</wpart:list>
```

Features of Erlang Web 1.2

- Request dictionary
 - *mechanism for accessing data in controllers and wparts*

```
wpart:fget("post", "form_input"),  
wpart:fget("session", "username"),
```

```
wpart:fset("node", node())
```

And then in your template: `<wpart:lookup key="node" format="atom"/>`

- i18n support
 - *files with multilingual translations*

in config/languages/pl.conf file: `{"pass", "Hasło"}`.

And then in your template: `<wpart:lang key="pass"/>`

Features of Erlang Web 1.3

- Annotations
 - *Define functions to be called before and after the controller*
- Form builder
 - *HTML form generated basing on Erlang record definition*
- Caching
 - *Stores pre-rendered parts of template or entire pages*
- Distribution
 - *Two-tier architectures with front-end and back-end nodes*
- Components
 - *Reusable plugins with support for Erlang Web*

Annotations

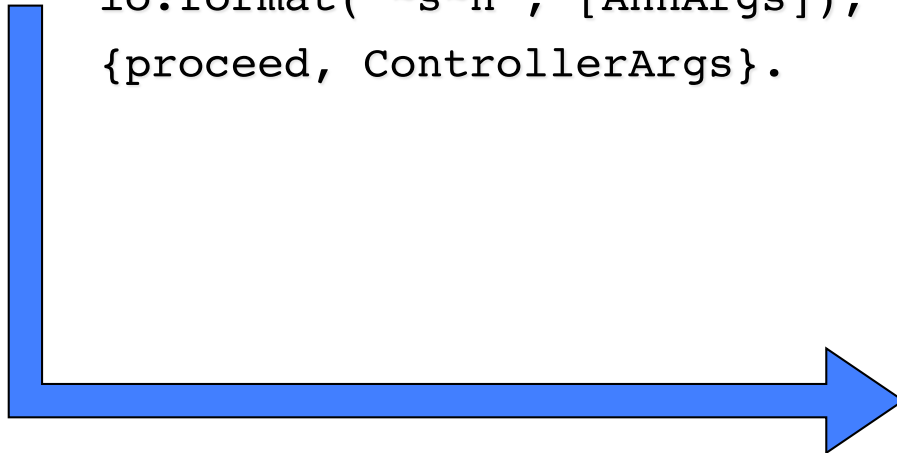
- Meta-language for controlling call flow
- Define functions to be called before executing the controller
 - *Authenticate user*
 - *Validate input data*
- And after executing the controller
 - *Invalidate cached entries*
 - *Log the activity*

Annotations

```
-module(my_utils).  
-export([logme/4]).  
-include_lib("eptic/include/e_annotation.hrl").
```

?BEFORE.

```
logme(AnnArgs, Mod, Fun, ControllerArgs) ->  
    io:format("~s~n", [AnnArgs]),  
    {proceed, ControllerArgs}.
```



```
-module(controller).  
-export([my_fun/0]).  
[...]
```

```
?LOGME("Calling my_fun").  
my_fun() ->  
    wpart:fget("post", "id"),  
    [...]
```

Form builder

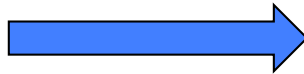
- Generate HTML form to edit Erlang records
- The form can be rendered using
 - *HTML tables* - `<table><tr><th/><td/>`
 - *Paragraphs* - `<p id="..."><label>`
 - *Lists* - `<label>`
 - *Divs* - `<div id="..."><label>`

Form builder

- Form fields generated basing on Erlang records

article.hrl

```
-record(article, {  
    id,  
    text,  
    hide  
}).
```



templates/article.html

```
<wpart:form type="article"  
    action="/article/create" />
```



```
-record(article_types, {  
    id = {integer, [...]},  
    title = {string, [...]},  
    hide = {enum, [...]}  
}).
```

Article text

Is visible?

☒ Yes
☐ No

Submit Query

Caching

- **Caching result of calling a controller**

```
{dynamic, "^/blog/list", {blog, list}, [{cache, persistent}]}.
```

- **Caching entire template**

```
{static, "^/faq$", "doc/faq.html", [{cache, normal}]}.
```

- **Caching part of a template**

```
<wpart:cache id="left_menu" groups="menus" type="persistent">  
    SOME CONTENT  
</wpart:cache>
```

Invalidating cache

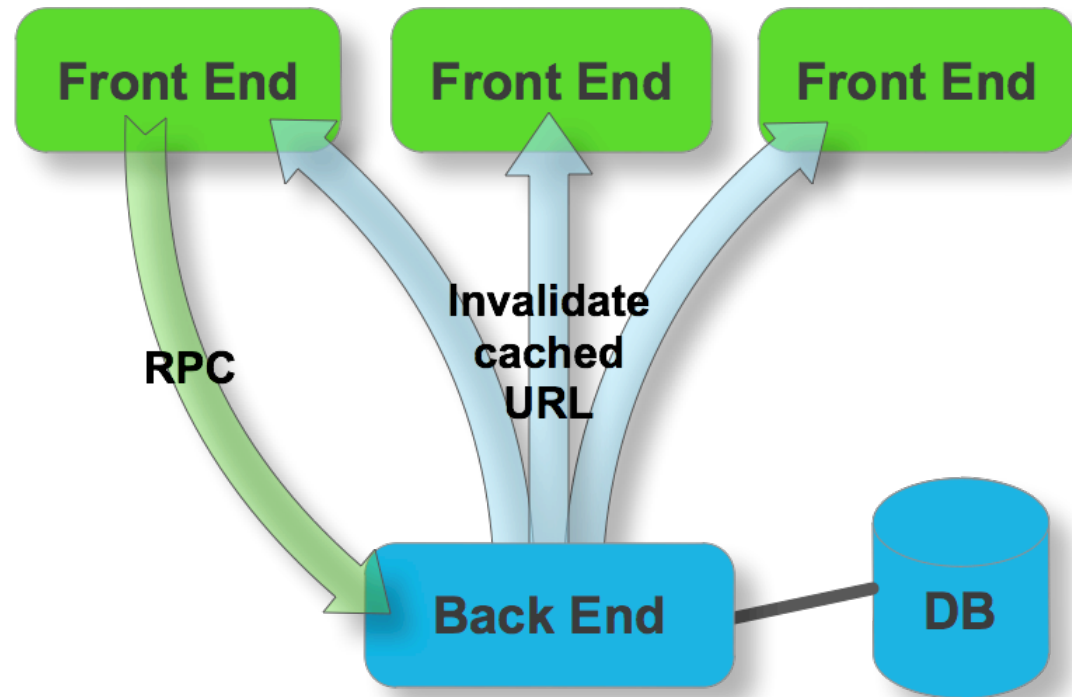
- Entries can be removed from cache using regular expressions
- There are some annotations to invalidate cache

```
?INVALIDATE([ "^/blog/list" ] ).  
update_blog() ->  
    [update your database here...]  
  
    {template, "blog.html"}.
```

```
?INVALIDATE_GROUPS([ "menus" ] ).  
update_menu() ->  
    [update your database here...]  
  
    {template, "blog.html"}.
```

Distribution

- Front End
 - *HTTP server*
 - *Dispatcher*
 - *Cache tables*
 - *Static content*
- Back End
 - *Invalidator*
 - *Controllers*
 - *Templates*
 - *Database*

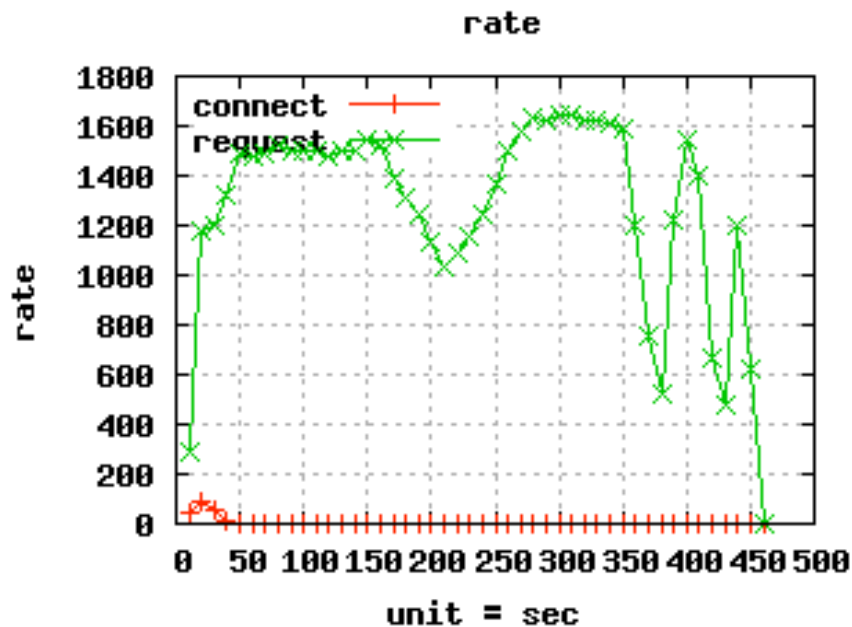


Stress testing

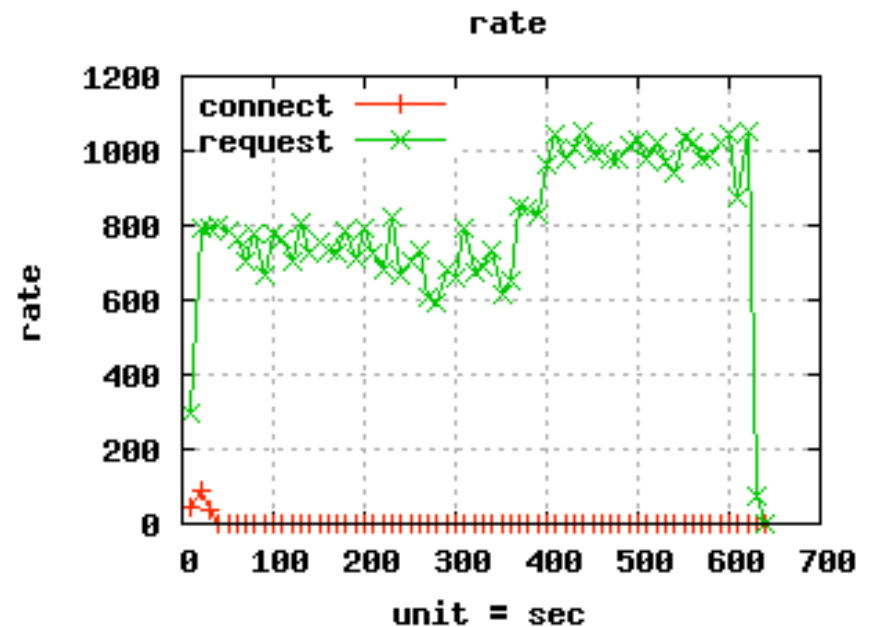
- One machine for Erlang Web application and one for Tsung
 - *Intel Core 2 Due 2.4 GHz*
 - *3 GB DDR2 RAM*
 - *Erlang R12-5, no hipe*
 - *Suse 11*
- Application implemented with
 - *INETS 5.0.12 or Yaws 1.80*
 - *Mnesia as database*
 - *dispatcher, template inheritance*
 - *Erlang Web cache or memcached (through merle)*

Stress testing

- Erlang Web cache
- Inets server
- Tsung (HTTP 1.1 connections)
 - 1500 pages / sec



- Memcached through merle
- Inets server
- Tsung (HTTP 1.1 connections)
 - 1000 pages / sec



Plugins

- `bin/e_component.erl list` - lists all the e_components
- `bin/e_component.erl search Keyword`
- `bin/e_component.erl details Name`
- `bin/e_component.erl install Name`
- `bin/e_component.erl path Path install Name`
- Available at <http://ecomponents.erlang-web.org>

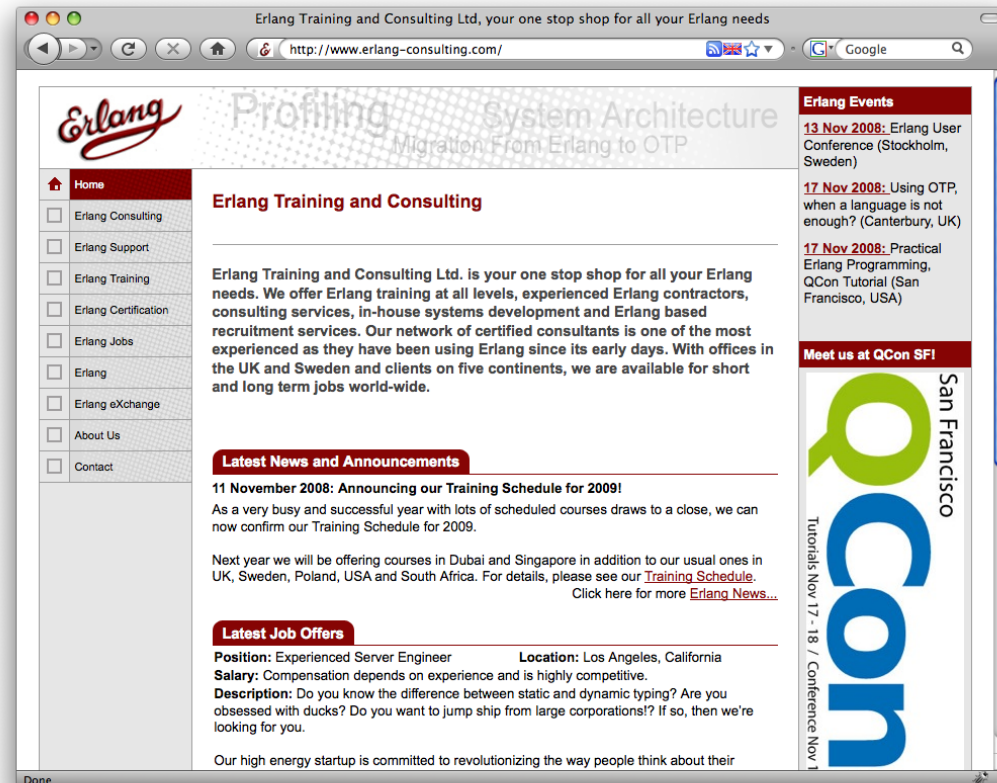


License

- Erlang Web Public License
 - *Derivative work of the Erlang Public License, Version 1.1*
 - *Changes definition of Modifications to include also wpart/wtype modules*

Example sites

- erlang-consulting.com



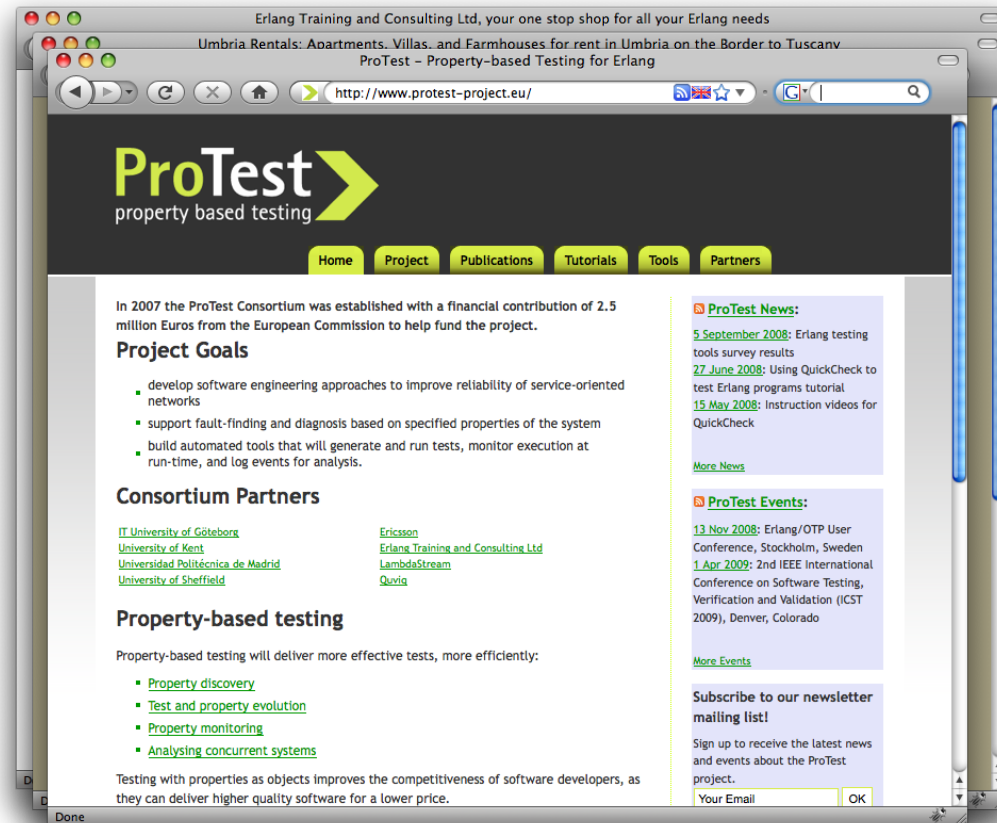
Example sites

- erlang-consulting.com
- umbria-rentals.com



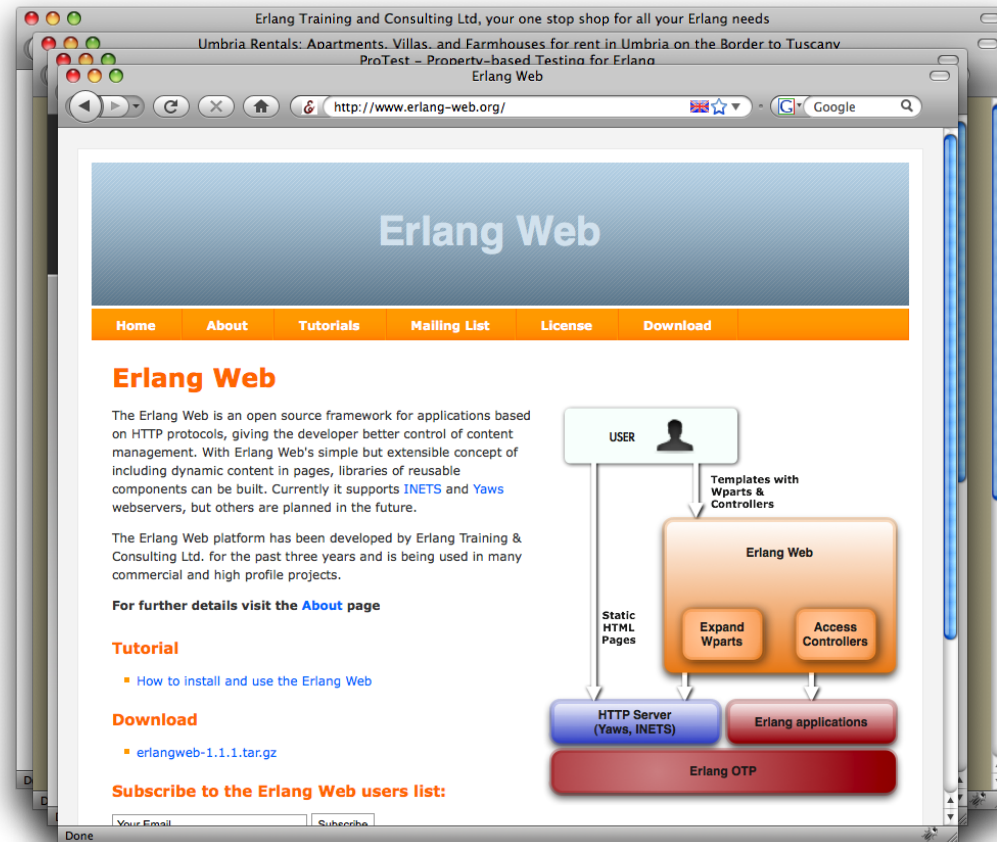
Example sites

- erlang-consulting.com
- umbria-rentals.com
- protest-project.eu



Example sites

- **erlang-consulting.com**
- **umbria-rentals.com**
- **protest-project.eu**
- **erlang-web.org**



Example sites

- erlang-consulting.com
- umbria-rentals.com
- protest-project.eu
- erlang-web.org
- erlang-factory.com



Resources

- Project page:
<http://www.erlang-web.org>
- Wiki:
<http://wiki.erlang-web.org>
- Sources (mercurial repository)
<http://bitbucket.org/etc/erlang-web>

Questions



Thank You!

For more information and to download visit
<http://www.erlang-web.org>