

The Good, The Bad & The Ugly

(Clojure & JRuby)

Allen Rohner
@arohner
@circleci
Clojure/West 2012

Who Am I

- Using Clojure in production since 2008
- Author of Scriptjure, a DSL for writing javascript, two years before ClojureScript
- Commits in Clojure core, contrib, ring, compojure, noir, lein, swank-clojure, pallet...

Circle

- Founder
- CircleCI.com
- Easy Continuous Integration for web apps

When this talk was
accepted....



TL;DR

It Is NOT about:

- “Ruby sucks”
- “JRuby sucks”

It Is NOT about:

- ~~“Ruby sucks”~~
- ~~“JRuby sucks”~~

It IS about

Ruby != Clojure

My Biases

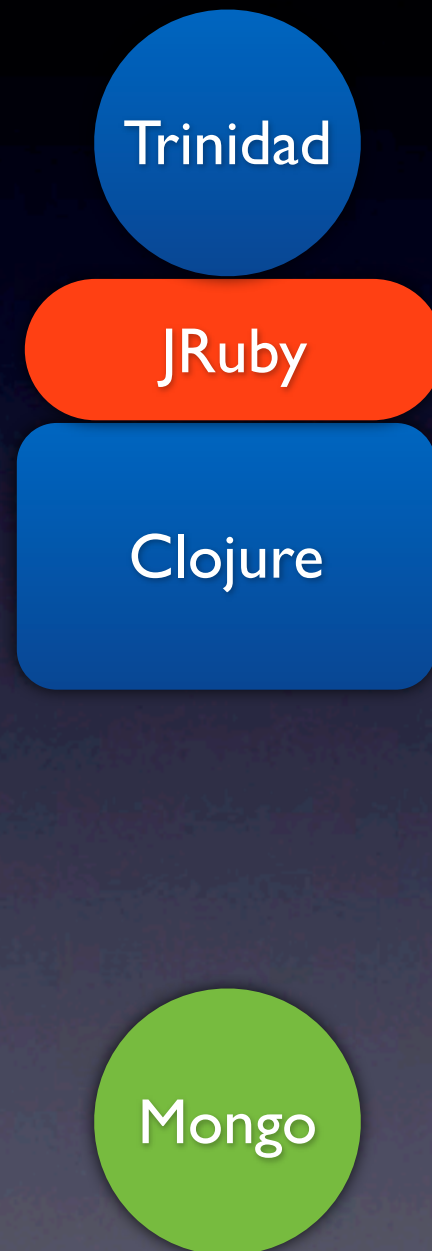
- 2006-2009: Rails 1.x
- 2008-2012: Clojure
- Expert in Clojure, Intermediate in Ruby

My History



How did we get into
this mess?

Architecture





The Good

Calling just works

```
(require 'foo.bar)  
(foo.bar/baz 3)
```

```
foo = JRCLj.new("foo.bar")  
foo.baz(3)
```

```
(-> (r/module "Foo")  
    (r/class "Bar")  
    (r/send :baz 3))
```

```
Foo::Bar.baz(3)
```


Fork Me

- <https://github.com/kyleburton/jrclj>
- <https://github.com/circleci/cljr>

The Clojure REPL
makes Ruby better

Individual tests

- (r/rspec “specs/controllers/foo.rb”)
- (r/rspec “-e” “that one test”)*

Fine-grained reloading

- (r/eval “load ./foo.rb”)

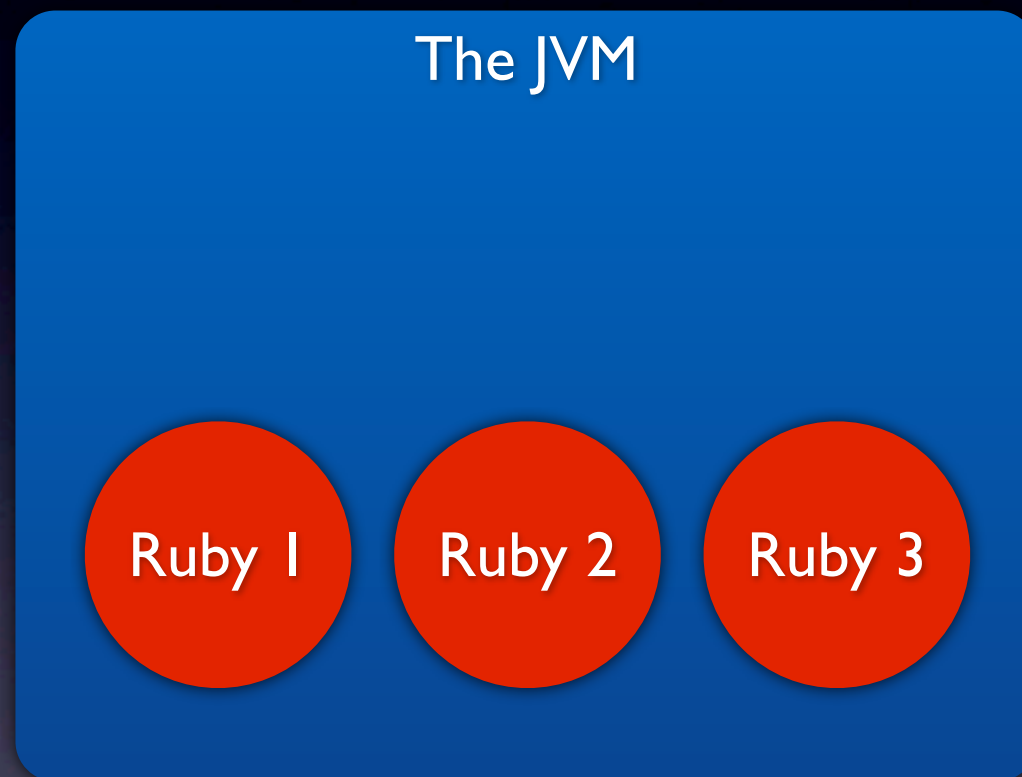
Clojure Concurrency from JRuby

- `Worker.send_email()`
- concurrency with no queues

Runtimes!

- (r/new-runtime)
- (def server (r/rails-server))

JRuby Runtimes



Combining WARs

- Warbler
- lein war
- Deploy both on tomcat/jetty
- (Not tested)

JRuby

- Impressive technology
- awesome maintainers



The Bad

But...

- Stuck on 1.6.5, couldn't upgrade or downgrade due to different bugs

1.9

- 1.9 support not finished
- Encoding
- SSL

Complexity

kLOC	Java	Self-hosted
Clojure (1.3.0)	51	28
JRuby (1.6.5)	291	600

Gems

- Often not tested against JRuby
- ExecJS
- Some just can't work in JRuby
- FFI

Slow Startup

- Ruby likes to restart, early & often
- helpers, rack middleware

Slow Startup

- 1:03 macbook pro
- 1:45 macbook air
- 3:00 ec2 m1.small

Slow Startup Matters

- need to RSpec all the time
- Spork & Guard didn't work
- rake tasks (generate, etc)
- rake assets:precompile (8m)

Dilemma:

- slow restart
- strangeness caused by Ruby code not expecting to reload
- Restart often, just in case

Setup

- Paul lost 2 days

CRuby in Dev, JRuby in production

Doesn't work!

Different Views

- Clojure likes one big process, lots of threads
- Ruby prefers lots of small processes
- Multithreaded JRubyOnRails: works in theory, not in practice. Weird classloading bugs, unsupported gems.

Duplication

- DB connections
- DB models
- Test data

Classpath

- Different depending on whether you ran ‘rails console’ or ‘lein repl’
- => different behavior
- Can be fixed, but it's one more headache
- If you didn't understand that, yes

Rails Env

- DB connection
- Email policy
- Class reloading
- error reporting
- Airbrake

Clojure Environment

- Clojure leaves that to you
- * (yes there's Noir's env, but it's minor compared to what Rails does)

Testing

- Rails loves to mock
- Clojure prefers pure fns

Rails Testing Env

- FactoryGirl returning fake objects
- DB Cleared between tests
- But who told Clojure?
- Are you both talking to the same DB?



The Ugly

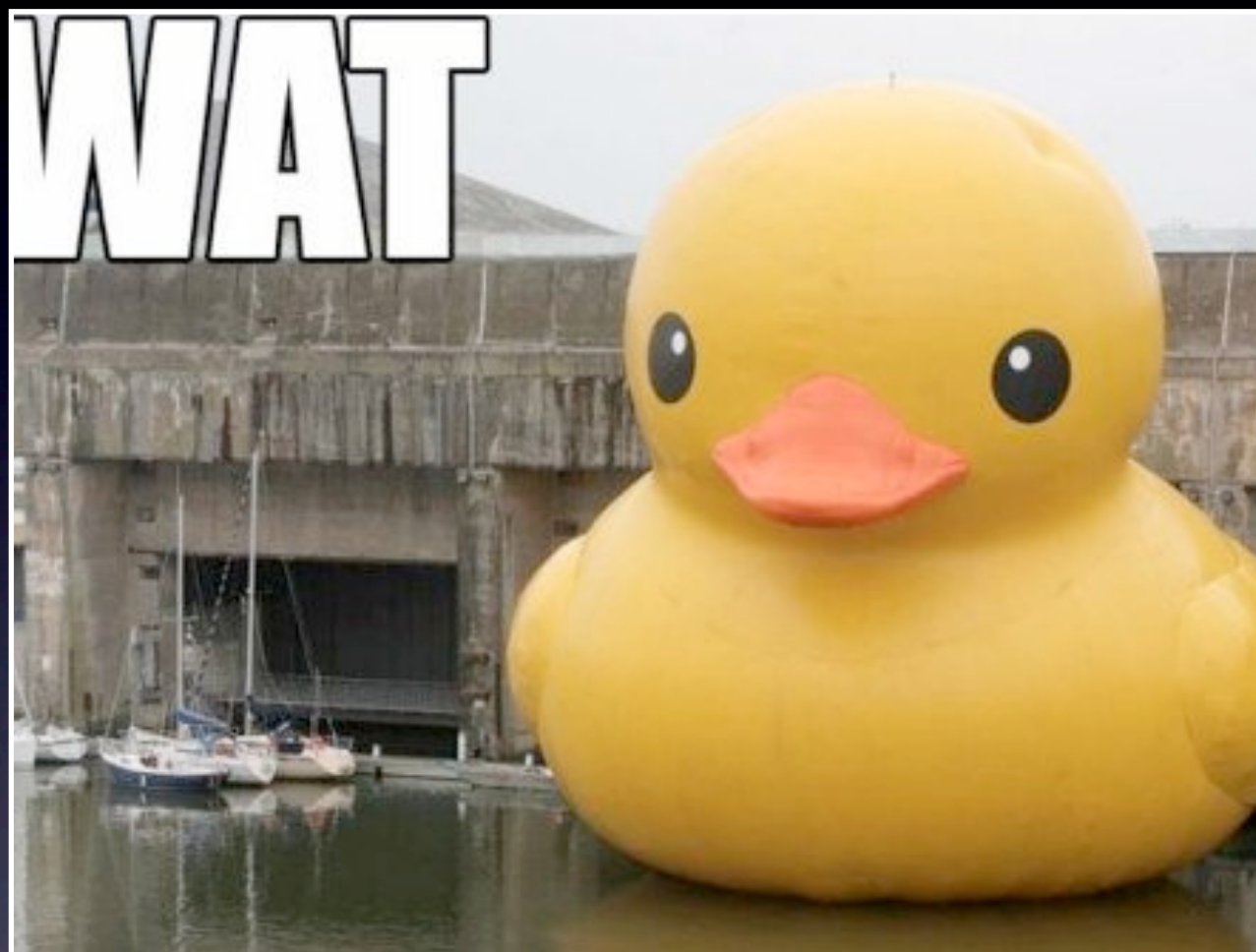
Naming

- `project.git-url =>`
`project["git-url"]`
- `(mongo/update! :project`
`{:giturl git-url}))`

Symbols vs. Keywords

- Wrappers
- Clojure :foo != Ruby :foo

- Rails code, calls Clojure
- Clojure calls back into Rails
- But might not be the same runtime
- might not have the same classpath
- might not be talking to the same DB
- might not be the same ruby version



Lessons Learned

Complexity is Multiplicative

Rails Magic

The closer you are to the 'standard' environment, the better things work

- Rails 3 (ok)
- using Mongo/Mongoid (not bad)
- and JRuby (hrm...)
- and Clojure (OMGWTF)

Avoid Mixing Paradigms

(In the same process)

Don't duplicate Models

Don't overestimate the
value of code reuse

Understanding Devise + OmniAuth

>

Writing OAuth2 login in Clojure

Yes, you will write
more code

But you'll understand it!

Yes, you will read Clojure source

But it's simpler!

The Future

- Clojure REST API
- Fat Javascript Client
- Knockout JS
- Sammy JS
- HamlCoffee
- Clojurescript?

- <https://github.com/circleci/mongofinil>
- <https://github.com/edgecase/dieter>

Questions?

@circleci
circleci.com