

MacRuby on Rails

Abstract

MacRuby is an implementation of Ruby 1.9 that is built directly on top of Mac OS X core technologies. Recently, MacRuby has become viable as a tool for developing useful desktop applications for Mac OS X. However, as of March 2011, MacRuby is still missing some functionality that is present in cRuby. Therefore, MacRuby is not able to run Ruby on Rails. In my presentation, I will explain how I modified MacRuby to make it a suitable foundation for running Rails. I would also like to explain some of the technical intricacies that I discovered along the way.



RubyConf 2011 - 2011/09/30
Network Applied Communication Laboratory Ltd.
Kouji Takaoka <kouji@netlab.jp>
[@takaokouji_en/takaokouji@facebook.com](http://takaokouji_en/takaokouji@facebook.com)
Illustrator: Yuki Morohoshi <morohoshi@netlab.jp>

This presentation was made possible by **NaCl**



Kouji Takao
宏治 高尾



Work

- Network Applied Communication Laboratory Ltd.
- System design, programming, planning and meeting



Spare Time

- MacRuby committer

→ Core

- cRuby committer

→ Readline module



Contact Information

- kouji.takao@gmail.com
- [@takaokouji_en](https://twitter.com/takaokouji_en)
- <http://facebook.com/takaokouji>
(mainly in Japanese)

My Company

My Company

- Network Applied Communication
Laboratory Ltd.

My Company

- Network Applied Communication Laboratory Ltd.
- a.k.a NaCl

My Company

- Network Applied Communication Laboratory Ltd.
- a.k.a NaCl
→ means salt

My Company

My Company

- <http://www.netlab.jp/>

My Company

- <http://www.netlab.jp/>
- Shimane, Japan

My Company

- <http://www.netlab.jp/>
- Shimane, Japan
- Since 2001

My Company

- <http://www.netlab.jp/>
 - Shimane, Japan
 - Since 2001
- ➡ 10th Anniversary!

My Company

- <http://www.netlab.jp/>
- Shimane, Japan
- Since 2001
 ➡ 10th Anniversary!
- 57 people

My Company

- <http://www.netlab.jp/>
- Shimane, Japan
- Since 2001
 - ➡ 10th Anniversary!
- 57 people
 - ➡ 6 people are cRuby committers

My Company

- <http://www.netlab.jp/>
- Shimane, Japan
- Since 2001
 - ➡ 10th Anniversary!
- 57 people
 - ➡ 6 people are cRuby committers
- SI, Training, etc...

NaCl

Network Applied Communication Laboratory

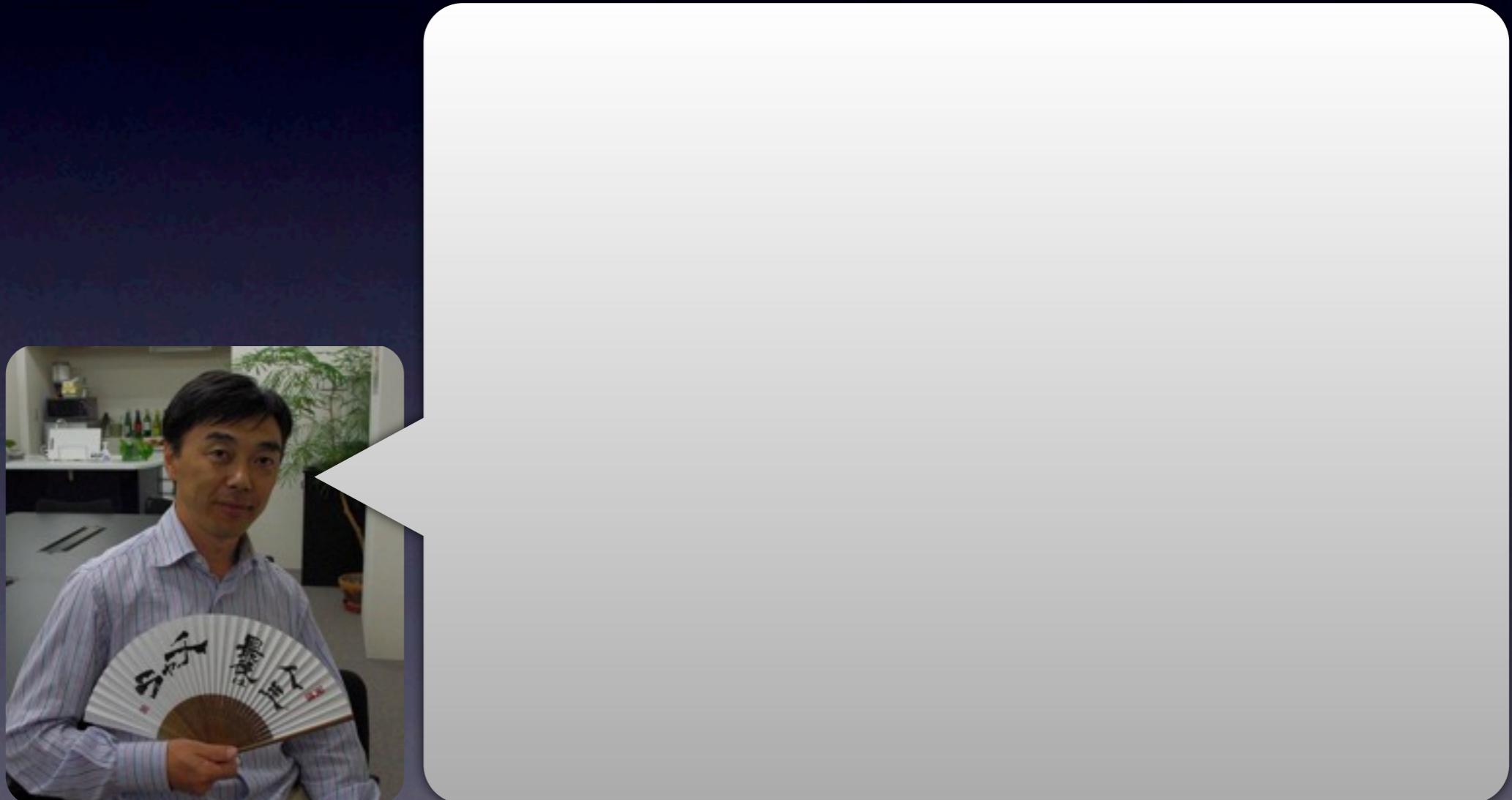
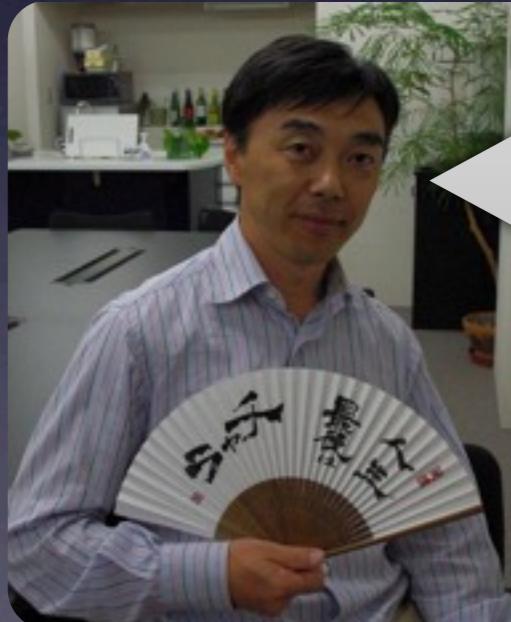
オープンソース・フロンティア
株式会社ネットワーク応用通信研究所

The NaCl Way

The NaCl Way

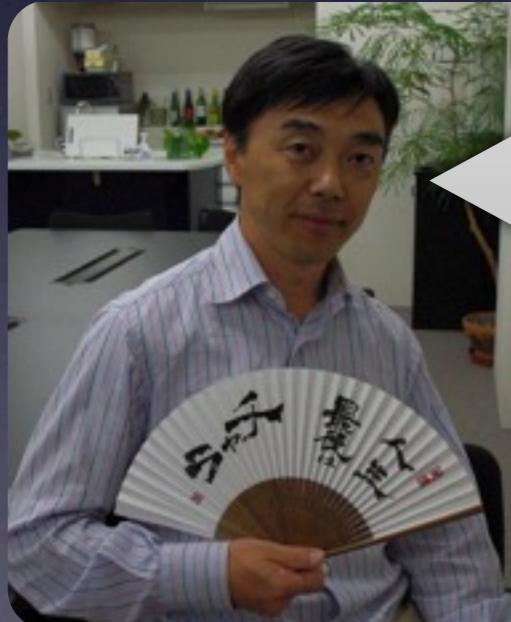


The NaCI Way



The NaCI Way

First, I want **You** to be happy.



The NaCI Way

First, I want **You** to be happy.

Next, I want you to make **Your Family** happy.



The NaCI Way

First, I want **You** to be happy.

Next, I want you to make **Your Family** happy.

Finally, if you have more happiness left, then please share it with **Our Company** :)



The NaCI Way

(as I see it)

The NaCI Way

(as I see it)

Employee

- First: Employee

The NaCI Way

(as I see it)

Employee

>

Employee's
Family

- First: Employee
- Next: Employee Family

The NaCI Way

(as I see it)

Employee

>

Employee's
Family

>>

Company

- First: Employee
- Next: Employee Family
- Finally: Company

The NaCI Way

(as I see it)

Employee > Employee's Family >> Company

- First: Employee
- Next: Employee Family
- Finally: Company

The NaCI Way

(as I see it)

Employee

I love my
company!!

-

-

-



MacRuby on Rails

RubyConf 2011 – 2011/09/30
Network Applied Communication Laboratory Ltd.
Kouji Takaoka <kouji@netlab.jp>
[@takaokouji_en/takaokouji@facebook.com](https://takaokouji_en/takaokouji@facebook.com)
Illustrator: Yuki Morohoshi <morohoshi@netlab.jp>



MacRuby Unofficial Character
Ruby Ninja @lrz

MacRuby on Rails

RubyConf 2011 - 2011/09/30
NetLab Applied Communication Laboratory Ltd.
Ko Takaoka <kouji@netlab.jp>
<<https://www.facebook.com/takaokouji>>
Illustration: Morohoshi <morohoshi@netlab.jp>



MacRuby Unofficial Character
Ruby Ninja @lrz

MacRuby on Rails

RubyConf 2011 - 2011/09/30
NetLab Applied Communication Laboratory Ltd.
Ko Takaoka <kouji@netlab.jp>
<[takaokouji@facebook.com](https://www.facebook.com/takaokouji)>
Illustration: Saki Morohoshi <morohoshi@netlab.jp>

MacRuby on Rails

In my presentation,
I will explain how I modified MacRuby to
make it a suitable foundation for running Rails.
I would also like to explain some of the
technical intricacies that I discovered along the
way.

RubyConf 2011 - 2011/09/30
NetLab Applied Communication Laboratory Ltd.
Ko Takaoka <kouji@netlab.jp>
[@takaokouji](https://facebook.com/takaokouji) takaokouji@facebook.com
Illustration: Saki Morohoshi <morohoshi@netlab.jp>





MacRuby Unofficial Character
Ruby Ninja @lrz

MacRuby on Rails

RubyConf 2011 - 2011/09/30
NetLab Applied Communication Laboratory Ltd.
Ko Takaoka <kouji@netlab.jp>
<<https://www.facebook.com/takaokouji>>
Illustration: Morohoshi <morohoshi@netlab.jp>

MacRuby

Ruby customized for Mac OS X



MacRuby



MacRuby

- MacRuby is a unique blend of Ruby 1.9 and Objective-C.

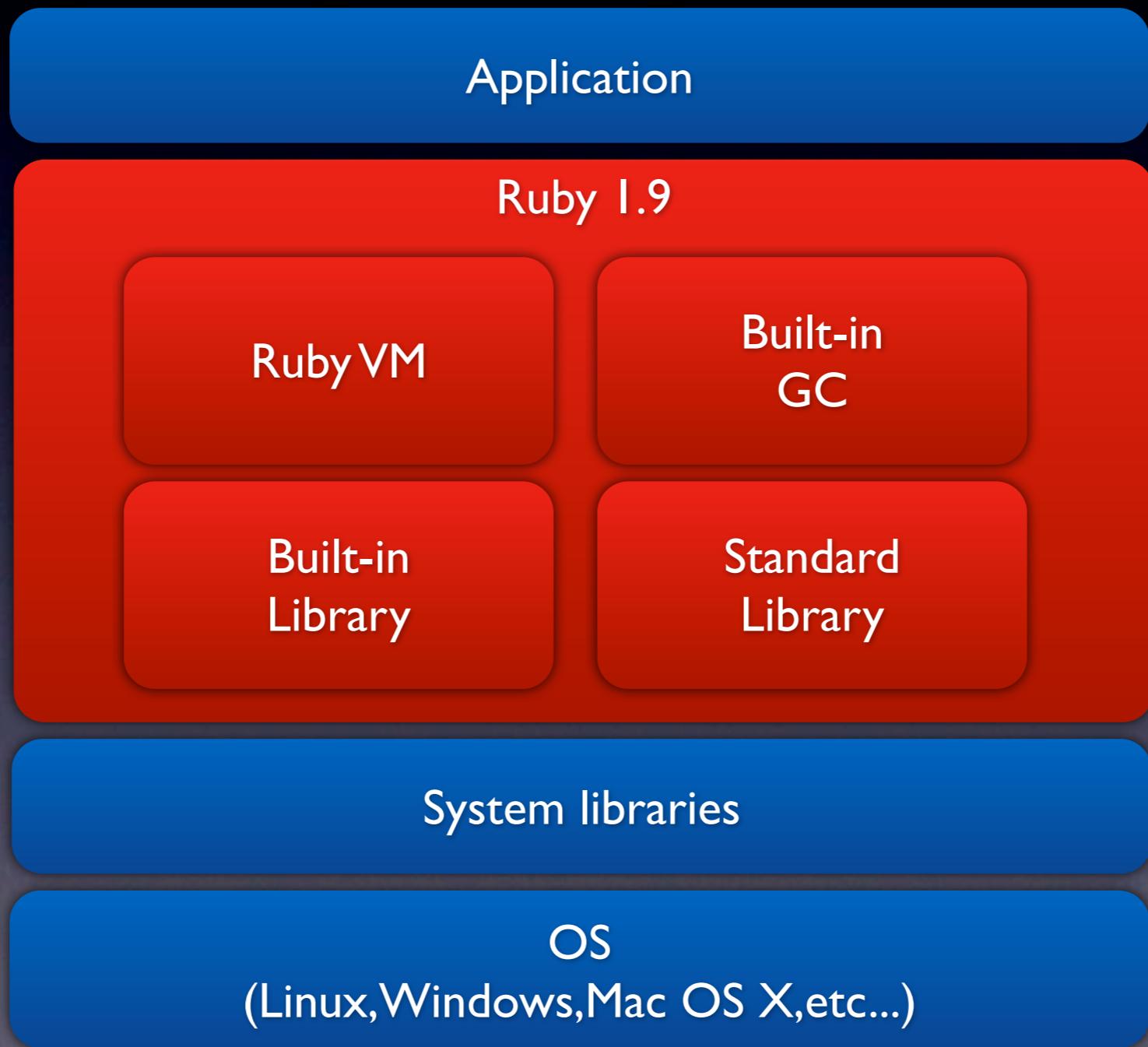


MacRuby

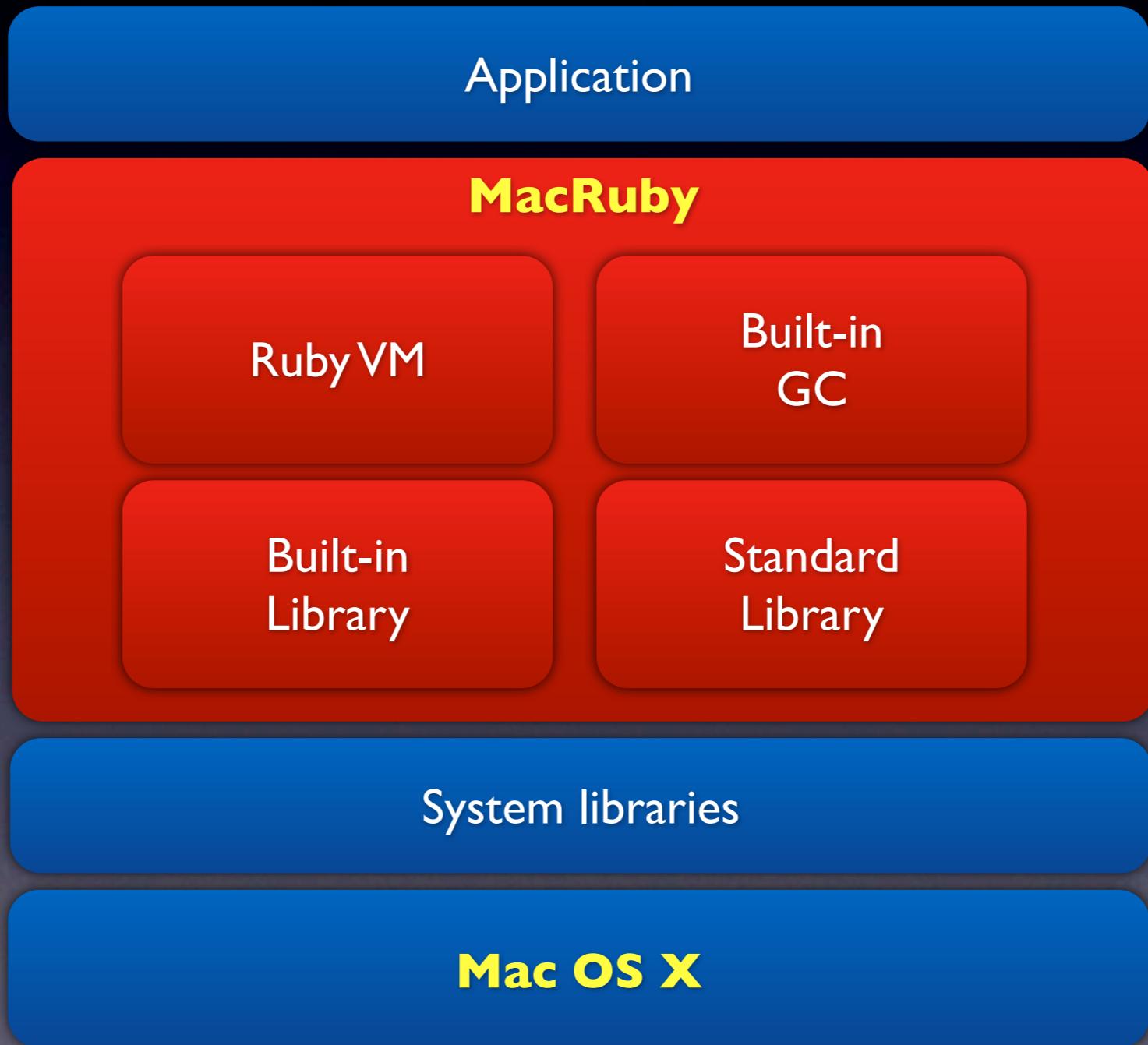
- MacRuby is a unique **blend of Ruby 1.9 and Objective-C**.
- The goal of the MacRuby project is to be **100% compatible** syntactically and behaviorally with Ruby 1.9.



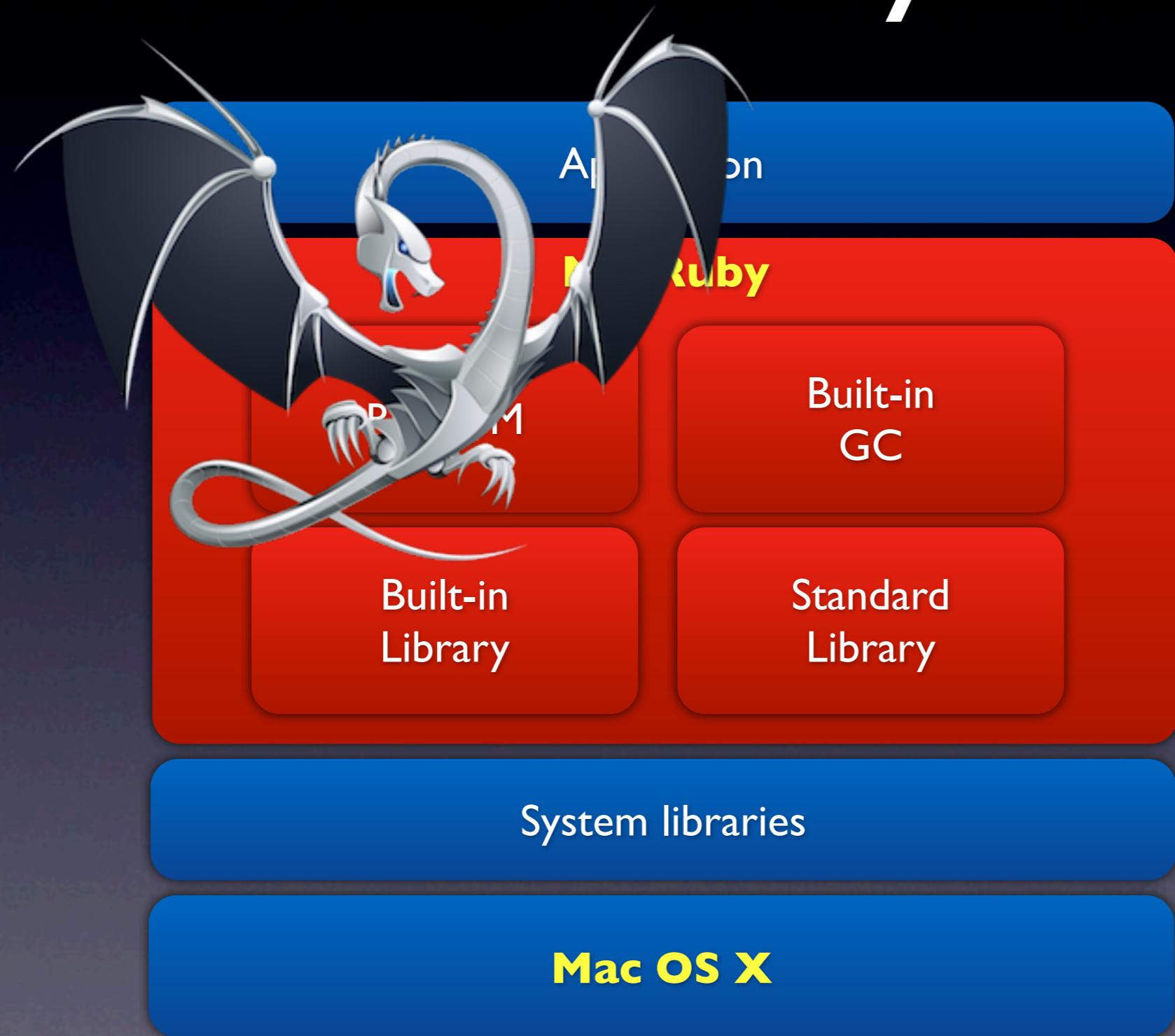
Ruby 1.9



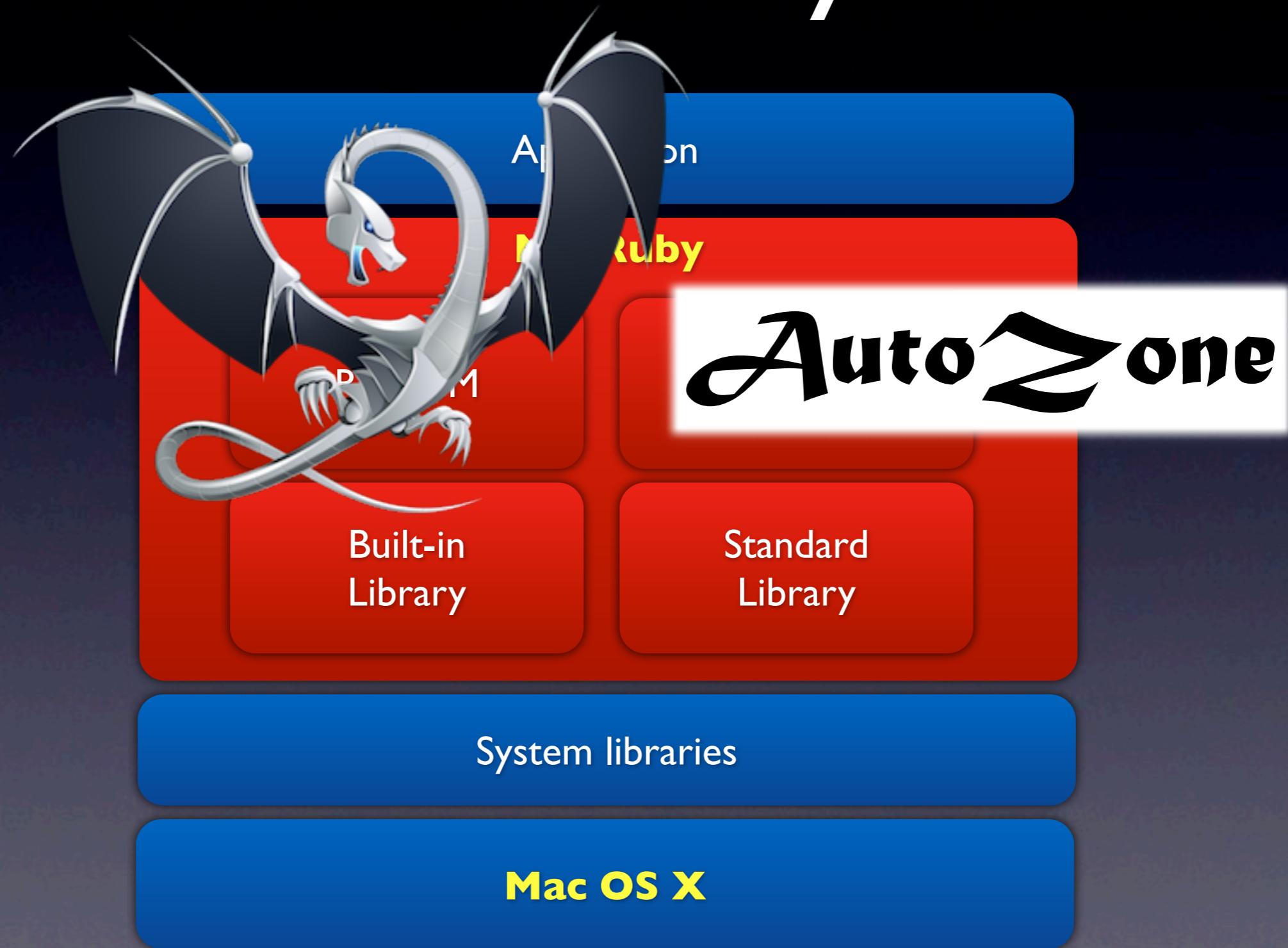
MacRuby



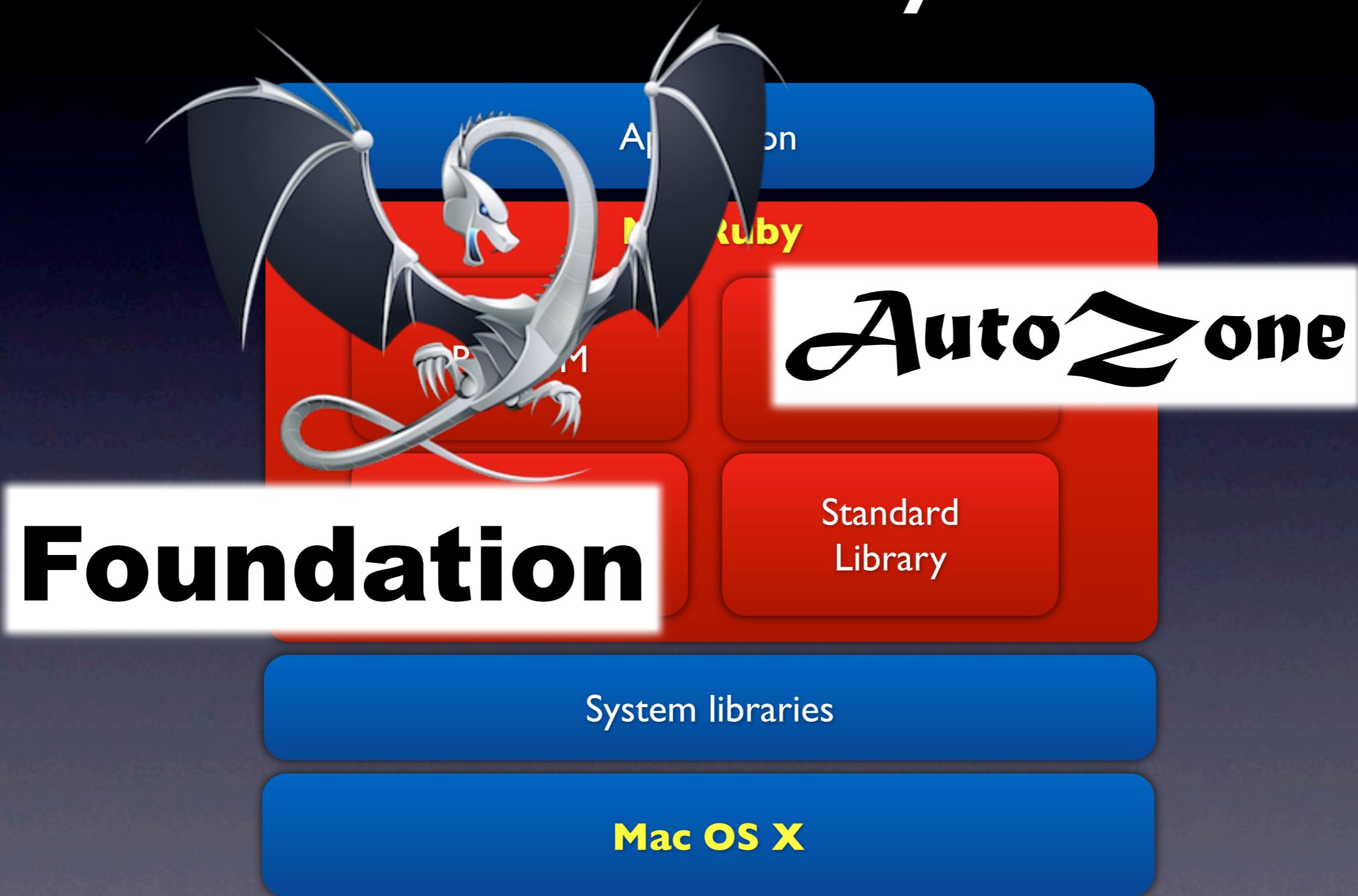
MacRuby



MacRuby



MacRuby



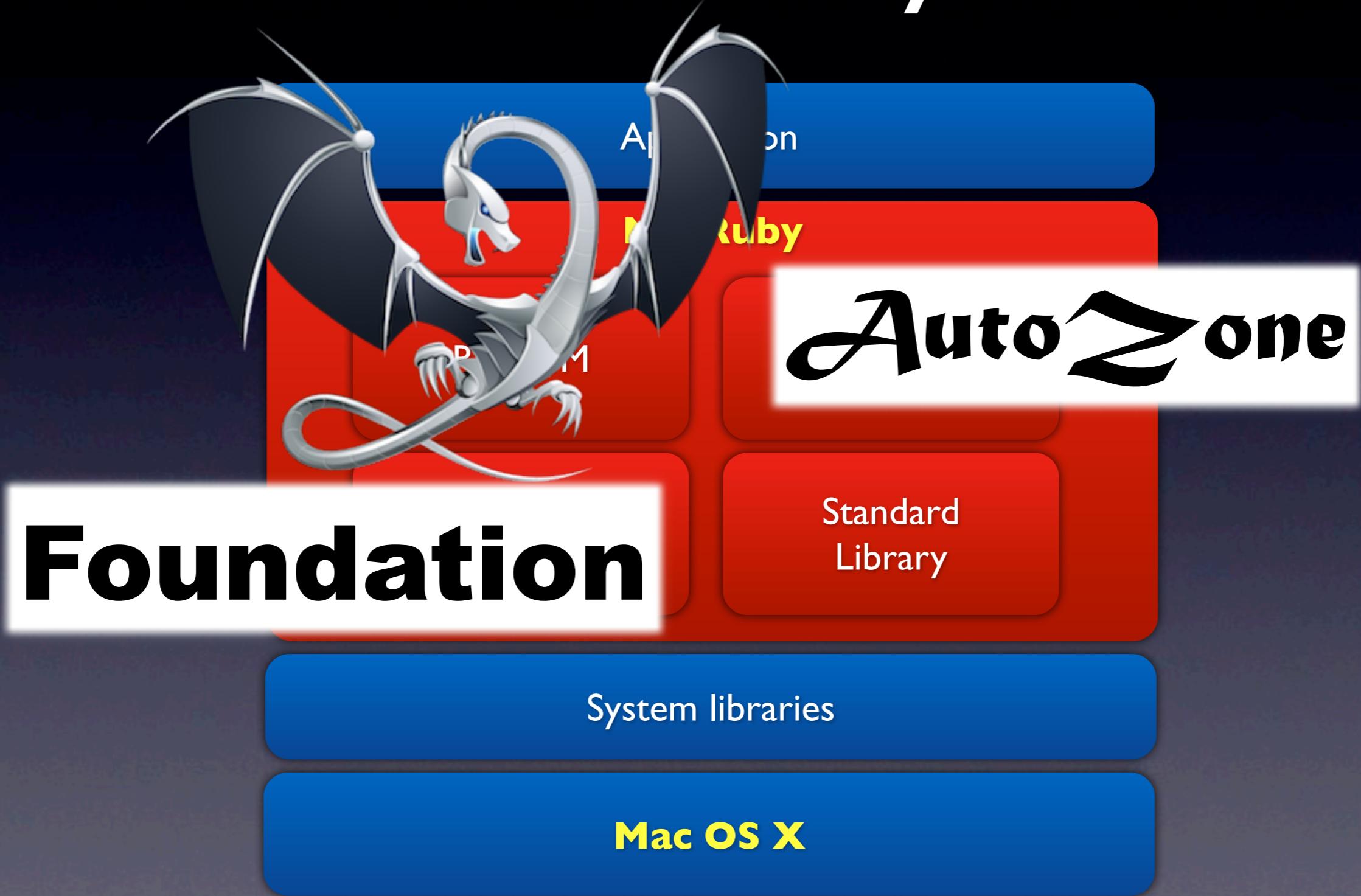
MacRuby



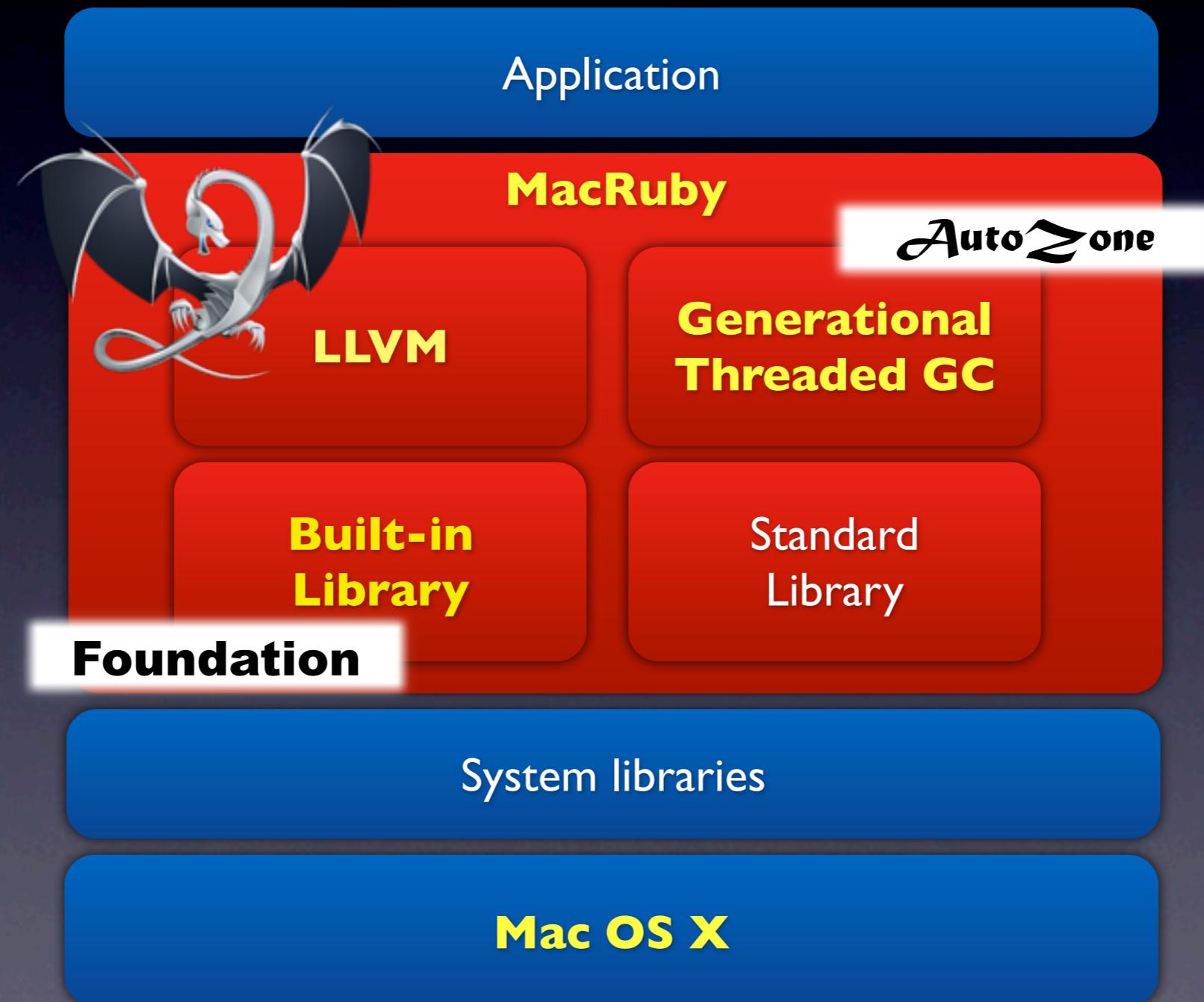
Foundation

```
"kouji".transform("latin-hiragana")  
#=> こうじ
```

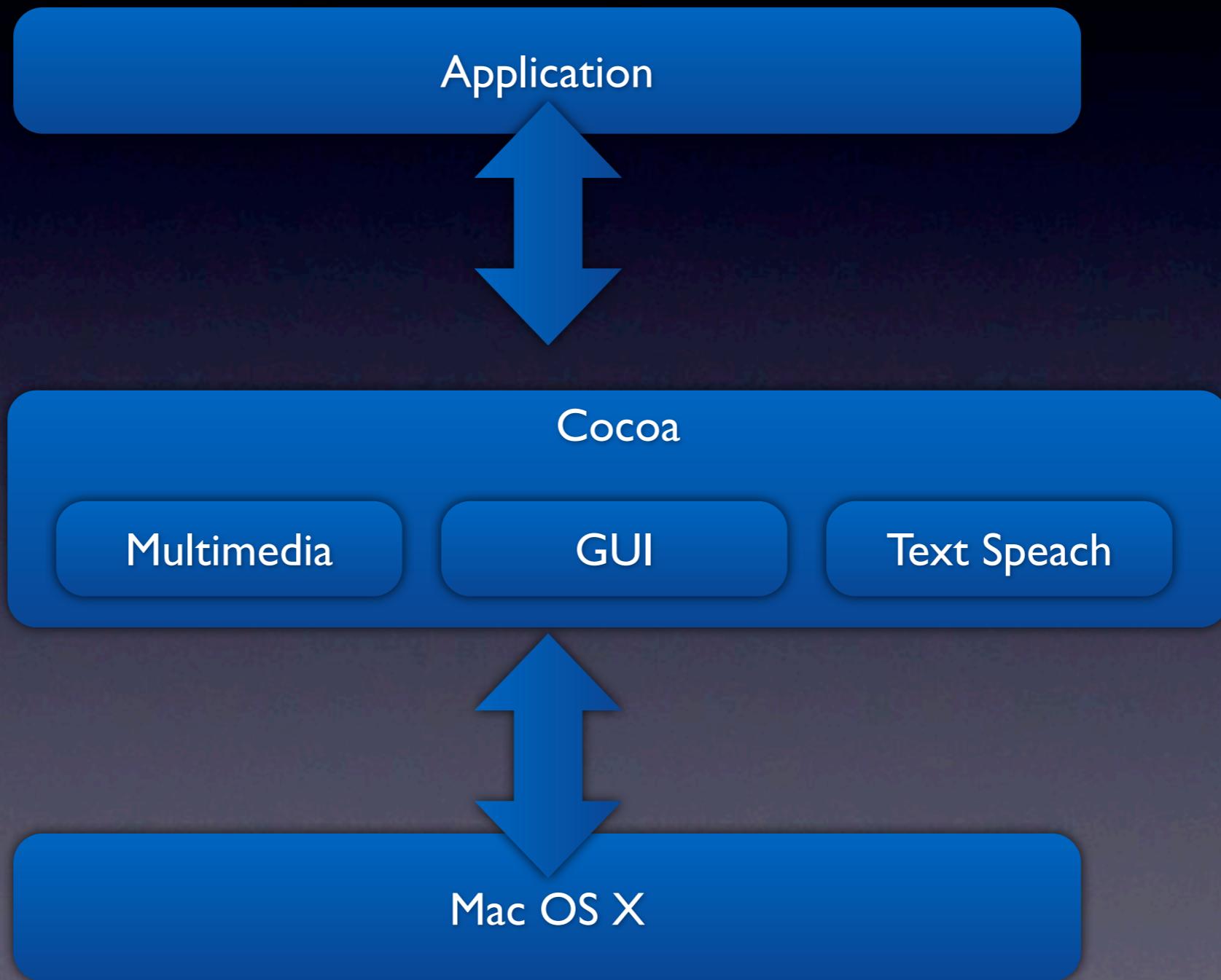
MacRuby



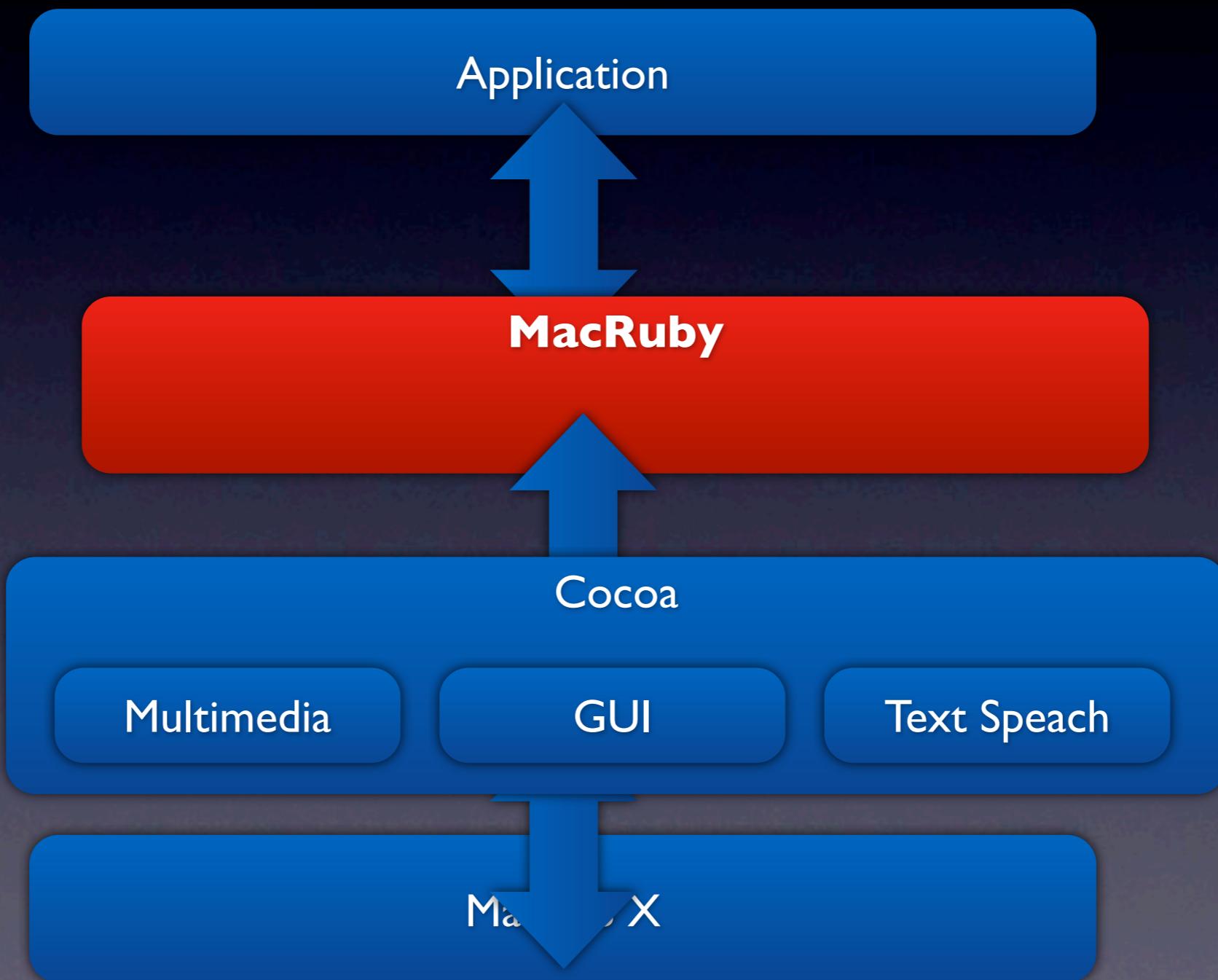
MacRuby



Cocoa



MacRuby X Cocoa



MacRuby X Cocoa

```
framework "cocoa"

def playSound(name)
    sound = NSSound.soundNamed(name)
    sound.play
end
```

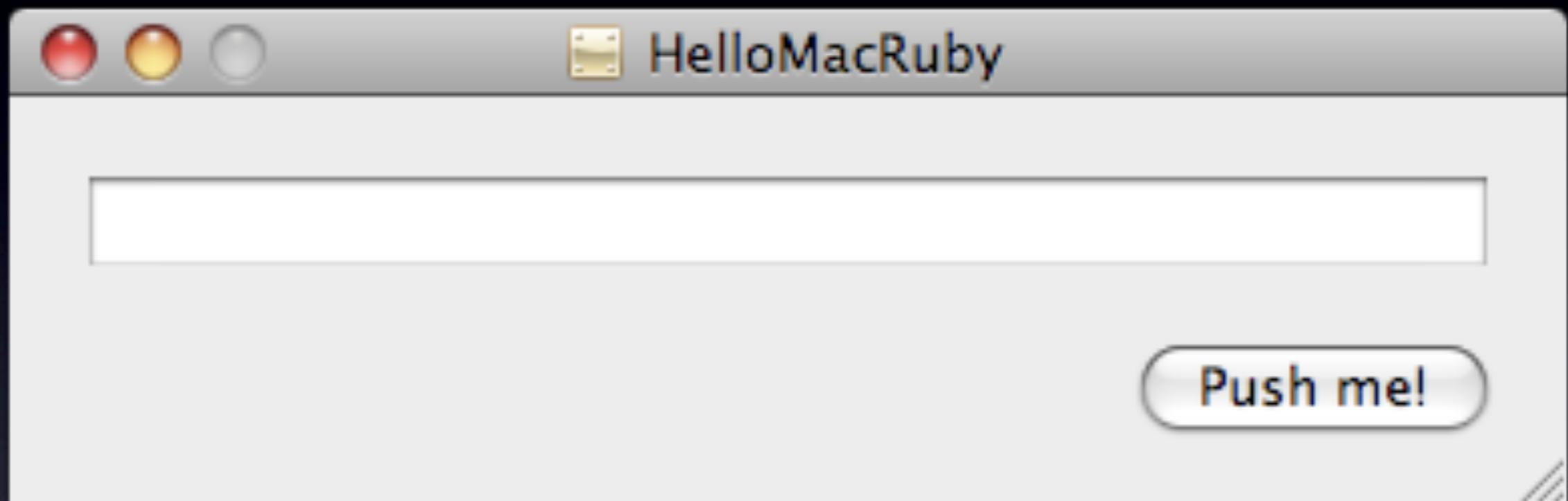
MacRuby X Cocoa

This is a sample program that plays a sound.

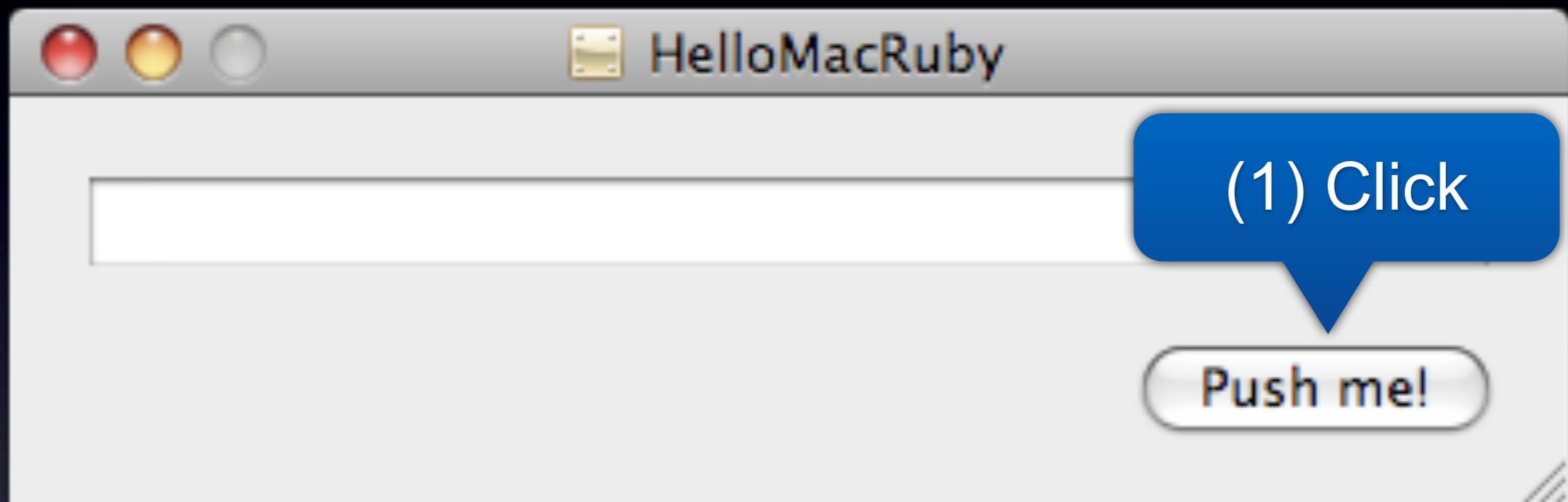
```
framework "cocoa"

def playSound(name)
    sound = NSSound.soundNamed(name)
    sound.play
end
```

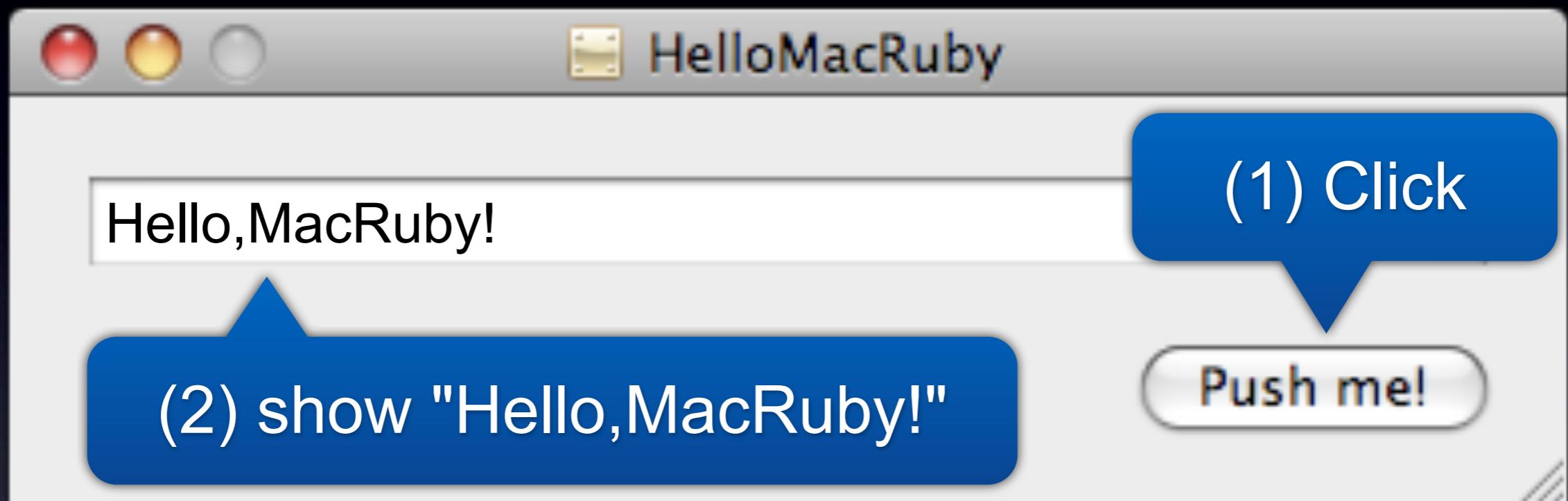
HelloMacRuby



HelloMacRuby



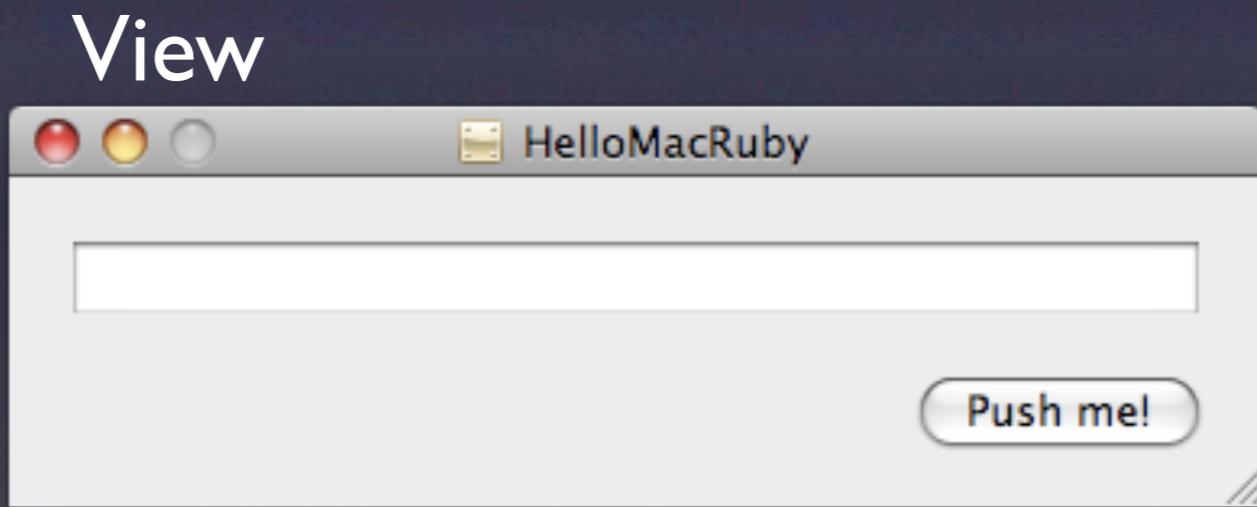
HelloMacRuby



HelloMacRuby

Ruby script: Controller

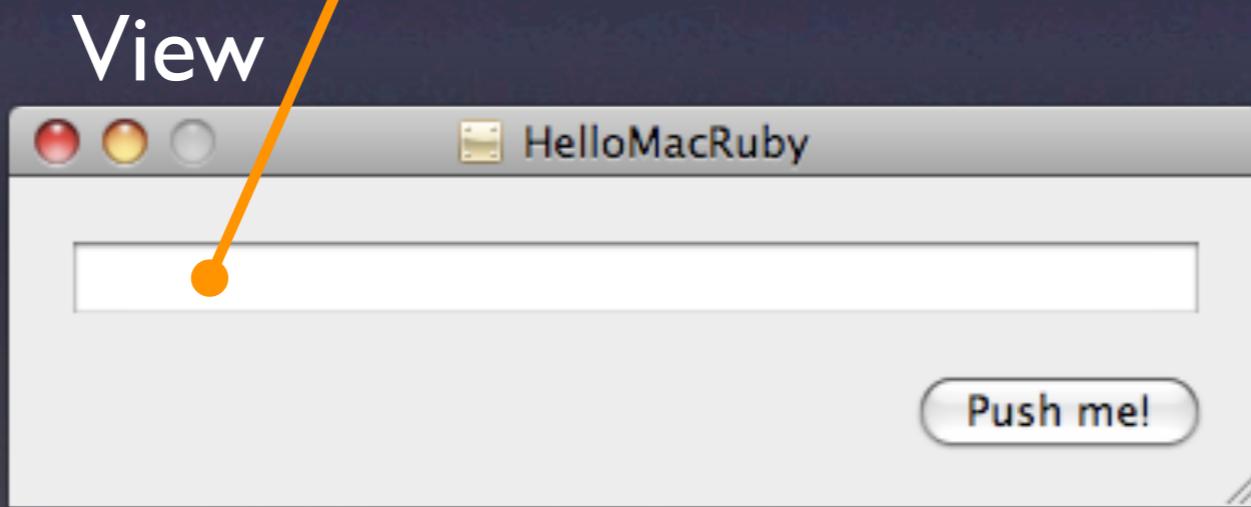
```
1: class AppController
2:   attr_accessor :textField
3:
4:   def hello(sender)
5:     @textField.stringValue = "Hello, MacRuby!"
6:   end
7: end
```



HelloMacRuby

Ruby script: Controller

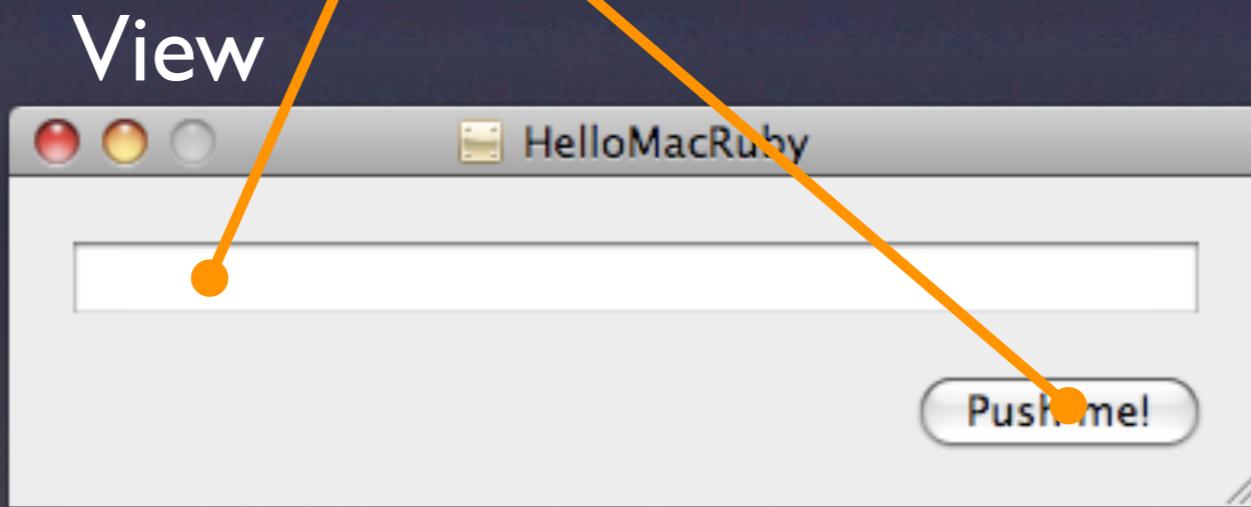
```
1: class AppController
2:   attr_accessor :textField
3:
4:   def hello(sender)
5:     @textField.stringValue = "Hello, MacRuby!"
6:   end
7: end
```



HelloMacRuby

Ruby script: Controller

```
1: class AppController
2:   attr_accessor :textField
3:
4:   def hello(sender)
5:     @textField.stringValue = "Hello, MacRuby!"
6:   end
7: end
```



MacRuby on Rails

Making Ruby on Rails work on MacRuby



Current Status



Current Status

```
$ rails new demo  
kouji:/Users/kouji/work/MacRuby/MacRuby/issues/work:0  
dhcp219$ env VM_DISABLE_RBO=1 macruby -S rails new demo  
create README  
create Rakefile  
create config.ru  
create .gitignore  
create Gemfile  
create app  
create app/controllers/application_controller.rb  
create app/helpers/application_helper.rb  
create app/mailers  
create app/models  
create app/views/layouts/application.html.erb  
create config  
create config/routes.rb  
create config/application.rb  
create config/environment.rb  
create config/environments  
create config/environments/development.rb  
create config/environments/production.rb  
create config/environments/test.rb  
create config/initializers  
create config/initializers/backtrace_silencers.rb
```



Current Status

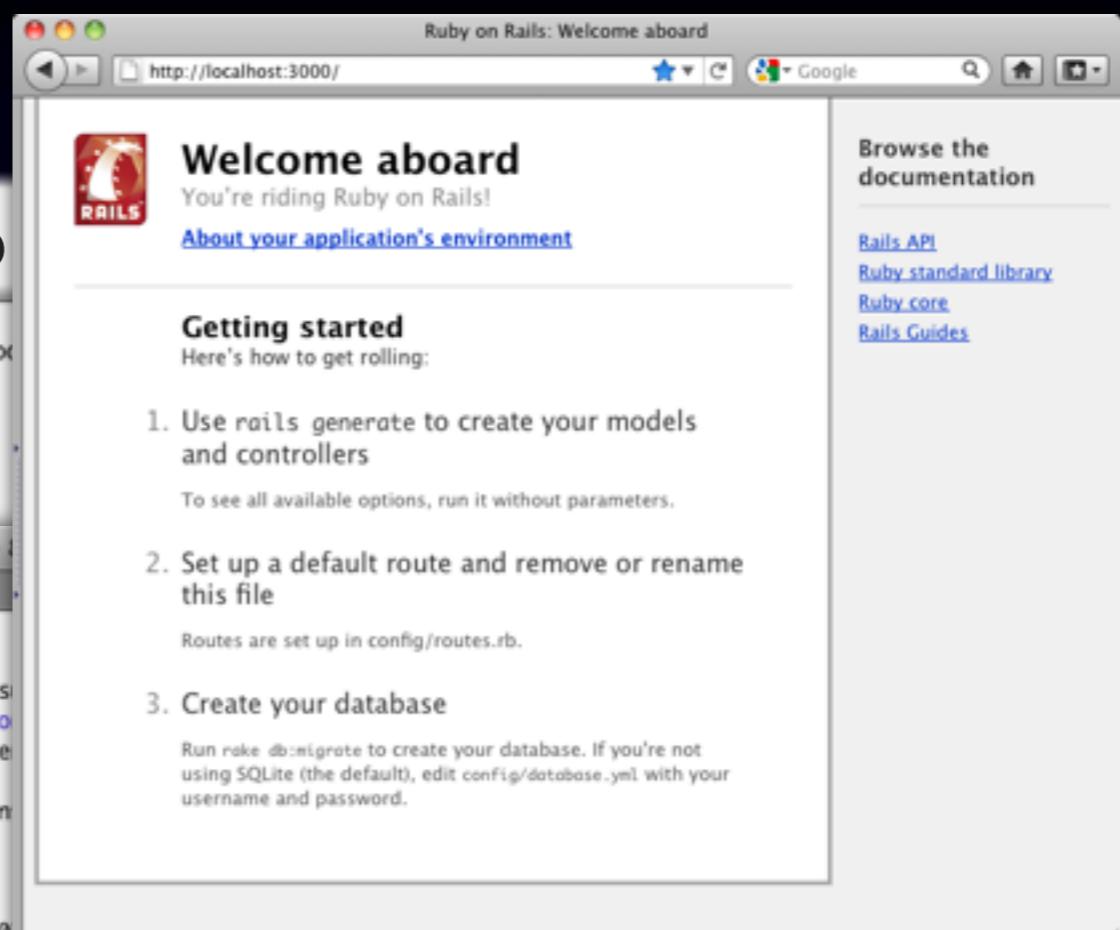
```
$ rails new demo  
kouji:/Users/kouji/work/MacRuby/MacRuby/issues/work:0  
dhcp219$ env VM_DISABLE_RBO=1 macruby -S rails new demo  
  create README  
  create Rakefile  
  create config.ru  
  create .gitignore  
  create Gemfile  
  create app  
  create app/controllers  
  create app/models  
  create app/views  
  create app/javascripts  
  create app/stylesheets  
  create app/assets  
  create app/jobs  
  create app/mailers  
  create app/controllers/bookmarks_controller.rb  
  invoke active_record  
    create db/migrate/20110714234202_create_bookmarks.rb  
  create app/models/bookmark.rb  
  invoke test_unit  
    create test/unit/bookmark_test.rb  
    create test/fixtures/bookmarks.yml  
  create config/routes.rb  
  route resources :bookmarks  
  invoke scaffold_controller  
    create app/controllers/bookmarks_controller.rb  
  invoke erb  
    create app/views/bookmarks  
    create app/views/bookmarks/index.html.erb  
    create app/views/bookmarks/edit.html.erb  
    create app/views/bookmarks/show.html.erb  
    create app/views/bookmarks/new.html.erb  
    create app/views/bookmarks/_form.html.erb  
  invoke test_unit  
    create test/functional/bookmarks_controller_test.rb  
  invoke helper  
    create app/helpers/bookmarks_helper.rb  
  invoke test_unit
```



Current Status



Current Status



7 months ago...



7 months ago...

March 2011



7 months ago...

March 2011

- I thought, "Does Rails work on MacRuby?".



7 months ago...

March 2011

- I thought, "Does Rails work on MacRuby?".
- I managed to **install** Rails.

```
$ sudo macgem install rails  
7 gems installed
```

```
$ sudo macgem install sqlite3-ruby  
2 gems installed
```



Perhaps...



Perhaps...

- I thought, "OK, it installed, but surely the Rails generators won't work...".



Perhaps...

- I thought, "OK, it installed, but surely the Rails generators won't work...".
- However, I managed to **create a Rails project!?**

```
$ macruby -S rails new demo
      create
      create  README
...
      create  vendor/plugins
      create  vendor/plugins/.gitkeep
```



Are you kidding me?



Are you kidding me?

- I tried to generate a scaffold...



Are you kidding me?

- I tried to generate a scaffold...
- It failed. After all Rails didn't work on MacRuby.

```
$ macruby -S rails generate scaffold Bookmark  
title:string description:text url:string  
...  
Assertion failed: ((size_t)pos <  
current_exceptions.size()), function  
pop_current_exception, file vm.cpp, line 3434.  
zsh: abort    macruby -S rails generate scaffold  
Bookmark title:string description:text
```



Are you kidding me?

- I tried to generate a scaffold...
- It failed. After all Rails didn't work on MacRuby.

```
$ macruby -S rails generate scaffold Bookmark  
title $ macruby -S rails server  
...  
Assertion failed: ((size_t)pos < current_exceptions.size()),  
function pop_current_exception, file vm.cpp, line 3434.  
zsh: abort    macruby -S rails server  
po  
zsh
```

Bookmark title:string description:text



My approach

Run Rails

SEGV or abort

Research

Coding

Run RubySpec

Run Rails



My approach



Required knowledge
and technology

- cRuby
- Objective-C
- C++
- LLVM



#860 catch/throw

→ MacRuby aborts if you use catch and throw inside a rescue clause.



#860 catch/throw

→ MacRuby aborts if you use catch and throw inside a rescue clause.

```
01: begin
02:   raise "A"
03: rescue
04:   catch(:foo) { throw :foo }
05: end
```



#860 catch/throw

→ MacRuby aborts if you use catch and throw inside a rescue clause.

```
01: begin
02:   raise "A"
03: rescue
04:   catch(:foo) { throw :foo }
05: end
```

:ok

Assertion failed: ((size_t)pos < current_exceptions.size()),
function pop_current_exception, file vm.cpp, line 3448.



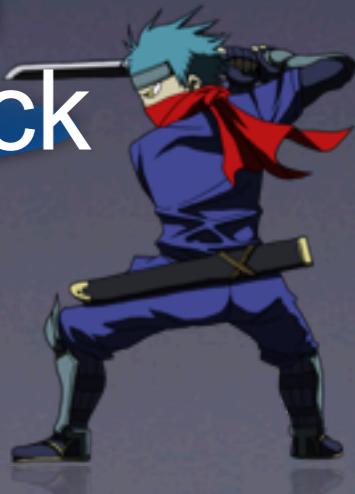
Exception

```
01: begin  
push 02: raise "A"  
  
03: rescue  
  
04: begin  
push 05: raise "B"  
  
06: rescue  
  
pop 07: end  
  
pop 08: end
```

"B"

"A"

Exception Stack



Exception and catch / throw

```
01: catch(:foo) {  
02:   begin  
03:     raise "A"  
04:   rescue  
05:     throw :foo  
06:   end  
07: }
```

push

pop

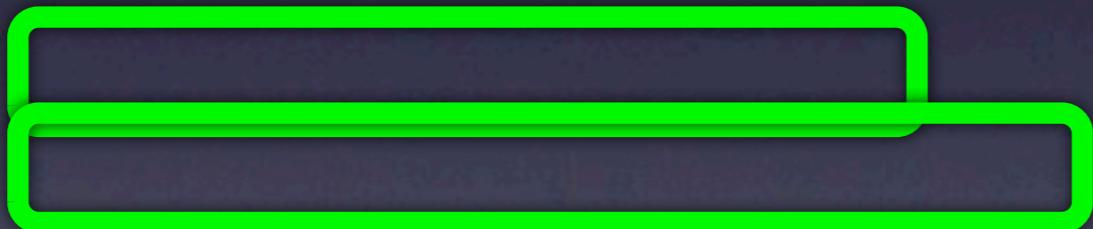
"A"

Exception Stack



throw

vm.cpp:4321:



throw

vm.cpp:4321:

```
VALUE
RoxorVM::ruby_throw(VALUE tag, VALUE value)
{
...
    if current_exception() != Qnil) {
        pop_current_exception();
    }
...
```



throw

Before being fixed(60723bf~), 'throw' always popped the current Exception.

vm.cpp:4321:

```
VALUE
RoxorVM::ruby_throw(VALUE tag, VALUE value)
{
...
    if current_exception() != Qnil) {
        pop_current_exception();
    }
...
```



#860 catch/throw

push

pop

pop

Abort!!

"A"

Exception Stack



#860 catch/throw

```
01: begin  
push 02:    raise "A"  
03: rescue  
pop 04:    catch( :foo ) { throw :foo }  
pop 05: end
```

Abort!!

"A"

Exception Stack

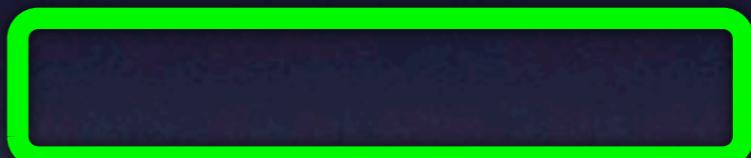
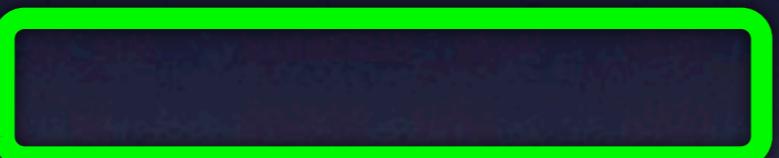


How to fix

push

store

pop



"A"

Exception Stack



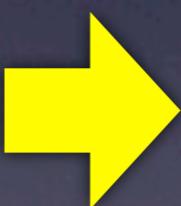
How to fix

push

```
01: begin
02:   raise "A"
03: rescue
04:   catch(:foo) { throw :foo }
05: end
```

store

pop



"A"

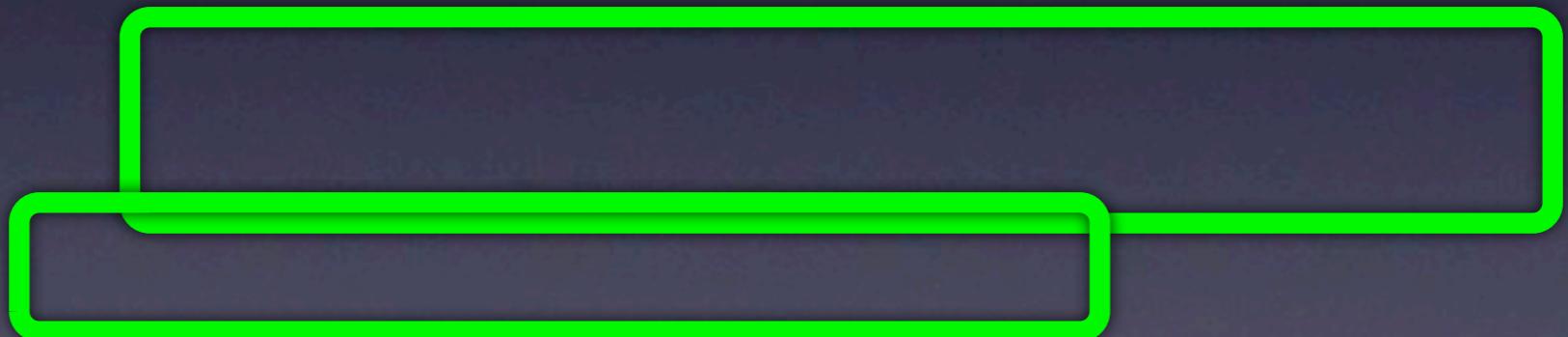
Exception Stack



How to fix

I changed the implementation so that MacRuby only
pop Exceptions if it needs to.(60723bf)

vm.cpp:4318:



How to fix

I changed the implementation so that MacRuby only pop Exceptions if it needs to.(60723bf)

vm.cpp:4318:

```
VALUE
RoxorVM::ruby_throw(VALUE tag, VALUE value)
{
...
while (catch_ptr->current_exception != current_exception()) {
    pop_current_exception();
}
...
```



#860 Fixed

push

store

push

pop

pop

```
01: begin
02:   raise "A"
03: rescue
04:   catch(:foo) {
05:     begin
06:       raise "B"
07:     rescue
08:       throw :foo
09:     end
10:   }
11: end
```

"B"

"A"



Exception Stack



#1192 Constant Lookup

→ Almost can't lookup nested constants.



#1192 Constant Lookup

→ Almost can't lookup nested constants.

```
module A
  B = 10
  Object.class_eval { B }
end
```

cRuby

MacRuby



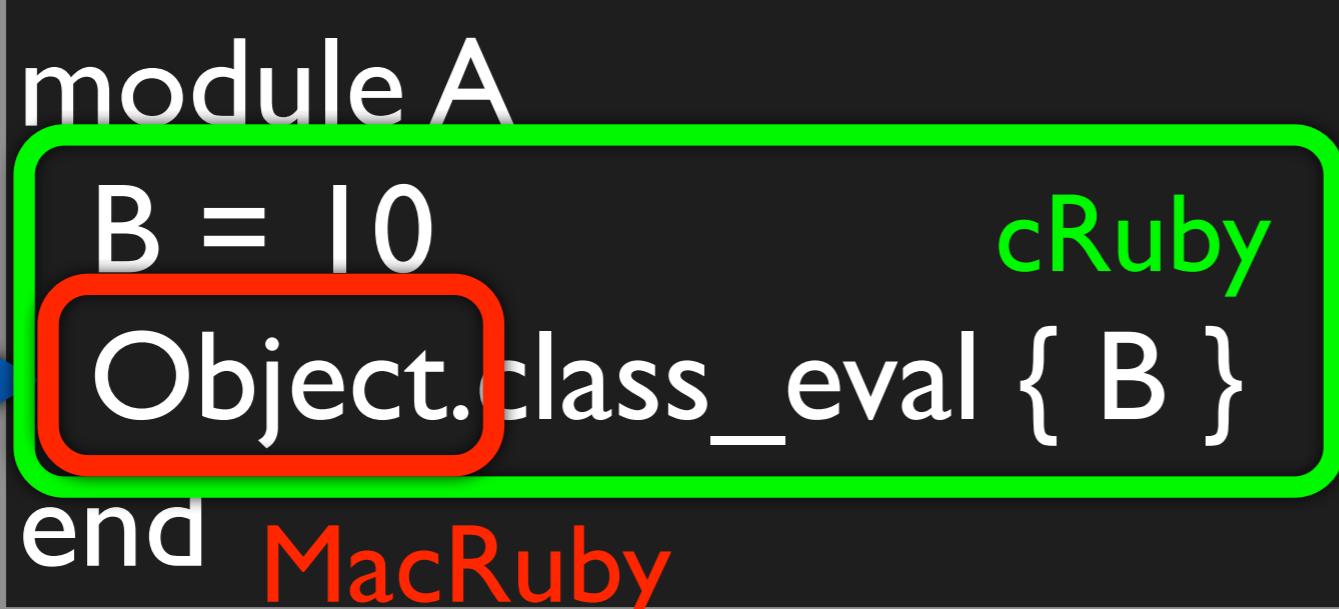
#1192 Constant Lookup

→ Almost can't lookup nested constants.

```
module A
  B = 10
  Object.class_eval { B }
end
```

cRuby

MacRuby



reduction.rb:3:in `block': uninitialized constant
B (NameError)



Lexical Constant Lookup



cRuby

Lexical Scope

Lexical Constant Lookup



Lexical Constant Lookup

```
1 module A
2   module B
3     CONST = "B's Const"
4     module ::A
5       p CONST
6     end
7   end
8 end
```

cRuby

Lexical Scope

Lexical Constant Lookup



But...

MacRuby had not implemented Lexical Constant Lookup.

O MacRuby



But...

MacRuby had not implemented Lexical Constant Lookup.

```
1:module A
2: module B
3:   CONST = "B's Const"
4:   module ::A MacRuby
5:     p CONST
6:   end
7: end
8: end
```



But...

MacRuby had not implemented Lexical Constant Lookup.

```
1:module A
2: module B
3:   CONST = "B's Const"
4:   module ::A MacRuby
5:     p CONST
6:   end
7: end
8: end
```

-e:5:in `block': uninitialized constant A::CONST (NameError)



Constant Lookup is complex

class_eval(&Block)

```
01: module A
02:   CONST = "A's CONST"
03:   def f()
04:     class_eval {
05:       p CONST
06:     }
07:   end
08: end
09:
10: class K
11:   CONST = "K's CONST"
12:   extend A
13:   f()
14: end
```

class_eval(String)

```
01: module A
02:   CONST = "A's CONST"
03:   def f()
04:     class_eval <<-EOS
05:       p CONST
06:     EOS
07:   end
08: end
09:
10: class K
11:   CONST = "K's CONST"
12:   extend A
13:   f()
14: end
```

class_eval(&Block)

cRuby

```
01 module A
02   CONST = "A's CONST"
03   def f()
04     class_eval {
05       p CONST
06     }
07   end
08 end
09
10 class K
11   CONST = "K's CONST"
12   extend A
13   f()
14 end
```

Lexical Scope



```
class_eval(&Block)
```

cRuby

```
01 module A
02   CONST = "A's CONST"
03   def f()
04     class_eval {
05       p CONST
06     }
07   end
08 end
```

Lexical Scope

"A's CONST"

```
10: class K
11:   CONST = "K's CONST"
12:   extend A
13:   f()
14: end
```



class_eval(String)

```
01: module A
02:   CONST = "A's CONST"
03:   def f()
04:     class_eval <<-EOS
05:       p CONST
06:     EOS
07:   end
08: end
09:
10: class K
11:   CONST = "K's CONST"
12:   extend A
13:   f()
14: end
```

cRuby

Dynamic Scope



class_eval(String)

```
01: module A
02:   CONST = "A's CONST"
03:   def f()
04:     class_eval <<-EOS
05:       p CONST
06:     EOS
07:   end
08: end
```

cRuby

"K's CONST"

```
10: class K
11:   CONST = "K's CONST"
12:   extend A
13:   f()
14: end
```

Dynamic Scope



Constant Lookup is complex

class_eval(String)

```
01:module A
02: CONST = "A's CONST"
03: def f
04:   class_eval <<-EOS
05:     p CONST
06:   EOS
07: end
08:end
09:
10:class K
11: CONST = "K's CONST"
12: extend A
13: f
14end
```

class_eval(&Block)

```
module A
CONST = "A's CONST"
def f
  class_eval {
    p CONST
  }
end
end

class K
CONST = "K's CONST"
extend A
f
end
```



Constant Lookup is complex

class_eval(String)

```
01:module A
02: CONST = "A's CONST"
03: def f
04:   class_eval <<-EOS
05:     p CONST #> "K's CONST"
06:   EOS
07: end
08:end
09:
10:class K
11: CONST = "K's CONST"
12: extend A
13: f
14end
```

class_eval(&Block)

```
module A
CONST = "A's CONST"
def f
  class_eval {
    p CONST
  }
end
end

class K
CONST = "K's CONST"
extend A
f
end
```



Constant Lookup is complex

class_eval(String)

```
01:module A
02: CONST = "A's CONST"
03: def f
04:   class_eval <<-EOS
05:     p CONST #=> "K's CONST"
06:   EOS
07: end
08:end
09:
10:class K
11: CONST = "K's CONST"
12: extend A
13: f
14end
```

class_eval(&Block)

```
module A
CONST = "A's CONST"
def f
  class_eval {
    p CONST #=> "A's CONST"
  }
end
end

class K
CONST = "K's CONST"
extend A
f
end
```



Constant Lookup is complex

class_eval(String)

```
01:module A
02: CONST = "A's CONST"
03: def f
04:   class_eval <<-EOS
05:     p CONST #=> "K's CONST"
06:   EOS
07: end
08:end
09:
10:class K
11: CONST = "K's CONST"
12: extend A
13: f
14end
```

class_eval(&Block)

```
module A
CONST = "A's CONST"
def f
  class_eval {
    p CONST #=> "A's CONST"
  }
end
end

class K
CONST = "K's CONST"
extend A
f
end
```

lookup CONST in order
of Constant K,A, Object



Constant Lookup is complex

class_eval(String)

```
01:module A
02: CONST = "A's CONST"
03: def f
04:   class_eval <<-EOS
05:     p CONST #=> "K's CONST"
06:   EOS
07: end
08:end
09:
10:class K
11: CONST = "K's CONST"
12: extend A
13: f
14end
```

class_eval(&Block)

```
module A
CONST = "A's CONST"
def f
  class_eval {
    p CONST #=> "A's CONST"
  }
end
end

class K
CONST = "K's CONST"
extend A
f
end
```

lookup CONST in order
of Constant K,A, Object

lookup CONST in order
of Constant A, Object.
Does not lookup from K.



Managed somehow



A very complex bug

Lexical



A very complex bug

```
01: module A
02:   B = 42
03: end
04:
05: A.class_eval do
06:   def self.f
07:     p B
08:   end
09: end
10:
11: A.f
```

Lexical



A very complex bug

CAUTION

'class_eval' doesn't add receiver(A) to
constant lookup scope.

```
04:  
05: A.class_eval do  
06:   def self.f  
07:     p B  
08:   end  
09: end  
10:  
11: A.f
```

Lexical



A very complex bug



A very complex bug

Ruby 1.9
42



A very complex bug

Ruby 1.9
42

Ruby 1.8.7
-e:8:in `f': uninitialized constant B (NameError)



A very complex bug

A CRuby's Bug

undefined constant B (NameError)



A very complex bug

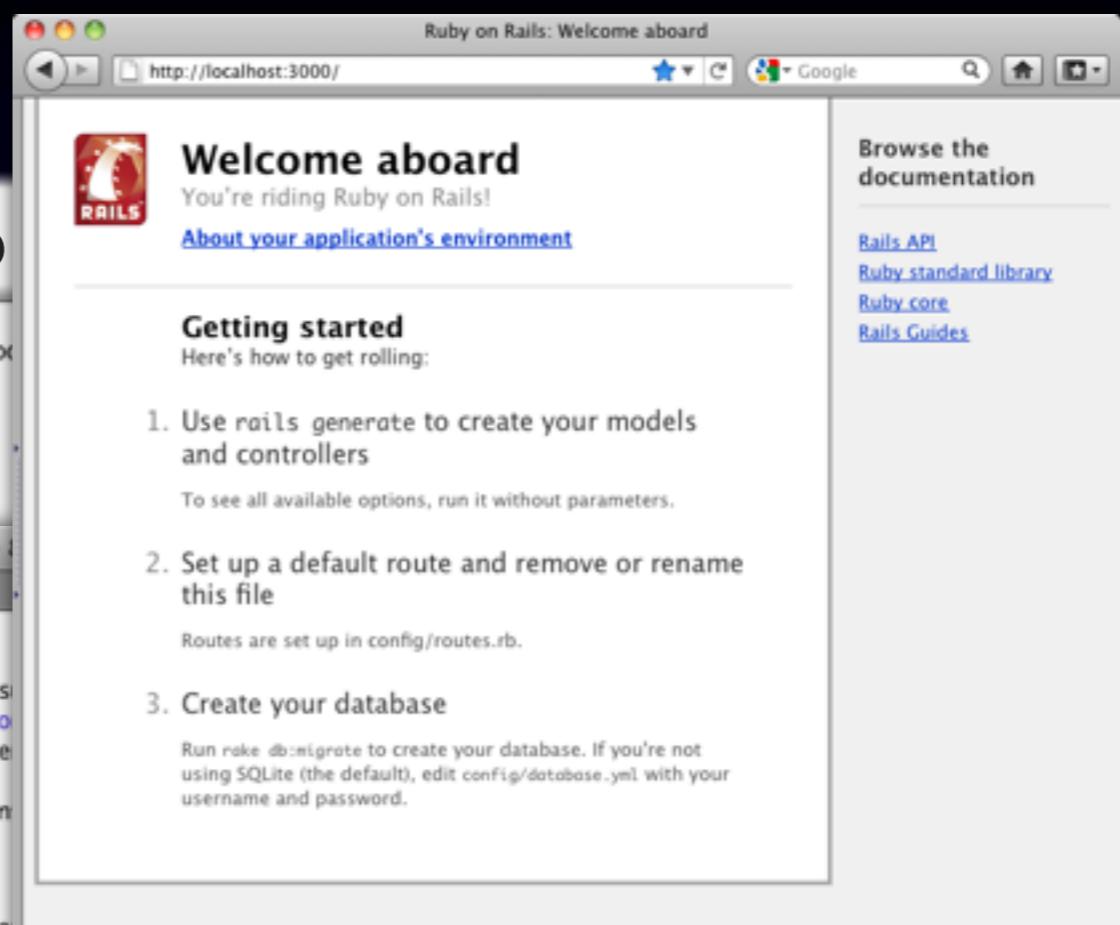
```
01: module A
02:   B = 42
03: end
04:
05: A
06:
07:
08:
09: end
10: A
```

A CRuby's Bug

undefined constant B (NameError)

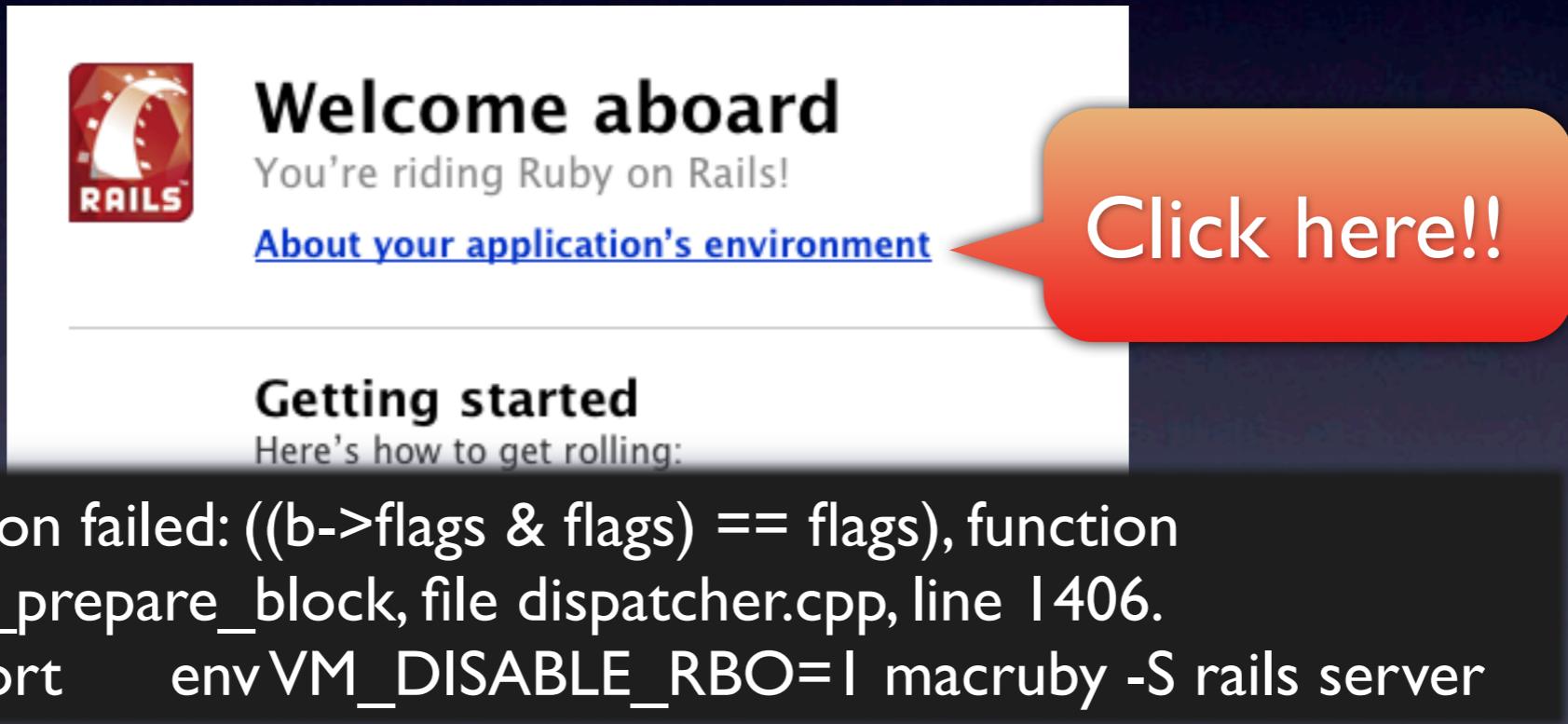


Demo



#I390 rb_vm_prepare_block

→ Abort when I pressed "About your application's environment" in rails 3.0.7 Welcome Page.



The MacRuby train trip continues...



The MacRuby train trip continues...

<http://www.macruby.org/>



The MacRuby train trip continues...

<http://www.macruby.org/>

@macruby



The MacRuby train trip continues...

<http://www.macruby.org/>

@macruby

GitHub:MacRuby/MacRuby



The MacRuby train trip continues...

<http://www.macruby.org/>

@macruby

GitHub:MacRuby/MacRuby

macruby-devel@lists.macosforge.org

Conclusion

- MacRuby
 - MacRuby is a unique blend of Ruby 1.9 and Objective-C.
 - The goal is to be 100% compatible with Ruby 1.9.
- MacRuby on Rails
 - We're now able to show the default Rails welcome page.
 - But there is still a lot to do.

Q & A

- If you have a question, please ask me, but if possible please speak slowly.
- or contact [@takaokouji_en](https://twitter.com/takaokouji_en) by Twitter
- or send e-mail to MacRuby ML

Acknowledgment

- RubyConf team
- Tor
- Everyone