

JRuby

Agile Glue for the Enterprise

Nick Sieger

Sun Microsystems, Inc

nick.sieger@sun.com



Nick

- works on Kenai.com
- works on JRuby
- tweets: twitter.com/nicksieger
- blogs: blog.nicksieger.com

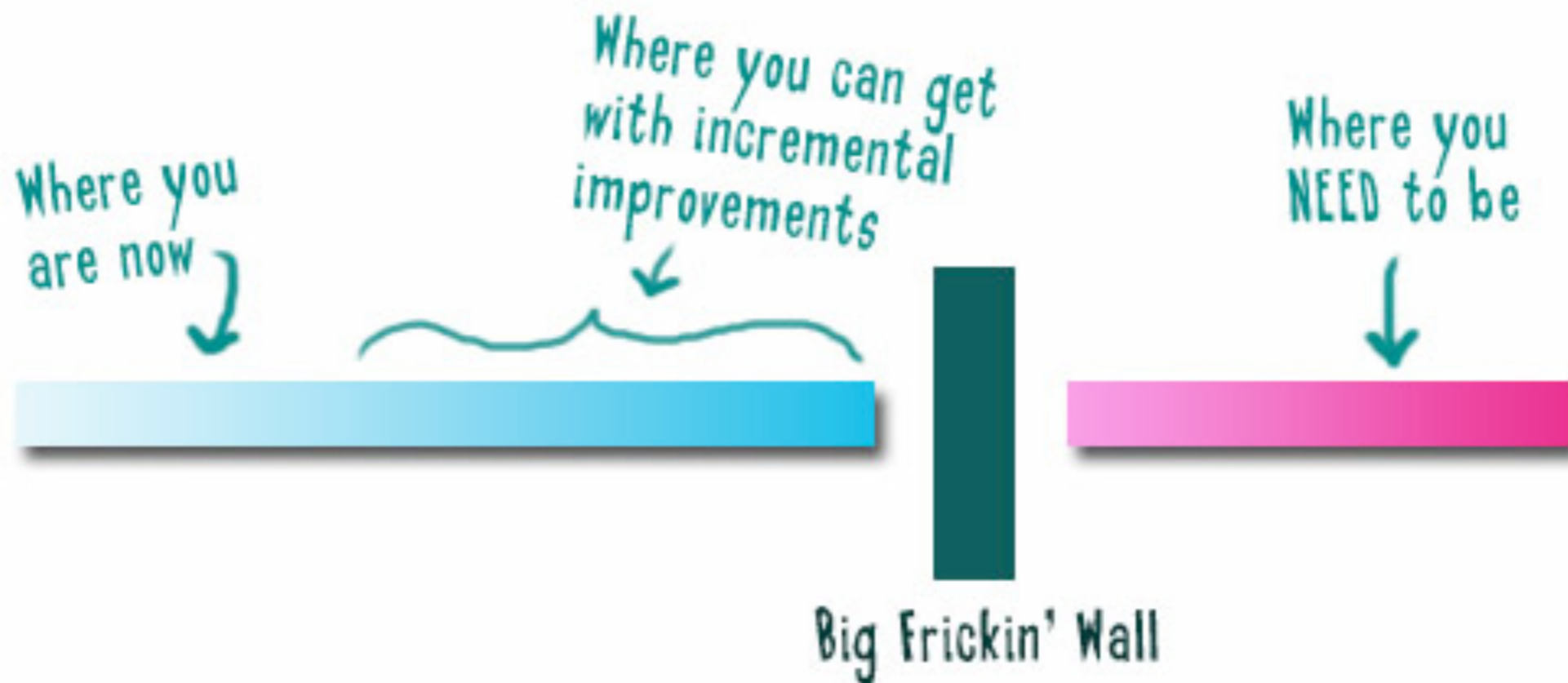


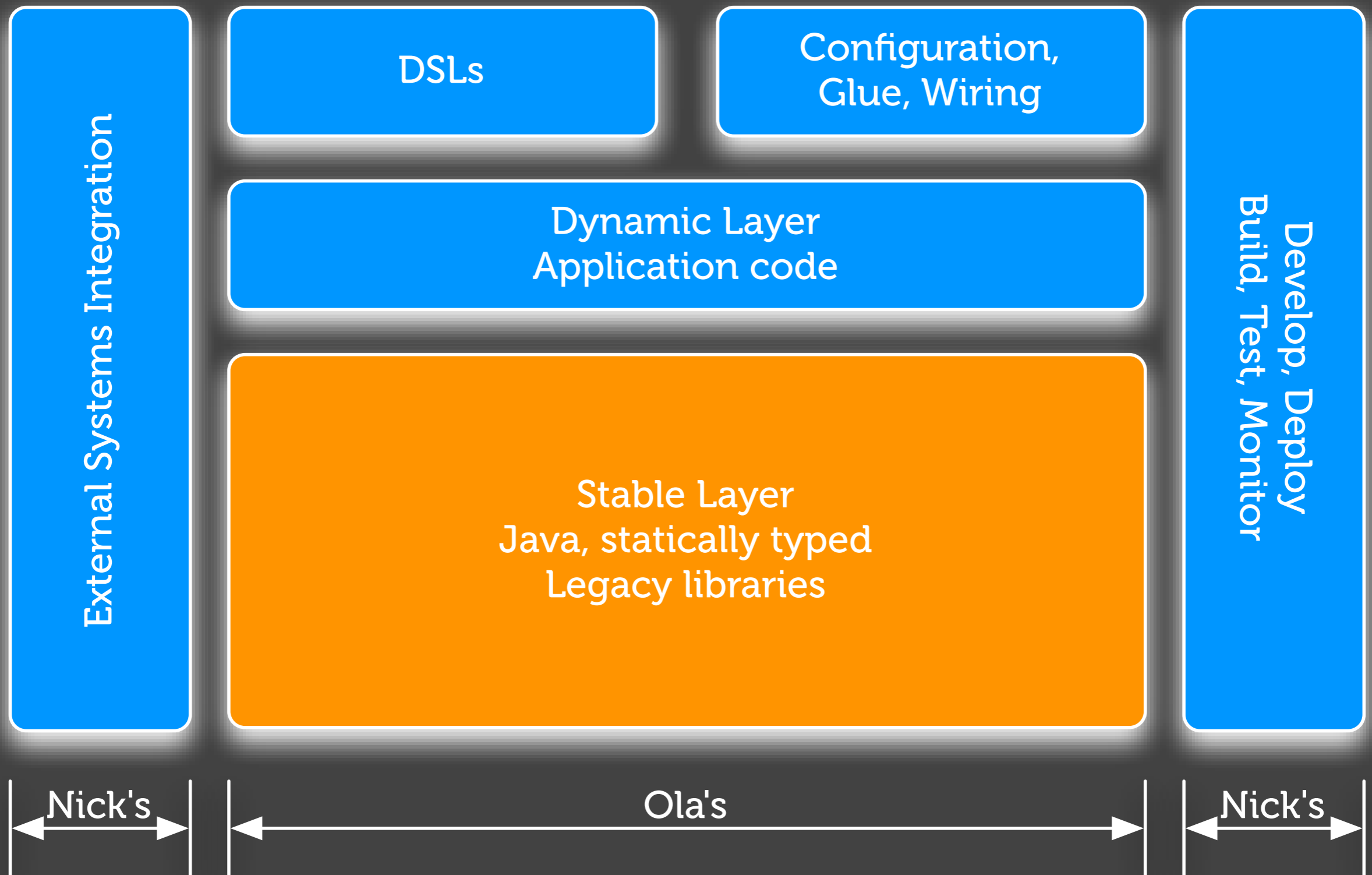
Why Ruby?

Speed



Incremental vs. revolutionary improvements...





Stable	Dynamic
Java	Ruby
Ceremony	Essence
Changes infrequently	Changes frequently
Fewer external dependencies	More external dependencies
Slower development velocity	Faster development velocity
Hardened	Malleable

Ruby

- **dynamic, strongly typed**
- **everything is an object**
- **language-level focus on readability, productivity**

Java

```
package demo;  
  
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Ruby

```
puts "Hello World"
```

Java

```
package demo;

public class Point {
    private int dimension;
    private int[] coordinates;

    public Point(int x, int y) {
        dimension = 2;
        coordinates = {x, y};
    }

    public void setDimension(int newDimension) {
        dimension = newDimension;
    }

    public int getDimension() {
        return dimension;
    }

    public void setCoordinates(int[] newCoordinates) {
        coordinates = newCoordinates;
    }

    public int[] getCoordinates() {
        return coordinates;
    }
}
```

Ruby

```
class Point
  attr_accessor :dimension, :coordinates

  def initialize(*coordinates)
    @dimension = coordinates.length
    @coordinates = coordinates
  end
end
```

Java

```
FileInputStream input = null;
FileOutputStream output = null;
try {
    try {
        input = new FileInputStream("/tmp/src.txt");
    } catch (IOException io1) {
        throw new RuntimeException("couldn't open input file", io1);
    }
    try {
        output = new FileOutputStream("/tmp/dest.txt");
    } catch (IOException io2) {
        throw new RuntimeException("couldn't open output file", io2);
    }

    byte[] buf = new byte[8192];
    int numBytesRead = 0;
    while ((numBytesRead = input.read(buf)) != -1) {
        output.write(buf, 0, numBytesRead);
    }
} catch (IOException io) {
    throw new RuntimeException("couldn't read/write or something", io);
} finally {
    try {
```

Ruby

```
File.open("/tmp/src.txt") do |input|  
  File.open("/tmp/dest.txt", "w") do |output|  
    output << input.read  
  end  
end
```


Ruby: Basics

```
# Numbers
```

```
1, 0xff, 0755, 3.14, 2.2e10
```

```
# Strings
```

```
"One", 'Two', %{Three}
```

```
# Literal arrays and hashes
```

```
array = [1, 2, 3]
```

```
hash = { 1 => "One", 2 => "Two", 3 => "Three" }
```

```
# Regular Expressions
```

```
regexp = /(One|Two|Three)/
```

```
# Ranges
```

```
inclusive, exclusive = (1..100), (1...100)
```

```
# Conditionals
```

```
if count > 10
```

```
  puts "Try again"
```

```
elsif tries == 3
```

```
  puts "You lose"
```

```
else
```

```
  puts "Enter a number"
```

```
end
```

```
# Case/Switch
```

```
case grade
```

```
when 90..100: 'A'
```

```
when 80..90 : 'B'
```

```
when 70..80 : 'C'
```

```
when 60..70 : 'D'
```

```
when Integer: 'F'
```

```
when /['A-D]/, /[F]/ : grade
```

```
else
```

```
  raise "Not a grade: #{grade}"
```

```
end
```

Ruby: Objects

```
point = Point.new(1, 2)
puts point.coordinates # => [1, 2]
puts point.coordinates() # same; parens optional
puts point.class # => Point
puts Point.class # => Class
Point.instance_methods - Point.superclass.instance_methods
# => ["dimension", "dimension=", "coordinates", "coordinates="]
```

Ruby: Open Classes

```
class Fixnum
  def even?
    self % 2 == 0
  end
end
```

```
3.even? # => false
```

```
class NilClass
  def empty?
    true
  end
end
```

```
"hi".empty? # => false
"".empty?   # => true
nil.empty?  # => true
```

Ruby: Blocks

```
File.open("500-byte-file", "w") do |f|  
  f << 'x' * 500  
end
```

```
Thread.new do  
  some_background_operation  
end
```

```
transaction do  
  database.execute "update users set password = password(' ')"  
end
```


Ruby: DSLs

```
class Project < ActiveRecord::Base
  belongs_to :portfolio
  has_one :project_manager
  has_many :milestones
end
```

Ruby: ...

(Rest left as an exercise to the viewer)

JRuby:
JVM + Ruby = ♥

JRuby

- **Current Release: 1.2.0 (Mar 16, 2009)**
- **Fastest Ruby 1.8-compatible runtime**
- **Native-threaded**
- **Easy access to Java libraries**

Hello JRuby

```
$ jirb  
irb(main):001:0> java.lang.System.out.println "Hello JIRB"  
Hello JIRB  
=> nil
```

IRB

(Interactive RuBy)

Hello Swing

```
require 'java'  
%w(Frame Button).each {|c| import "javax.swing.J#{c}" }
```

```
frame = JFrame.new("Hello Swing")  
frame.set_size 200, 200  
frame.add(button = JButton.new("Press me"))
```

```
count = 0  
button.add_action_listener do |event|  
  count += 1  
  event.source.text = "Pressed #{count} time#{ count > 1 ? 's' : ''}"  
end
```

```
frame.show
```

Java Integration (JI)

JI: Ruby-like Methods

Java

```
Locale.US  
System.currentTimeMillis()  
locale.getLanguage()  
date.getTime()  
date.setTime(0)  
file.isDirectory()
```



Ruby

```
Locale::US  
System.current_time_millis  
locale.language  
date.time  
date.time = 0  
file.directory?
```

JI: Ruby

Conversions

- **primitives: numbers, strings, booleans, nil**
- **collections: arrays->lists, hashes->maps**
- **attempt “principle of least surprise”**

JI: Ruby Conversions

```
ruby_array = [10, 5, 1]
java.util.Collections.sort(ruby_array)
ruby_array
# => [1, 5, 10]
```

JI: Java Extensions

```
h = java.util.HashMap.new
h["key"] = "value"
h["key"]
  # => "value"
h.get("key")
  # => "value"
h.each {|k,v| puts k + ' => ' + v}
  # key => value
h.to_a
  # => [["key", "value"]]
```


JI: Java Extensions

```
module java::util::Map
  include Enumerable

  def each(&block)
    entrySet.each { |pair| block.call([pair.key, pair.value]) }
  end

  def [](key)
    get(key)
  end

  def []=(key, val)
    put(key, val)
    val
  end
end
```

JI: Helpers

```
["a", "b", "c"].to_java :string  
# => new String[] {"a", "b", "c"}
```

```
import java.io.FileOutputStream  
stream = FileOutputStream.new("/tmp/hello.txt")  
stream.write("Hello".to_java_bytes)  
stream.close
```

JI: Interface Conversion

```
package java.util.concurrent;
public class Executors {
    // ...
    public static Callable callable(Runnable r) {
        // ...
    }
}
```

JI: Interface Conversion

```
class SimpleRubyObject  
end
```

```
import java.util.concurrent.Executors  
callable = Executors.callable(SimpleRubyObject.new)  
callable.call
```

```
# => undefined method `run' for #<SimpleRubyObject:0xfd5428>  
(NoMethodError)
```

```
class SimpleRubyObject  
  def run  
    puts "hi"  
  end  
end  
callable.call  
# => hi
```

JI: Closure Conversion

- taking that last example further...

```
import java.util.concurrent.Executors
callable = Executors.callable { puts "hi" }
callable.call
# => hi
```

```
button.addActionListener do |event|
  event.source.text = "Pressed!"
end
```

JI: Much more...

**How can I bring
JRuby into my org?**

Testing

Testing frameworks

- Test::Unit (JUnit analogue)
- RSpec (BDD in Ruby)
- Cucumber (BDD in plain text)
- *Many, many supporting tools*

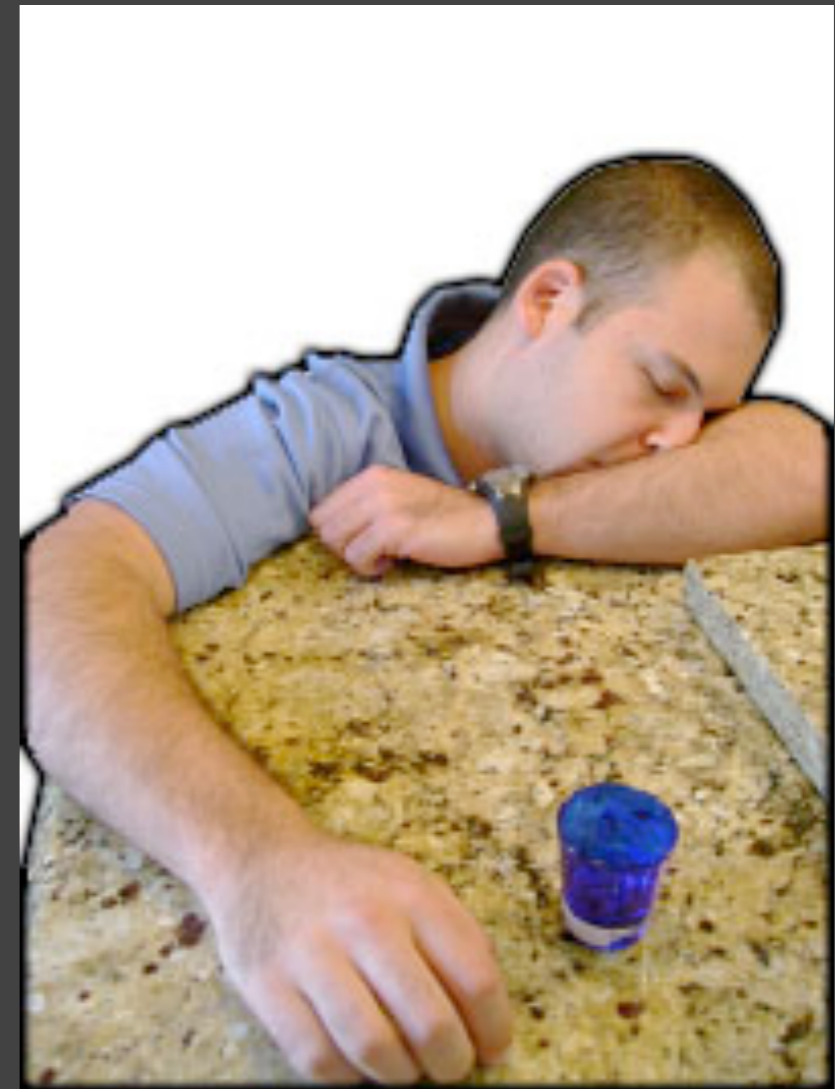
Builds

Rake

```
desc "This task will purchase your Vodka"  
task :purchaseAlcohol do  
  puts "Purchased Vodka"  
end
```

```
desc "This task will mix a good cocktail"  
task :mixDrink => :purchaseAlcohol do  
  puts "Mixed Fuzzy Navel"  
end
```

```
desc "This task will drink one too many"  
task :getSmashed => :mixDrink do  
  puts "Dood, everthing's blurry, can I halff noth'r drinnnk?"  
end
```



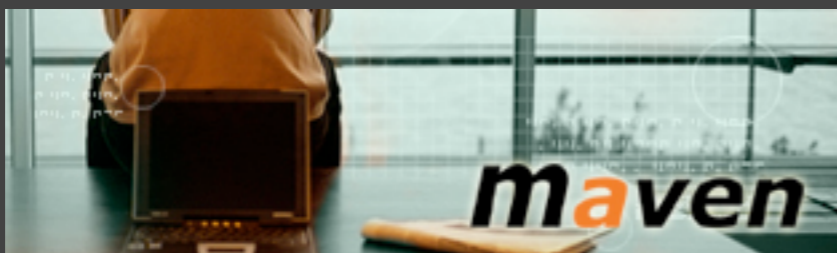
Rake

```
require 'rake/clean'
CLEAN << "build"

directory "build"

desc "Compile the code"
task "compile" => "build" do |t|
  files = FileList["src/**/*.java"]
  sh "javac -d #{t.prerequisites.first} #{files}"
end

desc "Create a jar file of the compiled code"
task "jar" => "compile" do
  sh "jar cf project.jar -C build ."
end
```



Maven

```
<plugin>
  <groupId>org.jruby.plugins</groupId>
  <artifactId>jruby-rake-plugin</artifactId>
  <version>1.2.0</version>
  <executions>
    <execution>
      <id>tests</id>
      <phase>test</phase>
      <goals><goal>rake</goal></goals>
      <configuration>
        <args>default</args>
      </configuration>
    </execution>
  </executions>
</plugin>
```

Route around Maven with Rake!

JMX

```
$ jruby -S gem install jmx
```

```
require 'rubygems'  
require 'jmx'
```

```
server = JMX.connect(:host => "localhost", :port => 8686,  
  :user => "...", :password => "...")  
memory_bean = server["java.lang:type=Memory"]  
puts memory_bean.heap_memory_usage.used
```

Application

Custom Embedding

JSR-223

```
// import javax.script.*;  
  
ScriptEngineManager factory = new ScriptEngineManager();  
ScriptEngine engine = factory.getEngineByName("jruby");  
engine.put("name", "ETE");  
engine.eval("puts 'Hello ' + $name");
```

<http://java.sun.com/developer/technicalArticles/scripting/jruby/>

JRuby's API

```
// import org.jruby.Ruby;
// import org.jruby.javasupport.JavaEmbedUtils;
// import org.jruby.runtime.builtin.IRubyObject;

Ruby runtime = JavaEmbedUtils.initialize(new ArrayList());
IRubyObject name = JavaEmbedUtils.javaToRuby(runtime, "ETE");
runtime.getGlobalVariables().set("$name", name);
runtime.evalScriptlet("puts 'Hello ' + $name");
```

More bare metal, but subject to change

App Startup

```
public class StartupScriptLauncher implements ServletContextListener {
    public void contextInitialized(ServletContextEvent event) {
        ServletContext servletContext = event.getServletContext();
        try {
            Ruby runtime = JavaEmbedUtils.initialize(Arrays.asList(new String[] {
                // set up any $LOAD_PATH items here
                servletContext.getRealPath("/WEB-INF"),
                servletContext.getRealPath("/WEB-INF/lib")
            }));
            // setup any global variables
            runtime.getGlobalVariables().set("$servlet_context",
                JavaEmbedUtils.javaToRuby(runtime, servletContext));
            // require e.g., WEB-INF/config/startup.rb
            runtime.getLoadService().require("config/startup");
        } catch (Exception ex) {
            servletContext.log(ex.getMessage());
        }
    }
}
```

Full version @ <http://gist.github.com/23909>



Rails

- Full-stack MVC web framework
- "Opinionated software"
- Convention over configuration
- Rich community, plugins, extensions...
- Runs great on JRuby!



David Heinemeier Hansson,
Rails creator

<http://flickr.com/photos/x180/2537416657/>

Rails: what can we use?

- **ActionPack: controllers/routing (C)**
- **ActionView: templates (V)**
- **ActiveRecord: ORM (M)**

ActionPack

```
class ProductsController < ActionController::Base
  # GET /products
  def index
    @products = product_manager.findAllProducts
  end

  private
  def product_manager
    ProductManagerFactory.getProductManager
  end
end
```

ActionView

```
<%# index.html.erb -%>
<dl>
  <% @products.each do |product| -%>
    <dt><%= h @product.name %></dt>
    <dd>Published:&nbsp;
      <%= time_ago_in_words(@product.publish_time) %><br/>
      <%= h @product.description %></dd>
  <% end -%>
</dl>
```

ActiveRecord

```
class Project < ActiveRecord::Base
  belongs_to :portfolio
  has_one    :project_manager
  has_many   :milestones
end
```


ActiveRecord: MySQL Test DB

```
class Employee < ActiveRecord::Base
  set_primary_key "emp_no"
  has_one :title, :foreign_key => "emp_no"
  has_and_belongs_to_many :departments,
    :join_table => "dept_emp",
    :foreign_key => "dept_no",
    :association_foreign_key => "emp_no"
end
```

<https://edge.launchpad.net/test-db>

Warbler

Package an entire
Rails application as
a WAR file

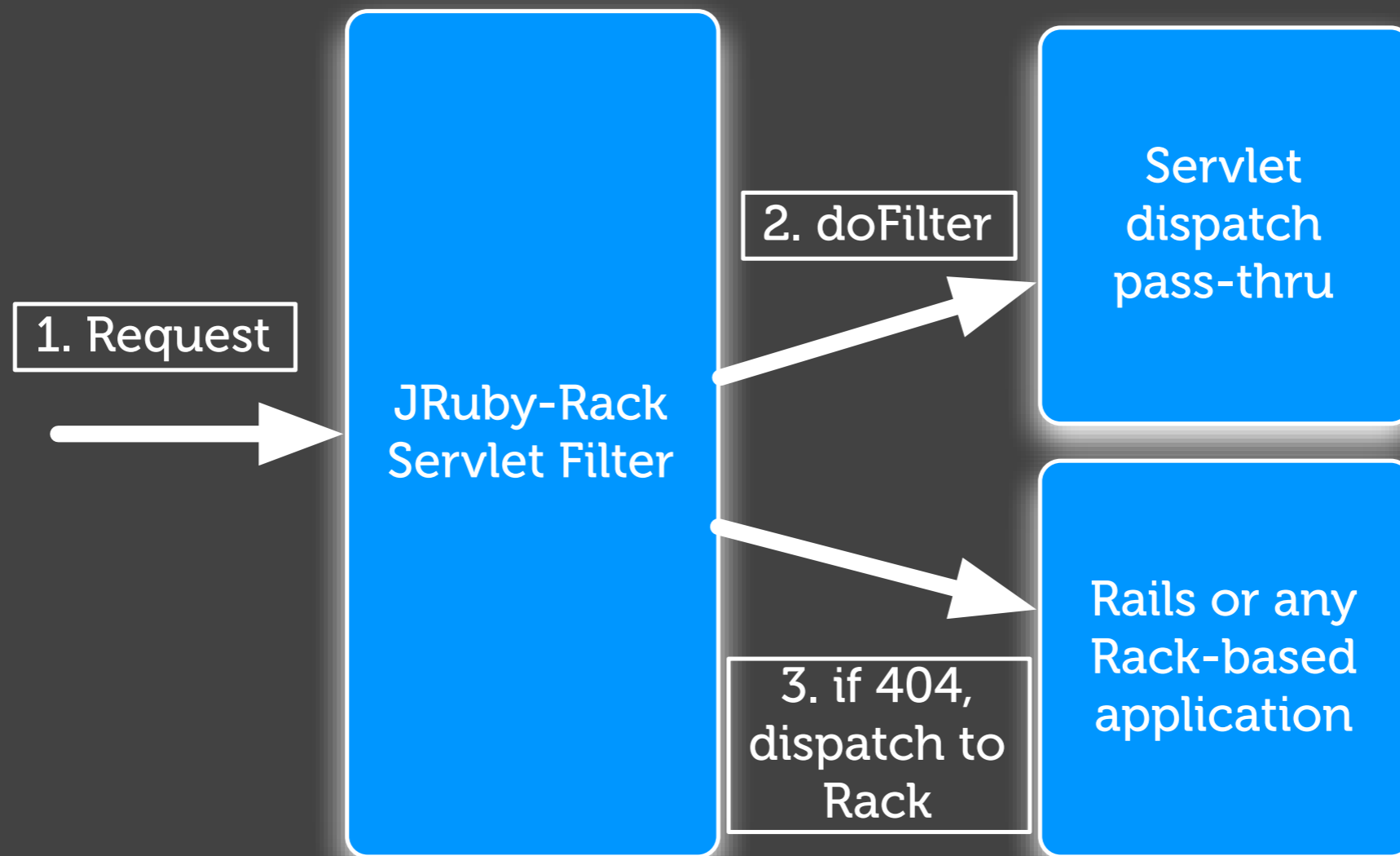


<http://warbler.kenai.com/>

Warbler

```
jruby -S gem install warbler
jruby -S warble config
    # => config/warble.rb
jruby -S warble
    # => myapp.war
```

JRuby-Rack



ActionPack with JSPs

```
class ProductsController < ActionController::Base
  # GET /products
  def index
    products = product_manager.findAllProducts
    servlet_request["products"] = products
    forward_to "WEB-INF/views/show_products.jsp"
  end

  private
  def product_manager
    ProductManagerFactory.getProductManager
  end
end
```

Recapitulation

- Use JRuby for...
 - Testing
 - Builds
 - Monitoring
 - Embedded scripts
 - Partial or full application stack

**What are you
waiting for?**

Resources

- Slides: <http://blog.nicksieger.com/files/jruby-ete.pdf>
- <http://www.jruby.org/>
- <http://wiki.jruby.org/>
- <http://jruby-rack.kenai.com/pages/Home>
- <http://warbler.kenai.com/sources/main/show>
- <http://github.com/nicksieger/advent-jruby/tree/master>

