

# ERb alternatyvos

RubyConfLT 2010  
Vidmantas Kabošis  
[vidmantas.kabosis.lt](http://vidmantas.kabosis.lt)



# Kas yra kas (intro)

- Šablono variklis == template engine
- Greitis
- Saugumas



# ERb

- Embedded Ruby
  - `<%= ... %>`
- ERB - kartu su standartiniu Ruby
- Autsaideris greičio ir saugumo atžvilgiu
- Turi realizacijų, sprendžiančių jo problemas
  - eruby
  - erubis
  - Ember
  - rbTenjin



# eruby

- C



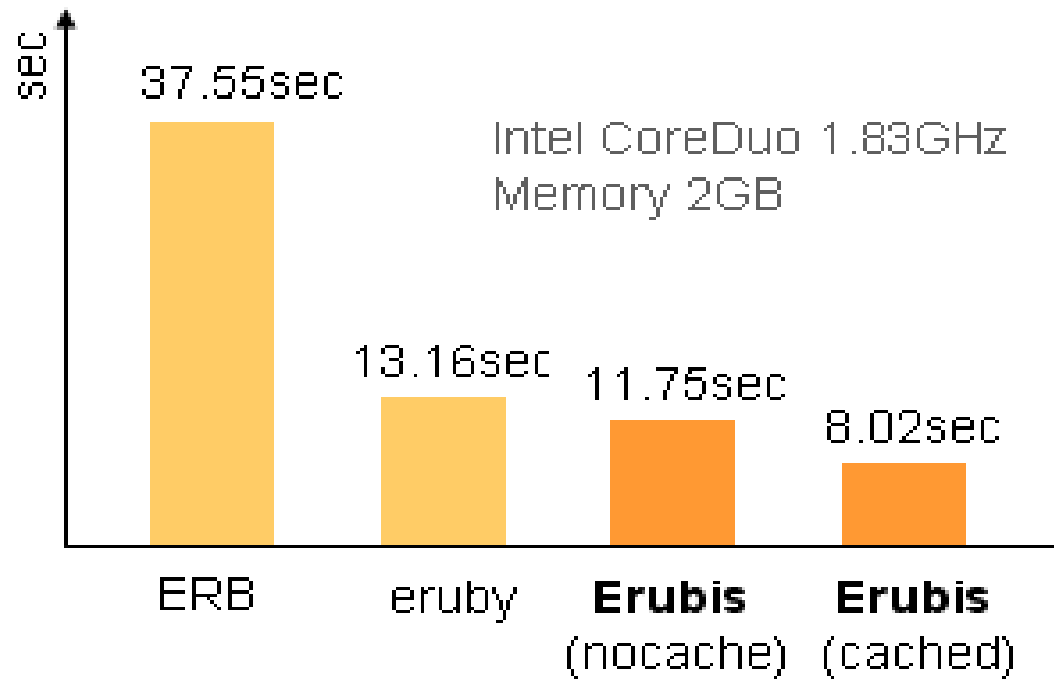
# erubis

- Ruby
- Greitesnis už ERB, eruby
- Keičiamas įterpimo šablonas
- Automatinis *eskeipinimas*
- Kešavimas (preprocessing)
  - [%= link\_to 'Create', :action=>'create'%]
- Priima vykdymo nurodymus (processing instructions) šablonuose
- Rails 3



# erubis

**Benchmark: elapsed time to create 10,000 pages**



<http://www.kuwata-lab.com/erubis/>

# ember

- Ruby
- Gali automatiškai papildyti trūkstamais `<% end %>` pagal identaciją
- Klaidų pranešimuose naudoja teisingus eilučių numerius
- Papildomos instrukcijos
  - `<%+ "doc/example.erb" %>`
- Ruby on Rails



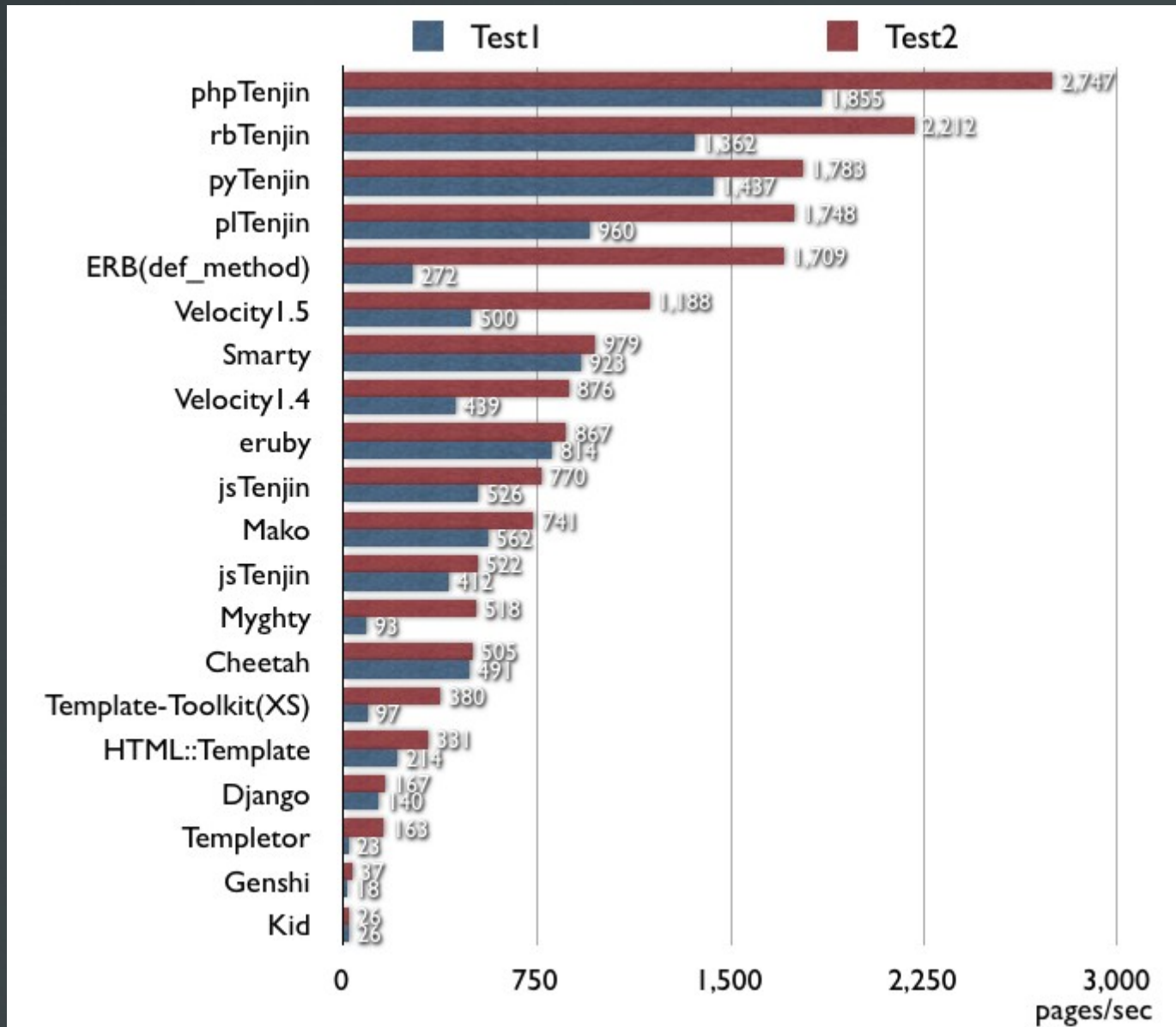
# rbTenjin (1/3)

- Lengvas, greitas (greičiausias?)
- Saugus „pagal nutylėjimą”
- Layouts, partials
- Konvertuoto kodo kešavimas
- ...
- Savybėmis labai panašus į erubis





# rbTenjin (2/3)



# rbTenjin (3/3)

```
##### layout
```

```
<html>
  <body>
    <h1>#{@title}</h1>
    <div class="main-content">
      #{@_content}
    </div>
    ...
```

```
##### template
```

```
<table>
  <?rb i = 0 ?>
  <?rb for item in @items ?>
  <?rb   i += 1 ?>
  <?rb   color = i % 2 == 0 ? '#FFCCCC' : '#CCCCFF' ?>
    <tr bgcolor="#{color}">
      <td>#{i}</td>
      <td>#{item}</td>
    </tr>
  <?rb end ?>
</table>
```



# Kitas požiūris

- TAL
- Smarty-like (Liquid)
- Ruby DSL
- wiki-style
- Kiti :-)

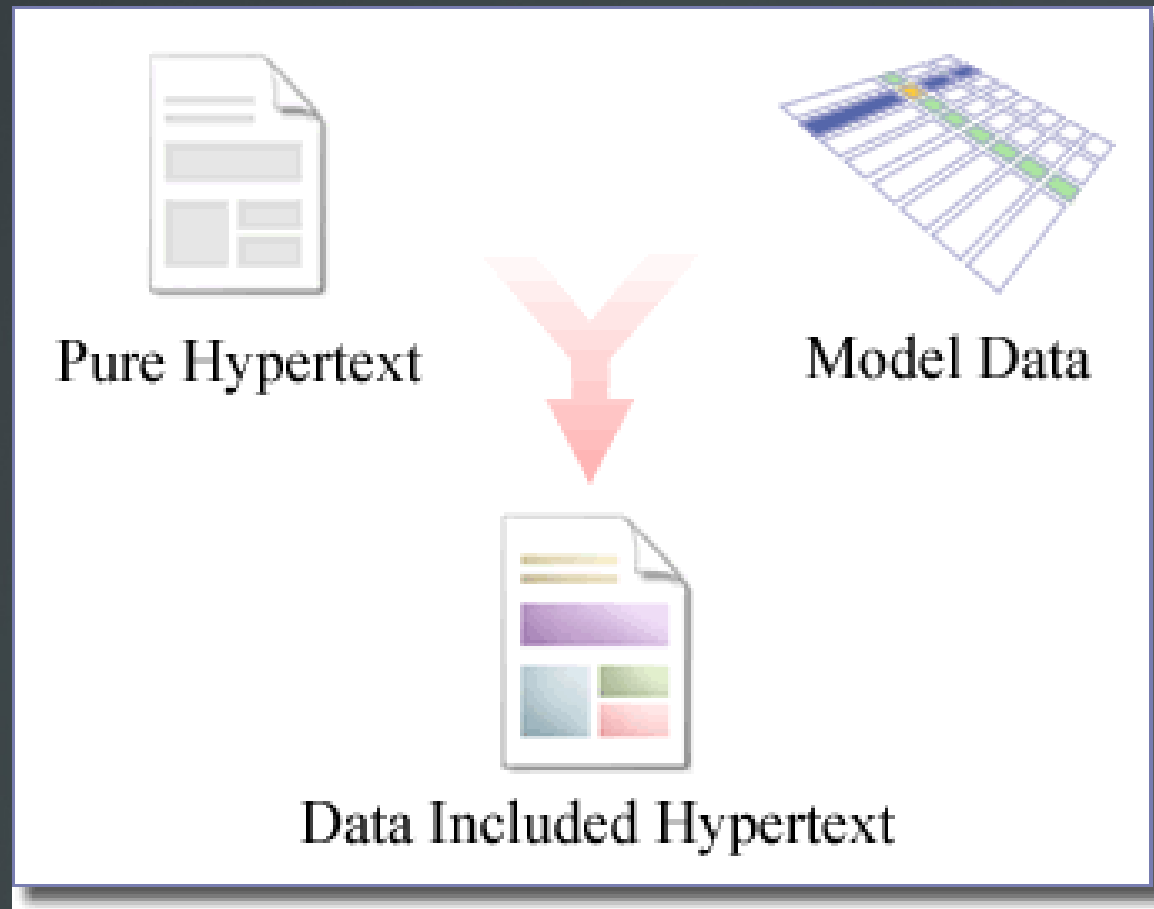


# TAL

- Template Attribute Language
- Elementai žymimi TAL aprašo atributais
- Validus XML/HTML dokumentas
- Parašytas „Zope”
- TALES, METAL
- Ruby realizacijos:
  - Amrita/Amrita2
  - Kwartz



# TAL: Amrita (1/3)



<http://amrita.sourceforge.jp/>

# TAL: Amrita (2/3)

- „Pure XML/HTML document”
  - $\langle ? \dots ? \rangle$ ,  $\langle \% = \dots \% \rangle$
- Išvestį kontroliuoja duomenys
- Gali būti sukompiliuotas į ruby
- Amrita 2



# TAL: Amrita (3/3)

```
<table border="1">
  <tr>
    <th>name</th><th>author</th>
  </tr>
  <tr id="table1">
    <td id="name">
      <td id="author">
    </tr>
</table>
```

```
# data
data = {
  :table1=>[
    { :name=>"Ruby", :author=>"matz" },
    { :name=>"perl", :author=>"Larry Wall" },
    { :name=>"python", :author=>"Guido van Rossum" },
  ]
}
```



# TAL: Kwartz

- „Independence of Presentation Logic”
- Šablonas + šablono logika (plogic)

```
#list1 {  
  logic: {  
    for member in @members  
      _stag          # start tag  
      _cont          # content  
      _etag          # end tag  
    end  
  }  
}
```

```
<tr id="list1">  
  <td id="mark:item1">foo</td>  
</tr>
```



# TAL

DHH:

*The pursuit of "no code"-templates reminds me of the search for the holy grail of the MDA camp with "no code"-programs. It's mirage, but its also a play on words of the "a rose by any other name..." variety.*

<http://www.loudthinking.com/arc/000405.html>



# Smarty-like (Liquid)

- Atskira „kalba” šablonuose
- Ruby realizacijos:
  - Liquid
  - Ruty



# Liquid (1/2)

- Iš „Shopify“, naudojamą production'e
- Ruby on Rails
- Saugus
- Sintaksė panaši į Ruby
- Plečiama (tagai, filtrai, blokai)



# Liquid (2/2)

```
Hello {{ '*rubyconf*' | textilize | upcase }}
```

```
{% if user.name == 'bob' and user.age > 45 %}
```

```
  Hello old bob
```

```
{% endif %}
```

```
# array = [1,2,3,4,5,6]
```

```
{% for item in array limit:2 offset:2 %}
```

```
  {{ item }}
```

```
{% endfor %}
```

```
# results in 3,4
```



# Ruty

- Ištakos: Django / Jinja Template Engine
- Šablonų paveldėjimas
- Sintaksė visiškai analogiška Liquid (?)
- Papildomos saugumo priemonės
- Lėtesnis už ERB, bet greitesnis už Liquid



# Ruby DSL

- Priešingai nei ERB/TAL/Liquid, aprašo šablonus su „pure Ruby”
- Pagrindinis žaidėjas - „Markaby”
  - „Maline” - miręs
  - Builder::XmlMarkup – tik XML dokumentams



# Ruby DSL: Markaby

- „Markup as Ruby”
- Naudojamas „Camping” microframework'e
  - Lengvai integruojamas į Rails
- Susidūrusiems su DSL, lengvai perprantamas

```
html do
  head do
    title 'Products: ' + action_name
    stylesheet_link_tag 'scaffold'
  end
  body do
    p flash[:notice], :style => "color: green"
    self << yield
  end
end
```





# wiki-style

- Nėra tikrosios „šablonų kalbos”
- Skirtos vartotojų įvesčiai apdoroti
- Markdown:
  - BlueCloth, Maruku, Kramdown, RDiscount
- Textile:
  - RedCloth, ?





# Kiti

- HAML
- Radius



# Kiti: HAML (1/2)

- „Markup haiku”
- Principai:
  - Markup should be DRY
  - Markup should be well-intended
  - HTML structure should be clear
  - Markup should be **meaningful**
- HAML sesutè CSS'ui: SASS



# Kiti: HAML (2/2)

- Klasė žymima `.`, id - `#`, atributai - `()`, tag'as - `%`

```
#content
```

```
  .left.column
```

```
    %h2 Welcome to rubyconf!
```

```
    %p= print_information
```

```
  .right.column
```

```
    = render :partial => "sidebar"
```



# Kiti: Radius

- Šablonų kūrimui
- Leidžia patogiai apibrėžti savo tag'us ir juos parsinti
- Šablonas – beveik XML (namespace)

```
context.define_tag "hello" do |tag|  
  "Hello #{tag.attr['name'] || 'World'}!"  
end
```

```
parser = Radius::Parser.new(context)  
parser.parse('<p><radius:hello name="RubyConf" /></p>')
```

# Pabaigai

- Svarbiausia – produktyvumas
- Naudojami:
  - ERB plačiaja prasme
  - HAML
  - Liquid



end

?

