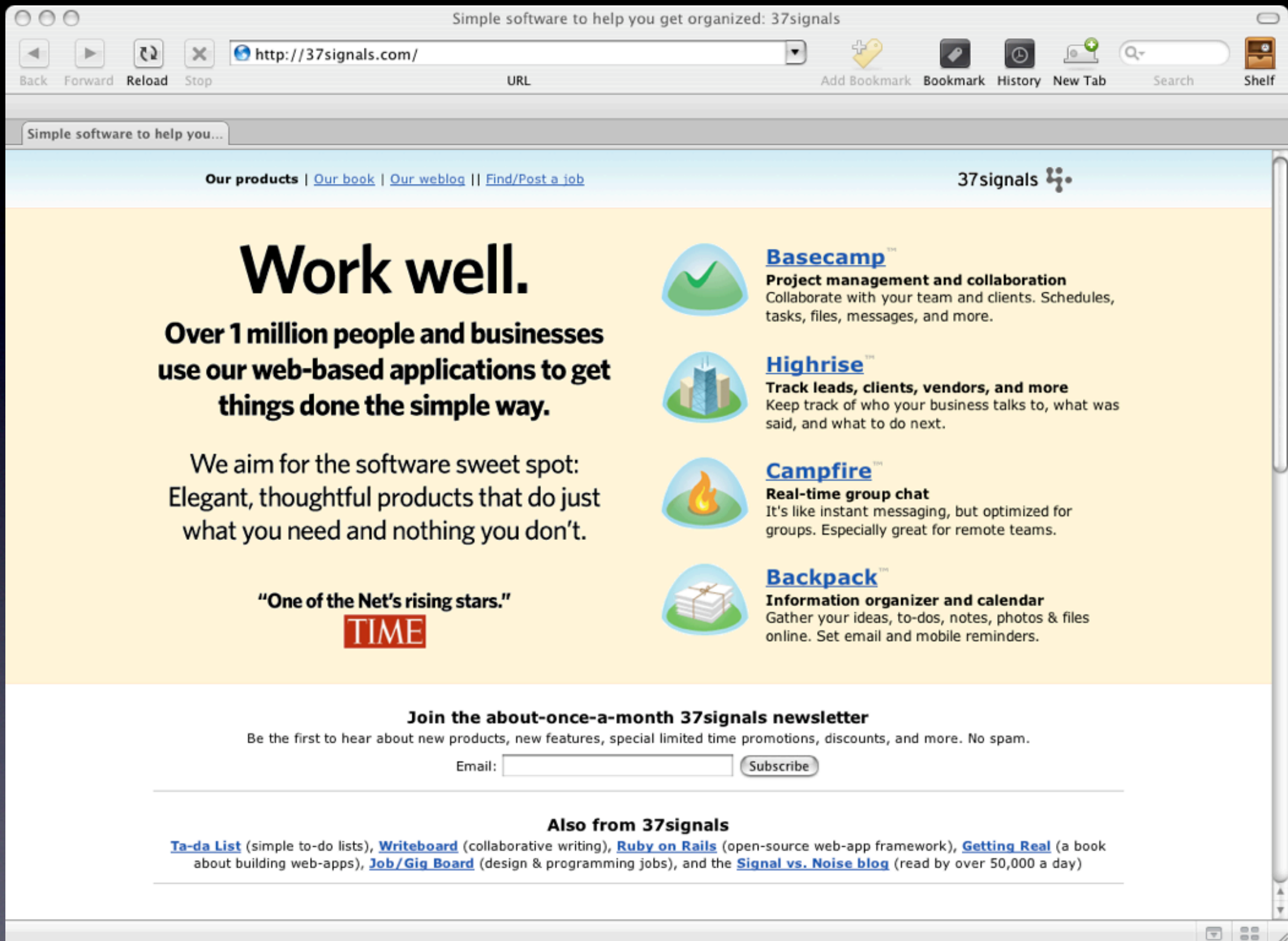




Ruby On Rails

Framework Agile



Work well.

Over 1 million people and businesses use our web-based applications to get things done the simple way.

We aim for the software sweet spot: Elegant, thoughtful products that do just what you need and nothing you don't.

"One of the Net's rising stars."



Basecamp™

Project management and collaboration
Collaborate with your team and clients. Schedules, tasks, files, messages, and more.



Highrise™

Track leads, clients, vendors, and more
Keep track of who your business talks to, what was said, and what to do next.



Campfire™

Real-time group chat
It's like instant messaging, but optimized for groups. Especially great for remote teams.



Backpack™

Information organizer and calendar
Gather your ideas, to-dos, notes, photos & files online. Set email and mobile reminders.

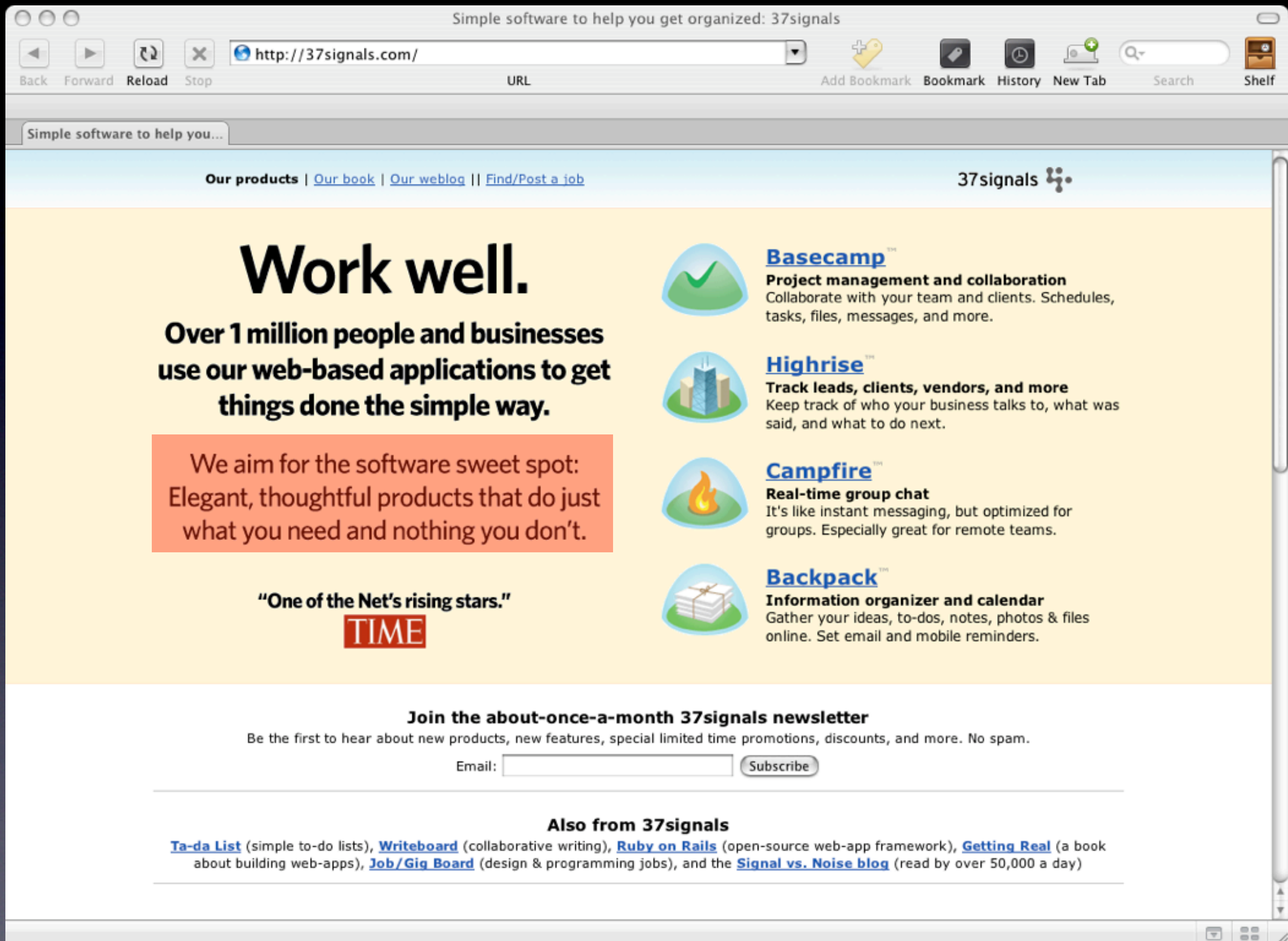
Join the about-once-a-month 37signals newsletter

Be the first to hear about new products, new features, special limited time promotions, discounts, and more. No spam.

Email:

Also from 37signals

[Ta-da List](#) (simple to-do lists), [Writeboard](#) (collaborative writing), [Ruby on Rails](#) (open-source web-app framework), [Getting Real](#) (a book about building web-apps), [Job/Gig Board](#) (design & programming jobs), and the [Signal vs. Noise blog](#) (read by over 50,000 a day)



Ruby on Rails was created by David Heinemeier Hansson, a partner at 37signals, then extended and improved by a core team of committers and hundreds of open-source contributors.

Révolution?

caramba !

Simplification

0

Opiniated software

Design by comitee

Immédiat

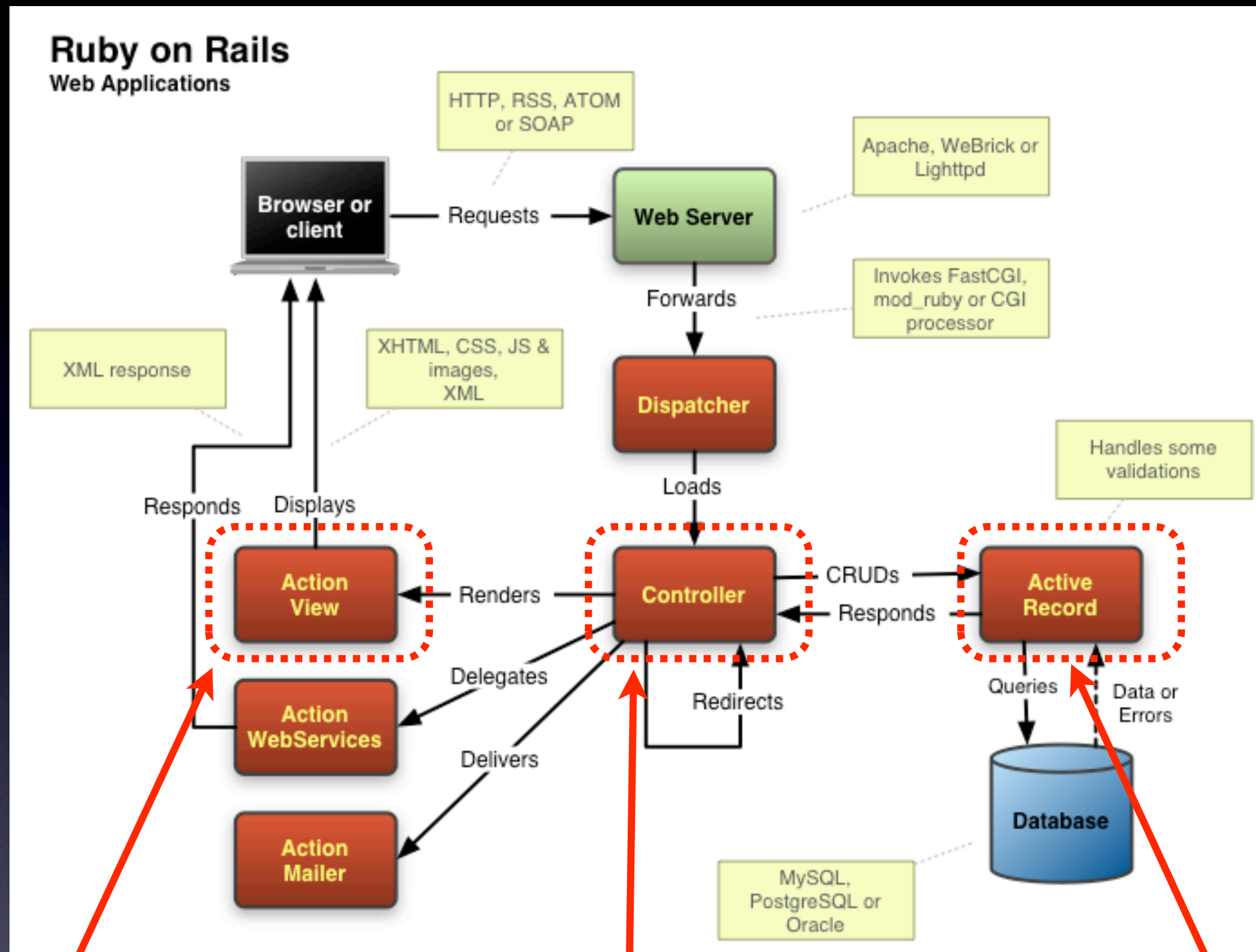
Configurable

Périmètre limité

Exhaustif

Contraint

Flexible



Vue

Contrôleur

Modèle



Conventions

versus

Configuration



Pourquoi expliciter ce qui est vrai dans 90% des cas ?



Don't Repeat Yourself

Pour chaque information, une seule source.
(The Pragmatic Programmer, Andrew Hunt & Dave Thomas)

Non, Rails n'est pas le nouveau
Silver Bullet.

Oui, dans le bon contexte, il
apporte des gains de productivité
étonnants.

A large, white iceberg with blue-tinged edges floats in a dark blue, choppy ocean. The sky is overcast with grey clouds. The word "Rails" is written in a teal color above the iceberg, and "Ruby" is written in white below it.

Rails

Ruby



Ruby est un langage **orienté-objet**, très similaire à Smalltalk, créé par Yukihiro Matsumoto (Matz) autour de 1995.

- *Typé dynamiquement*
- *Open source*
- *Closures*
- *Classes ouvertes*
- *Expressif*



demo IRB

Typage dynamique ?
Réouverture de classes ?

Embrace these changes ! ;-)

Donnons la parole à 2 petits jeunes

Ron Jeffries (2002)

What productivity jump do you see in the near future?

Honestly I don't see one. I hope that the agile methods like XP will help teams to be more productive, and I hope that the world will learn that *dynamically typed languages* like **Ruby** and **Smalltalk** are *more productive* than those with static typing.

Donnons la parole à 2 petits jeunes

Ron Jeffries (2002)

What productivity jump do you see in the near future?

Honestly I don't see one. I hope that the agile methods like XP will help teams to be more productive, and I hope that the world will learn that *dynamically typed languages* like **Ruby** and **Smalltalk** are *more productive* than those with static typing.

"I always knew one day Smalltalk would replace Java. I just didn't know it would be called **Ruby**."

Kent Beck (attr.)

Edition



Compilation



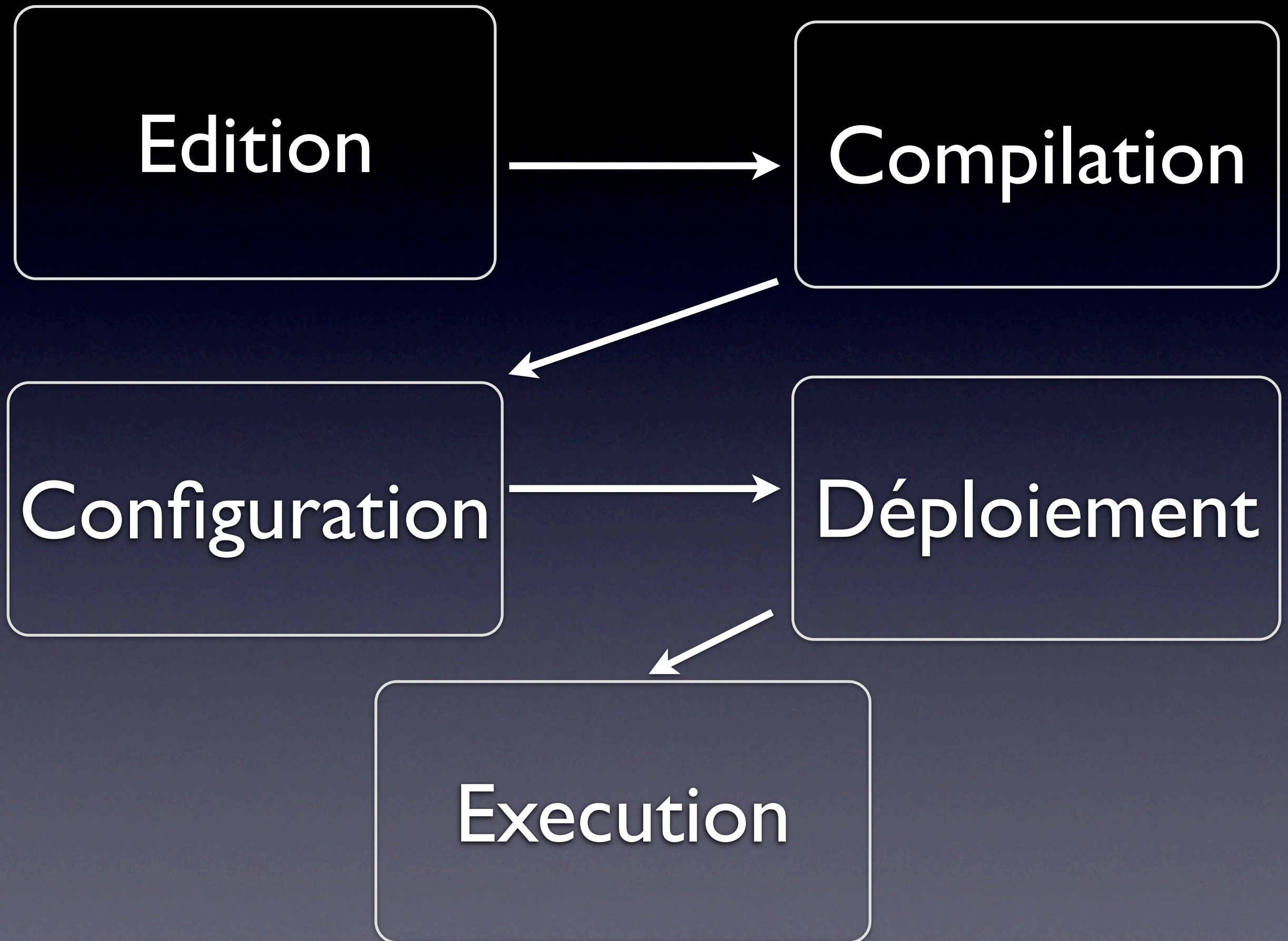
Configuration



Déploiement



Execution







demo

ruby
rails
mySql

demo

3 environnements isolés

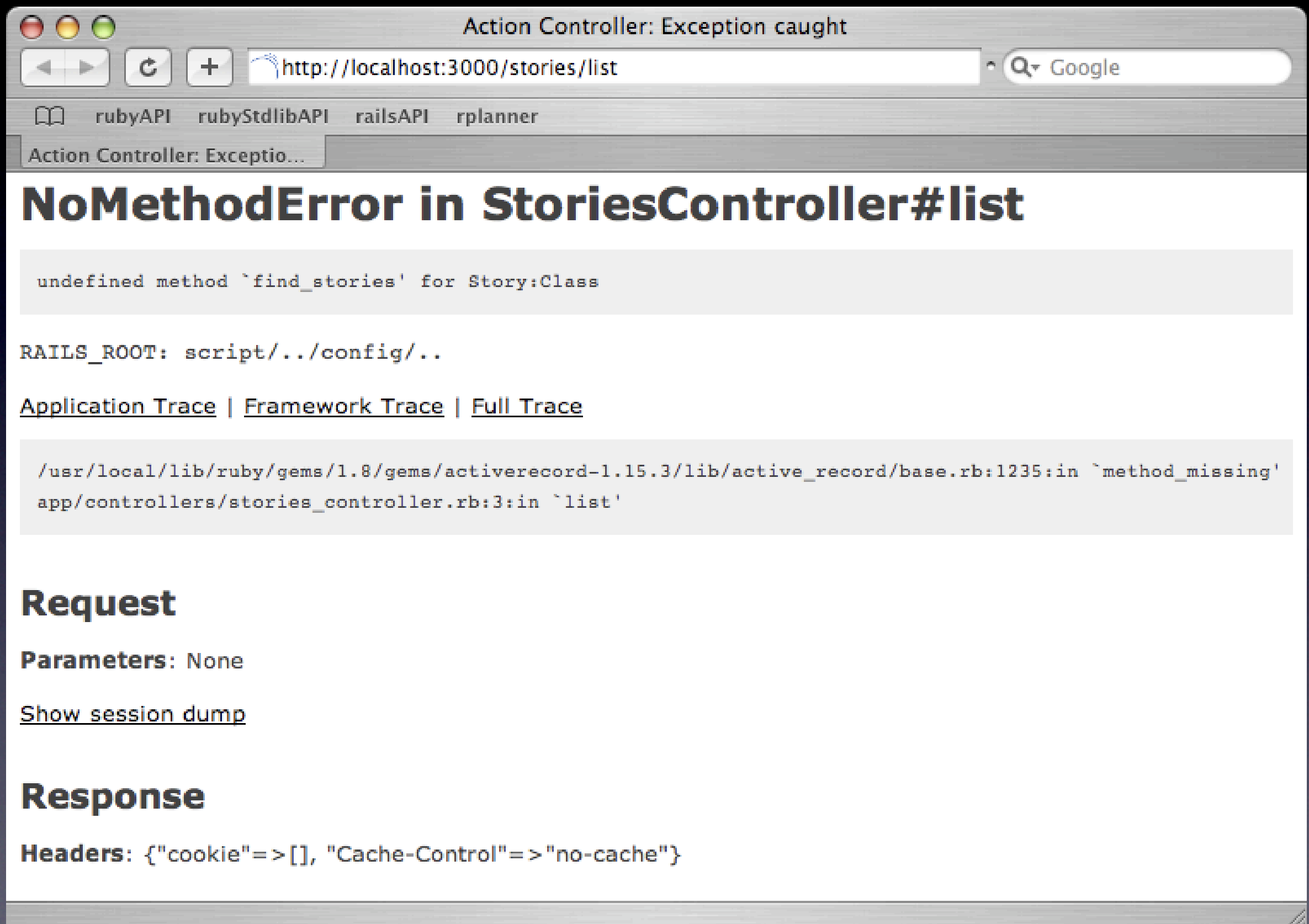
3 bases de données
3 configurations

Développement

Rechargement des classes

Développement

Rechargement des classes
Affichage détaillé des erreurs



Développement

Rechargement des classes
Affichage détaillé des erreurs

Développement

Rechargement des classes
Affichage détaillé des erreurs
Logs exhaustifs

log/development.log

Processing StoryController#update (for 127.0.0.1 at 2007-04-23 15:53:54) [POST]

Session ID: 8bee5e2b229109e54612b6d5748b3a7c

Parameters: {"commit"=>"Edit", "story"=>{"title"=>"show how RoR rocks"}, "action"=>"update", "id"=>"1", "controller"=>"story"}

[4;35;ImStory Columns (0.003269)[0m [0mSHOW FIELDS FROM stories[0m

[4;36;ImStory Load (0.000692)[0m [0;ImSELECT * FROM stories WHERE (stories.`id` = 1) [0m

[4;35;ImSQL (0.000184)[0m [0mBEGIN[0m

[4;36;ImStory Update (0.000359)[0m [0;ImUPDATE stories SET `created_at` = '2007-04-23 15:52:35', `title` = 'show how RoR rocks', `updated_at` = '2007-04-23 15:53:54' WHERE `id` = 1[0m

[4;35;ImSQL (0.000767)[0m [0mCOMMIT[0m

Redirected to <http://localhost:3000/story/show/1>

Completed in 0.01527 (65 reqs/sec) | DB: 0.00527 (34%) | 302 Found [<http://localhost/story/update/1>]

Développement

Rechargement des classes
Affichage détaillé des erreurs
Logs exhaustifs

Test

Données réinitialisées à chaque test

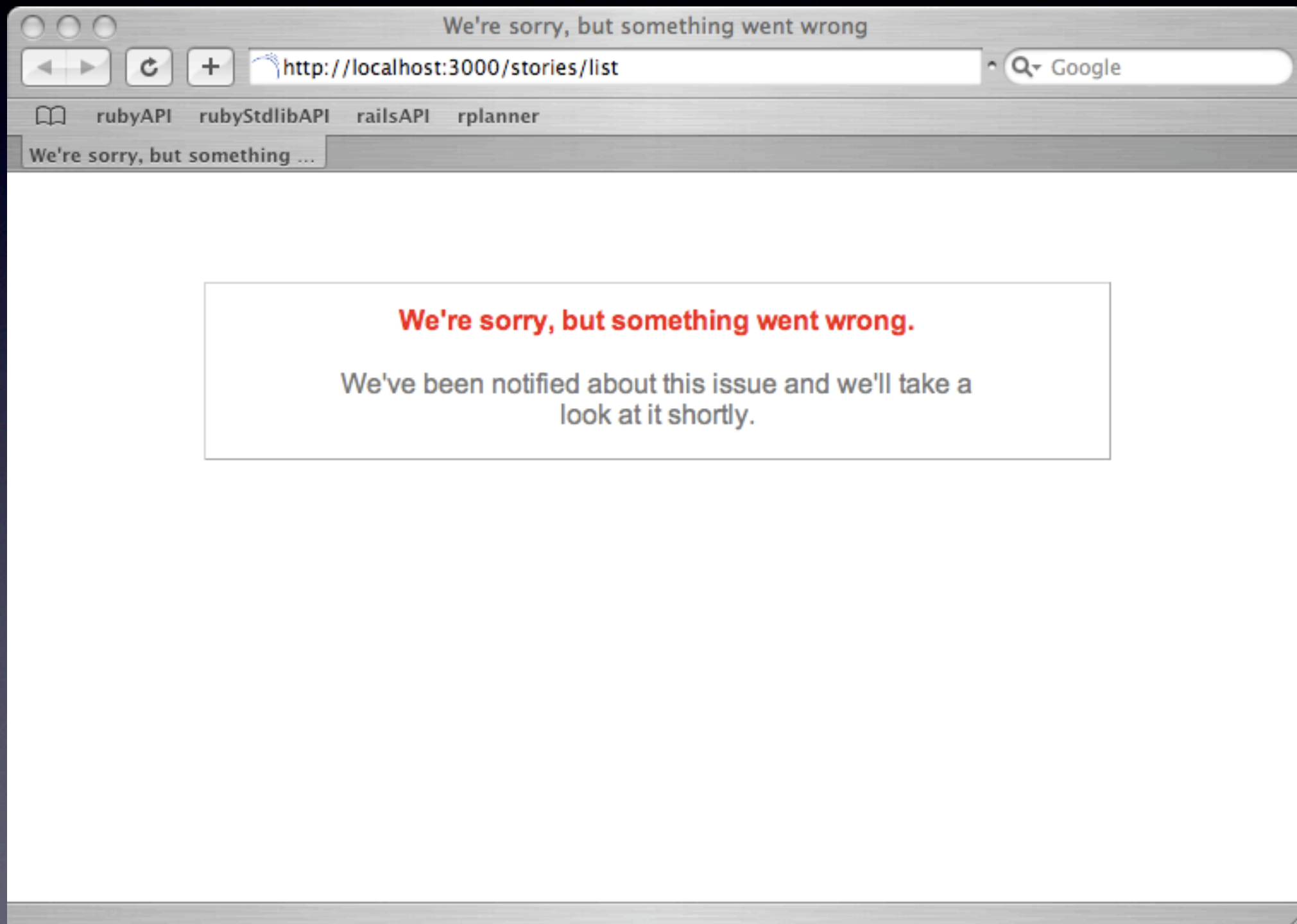
Reproductibilité et Consistance des
tests

Production

Performance

Logs d'erreur uniquement
Affichage simplifié des erreurs

/public/500.html



Production

Performance

Logs d'erreur uniquement
Affichage simplifié des erreurs

demo

initialisation des environnements

Le modèle

pattern “Active Record” de Martin Fowler

Un objet

Un objet

Une ligne de la table

Encapsule les accès base

Logique métier

demo

ActiveRecord

Configuration

```
class Task < ActiveRecord::Base
  set_table_name "tasks"
  set_primary_key "id"
  belongs_to :story, :class_name => "Story", :foreign_key => "story_id"
end
```


~~Configuration~~ Convention

```
class Task < ActiveRecord::Base
  set_table_name "tasks"
  set_primary_key "id"
  belongs_to :story, :class_name => "Story", :foreign_key => "story_id"
end
```


Convention

```
class Task < ActiveRecord::Base
```

```
  belongs_to :story  
end
```


Eager loading

Eager loading

```
Story.find(:all, :include => :tasks)
```


Callbacks

Callbacks

```
class Story < ActiveRecord::Base
  after_create :notify_by_email

  def notify_by_email
    ...
  end
end
```


ActiveRecord, c'est aussi

relations 1-1

relations n-n

transactions

validations

observateurs

composition de modèles

indépendance vis à vis du SGBD

...

ActionController

Les contrôleurs

Les vues

`/stories/show/7?key=val`

`/stories/show/7?key=val`

`config/routes.rb`

`map.connect :controller/:action/:id`

`/stories/show/7?key=val`

`config/routes.rb`

`map.connect :controller/:action/:id`

`app/controllers/stories_controller.rb`

`class StoriesController < ApplicationController`

`end`

`/stories/show/7?key=val`

`config/routes.rb`

`map.connect :controller/:action/:id`

`app/controllers/stories_controller.rb`

```
class StoriesController < ApplicationController
  def show
```

```
    end
```

```
end
```


/stories/show/7?key=val

config/routes.rb

```
map.connect :controller/:action/:id
```

app/controllers/stories_controller.rb

```
class StoriesController < ApplicationController
  def show
    params['id'] # returns '7'
  end
end
```


/stories/show/7?key=val

config/routes.rb

```
map.connect :controller/:action/:id
```

app/controllers/stories_controller.rb

```
class StoriesController < ApplicationController
  def show
    params['id'] # returns '7'
    params['key'] # returns 'val'
  end
end
```


`/stories/show/7?key=val`

`config/routes.rb`

`map.connect :controller/:action/:id`

`app/views/stories/show.rhtml`

`<html> ... <%= ruby code %> ... </html>`

`/stories/show/7?key=val`

`config/routes.rb`

`map.connect :controller/:action/:id`

`app/views/stories/show.rhtml`

`<html> ... <%= ruby code %> ... </html>`

`app/views/stories/show.rxml`

`xml.story do`

`xml.task('xpdayer', :estimation => 3)`

`end`

```
<story>
  <task estimation="3">
    xpdayer
  </task>
</story>
```


demo

ActionController, c'est aussi

Filtres

Sessions

Cookies

Caches

Pagination

et aussi...

Ajax

prototype.js et script.aculo.us.js

les templates RJS

générer du javascript sans écrire de javascript

Rails est orienté *Tests*

Tests

- Tests unitaires
- Tests fonctionnels
- Rake
- Stubs/Mocks
- rspec

Inspiré de la présentation de **Mike Clark**,
<http://www.clarkware.com/>

Tests unitaires Modèle

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ProjectTest < Test::Unit::TestCase
```

```
  fixtures :projects
```

```
  def test_project_must_have_a_name
```

```
    unknown = Project.new
```

```
    assert !unknown.valid?
```

```
  end
```

```
end
```

Dans ../test/unit/project_test.rb

Fixtures

```
alpha:  
  id: 1  
  name: alpha  
beta:  
  id: 2  
  name: beta
```

Dans .../test/fixtures/projects.yml

Fixtures

```
alpha:  
  id: 1  
  name: alpha  
beta:  
  id: 2  
  name: beta
```

```
<% for i in 3..100 %>  
project_<%=i%>:  
  id: <%=i%>  
<% end %>
```

Dans ../test/fixtures/projects.yml

Tests fonctionnels Contrôleur

```
...  
class HomeControllerTest < Test::Unit::TestCase  
  ...  
  # let's test our main index page  
  def test_index  
    get :index  
    assert_response :success  
  end  
end
```

Dans ../test/functional/home_controller_test.rb

Automatisation avec **Rake**

Ecrit par *Jim Weirich*

Quelques *tasks* disponibles dans Rails

```
rake db:test:clone_structure # Recreate the test databases from the development
rake test                  # Test all units and functionals
rake test:functionals      # Run the functional tests in test/functional

rake test:recent           # Test recent changes
rake test:uncommitted      # Test changes since last checkin (only Subversion)
rake test:units            # Run the unit tests in test/unit
```


flexmock en 30s

Ecrit par *Jim Weirich*

```
...-Computer:~ root# gem install flexmock
Need to update 1 gems from http://gems.rubyforge.org
.
complete
Successfully installed flexmock-0.6.0
Installing ri documentation for flexmock-0.6.0...
Installing RDoc documentation for flexmock-0.6.0...
```


flexmock en 30s

```
class TestOrateur < Test::Unit::TestCase

  def test_auditoire_rigole_quand_orateur_presente
    auditoire = flexmock("auditoire")

    auditoire.should_receive(:rigole).once

    st = Orateur.new(auditoire)
    st.presente
  end

end
```


flexmock en 30s

```
class TestOrateur < Test::Unit::TestCase

  def test_auditoire_rigole_quand_orateur_presente
    1) auditoire = flexmock("auditoire")

      auditoire.should_receive(:rigole).once

      st = Orateur.new(auditoire)
      st.presente
    end
  end
end
```


flexmock en 30s

```
class TestOrateur < Test::Unit::TestCase

  def test_auditoire_rigole_quand_orateur_presente
    1) auditoire = flexmock("auditoire")

    2) auditoire.should_receive(:rigole).once

        st = Orateur.new(auditoire)
        st.presente
    end
  end
end
```


flexmock en 30s

```
class TestOrateur < Test::Unit::TestCase

  def test_auditoire_rigole_quand_orateur_presente
    1) auditoire = flexmock("auditoire")

    2) auditoire.should_receive(:rigole).once

        st = Orateur.new(auditoire)
        st.presente 3)
    end

  end

end
```


flexmock en 30s

```
class Orateur
  def initialize(auditoire)
    @auditoire = auditoire
  end
  def presente
    @auditoire.rigole
  end
end
```


rspec <http://rubyforge.org/projects/rspec>

```
...  
context "A new stack" do  
  ...  
context "A stack with one item" do  
  setup do  
    @stack = Stack.new  
    @stack.push 1  
  end  
  specify "should return top when you send it 'top'" do  
    @stack.top.should_equal 1  
  end  
  specify "should not be empty" do  
    @stack.should_not_be_empty  
  end  
end  
...
```


rspec <http://rubyforge.org/projects/rspec>

```
Computer:~ st$ spec stack_spec.rb -f s
```

A new stack

- should be empty

A stack with one item

- should return top when you send it 'top'
- should not be empty

Finished in 0.052539 seconds

3 specifications, 0 failures

Et tout le reste !

Scaffold

ActionMailer

RJS

XMLBuilder

REXML

REST




CAPISTRANO

JRuby

Rails supporte l'Agilité

- Le feedback est immédiat
- Conventions
- Séparation des environnements
- Les tests sont encouragés

Rails a besoin de l'Agilité

-  Dynamisme de l'environnement
tests unitaires "obligatoires"
-  On peut aller rapidement ... dans le mur
besoin de discipline
-  L'adoption des valeurs Agiles aide à l'adoption
de Rails



merci.
des questions ?

Beauty leads to happiness
Happiness leads to productivity
Thus

Beauty leads to productivity

DHH, ruby on rails creator

“Rails is the most thought-out web development framework I ever used. Nobody has done it like this before”

James Duncan Davidson, creator of Tomcat and Ant

“Rails is no match for a Java
based web app six pack of
JSF-Shale-Tiles-SiteMesh-
Hibernate-Spring”

David Geary, Struts committer, co-auteur de Core JSF

Dave Thomas on J2EE

“Using the full might of a J2EE stack to write a small stand-alone application is using a sledgehammer to crack a nut. But I keep hearing the sound of nuts being pulverized as developers seem to think that using anything other than J2EE is somehow unprofessional.”

More pragmatic Dave

“I’d rather write in a language that let’s me focus on the application, and which lets me express myself clearly and effectively.”



la rencontre de l'Agilité et de l'Open Source

