

# Why JRuby?

**Fiona Tay**

@msfionatay

Engineer, Pivotal Labs

**This is for the  
person who just  
wants to make web  
apps.**

**This is about the  
Good, The Bad, and  
The Ugly.**

# I'm a Rubyist



Ruby  
*A Programmer's Best Friend*

# I'm also a JRubyist



<http://github.com/GreenplumChorus/chorus>

# An Introduction

# It's a Ruby implementation

1. Install it

2. Write Ruby code

# It's a Ruby implementation

```
$ rvm use jruby-1.7.0  
$ irb  
> 1+1  
2
```



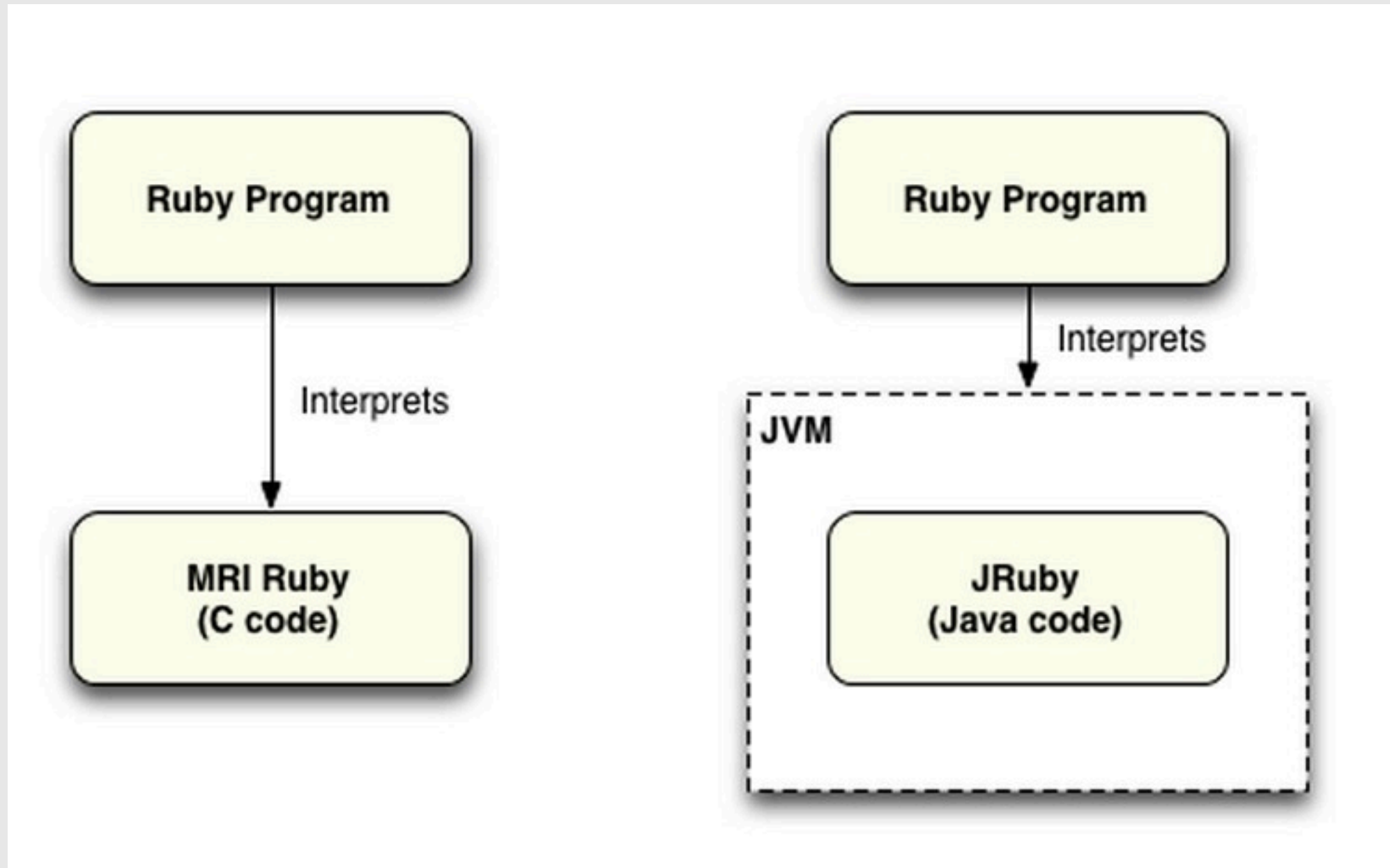
# Ruby implementations

Interchangeable to an extent

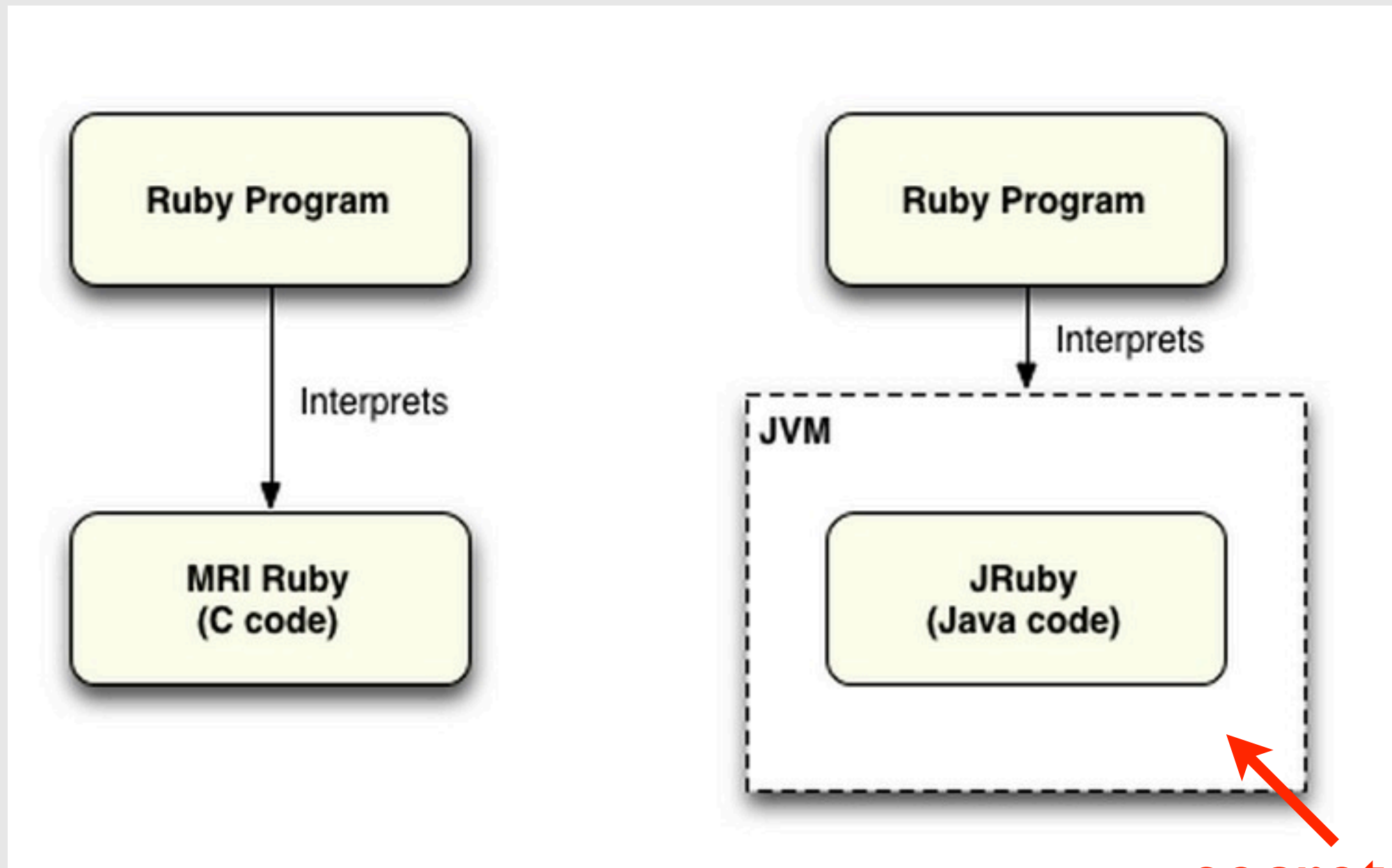
Popular implementations:

- MRI
- JRuby
- Rubinius
- MacRuby

# Under the hood



# Under the hood



secret sauce

# Interchangeability

**1+1 should be 2 in all implementations**

# Implementation differences

Some implementations are more equal than others

Differences:

- Speed of execution
- Support for libraries
- Implementation bugs

# History of JRuby

Around since 2002

Has had traction in real projects since ~2005

# JRuby in the real world

**ThoughtWorks®**

# JRuby in the real world





# The Good

# Access to Java libraries

**Incredibly powerful tool**

**Everyone and his/her dog writes Java**

# Java libraries: NLP

**Stanford Parser is the NLP library of choice**

# Java libraries: Email

**JavaMail is a better alternative to ActionMailer**

# Access to Java libraries

JRuby has deep integration with Java  
Directly call Java code from Ruby

# Directly calling Java code

```
require 'java'

frame = javax.swing.JFrame.new
frame.getContentPane.add
javax.swing.JLabel.new('Hello, World!')
frame.set_visible true
```

# Using other languages

## Other languages built on the JVM

- Scala – special support
- Clojure
- Groovy

# Using Ruby gems

If you're lucky, someone has made a Ruby gem that wraps the Java library

Also wrapping a gem is a good idea



# Easier app deployment

Not easier than `git push heroku master`

Only relevant if you're deploying to large server system

# Easier app deployment

Leverage existing Java-based deployment infrastructure

# Leverage infrastructure

**Warbler packages everything into a single WAR file**  
**Standard file format for Java deploys**

# Other deployment options

**TorqueBox provides additional services**

# Better performance

**Better compiler than MRI**

**Benefits from improvements to JVM**

# Better performance

Access to highly optimized Java libraries

# The Bad

# Gem availability





# Gem availability

Gems written in pure Ruby work with JRuby

Gems with C extensions don't work with JRuby

Typically, these are older, less-maintained gems

# Fixing gems

Sometimes, you can fix the existing gem  
Could be as simple as replacing its dependencies  
Or could be complex

# Fixing gems

Patched a job queueing system

Took more work than we expected

YMMV

# Long app startup time

Inhibits test-driven development

Due to Rails startup time

Difficult to set up Spork with JRuby

# Long app startup time

C'est la vie

# Less mature technology

**MRI still the default**

**Though JRuby has matured**

**Libraries developed against MRI**

# The Ugly

# Concurrency

**“Concurrency is when several computations are executing simultaneously, and potentially interacting with each other.” – Wikipedia**



# Concurrency

JRuby has real threads

MRI Ruby has threads with Global Interpreter Lock

# Concurrency and Ruby

Threads are a major innovation

But MRI has never really had a focus on concurrency

# Concurrency is great

Cheaper to scale

Good multi-threaded web servers

# More bang for your buck

One server: MRI runs two Ruby processes

One server: JRuby runs two Ruby processes with 4 threads each

Can use JRuby on Heroku

# Concurrency is not free

## Bugs in non-thread safe libraries

- Rails
- Airbrake

Libraries developed without a focus on concurrency

# Concurrency is not free

Writing thread-safe code is difficult

Even for Java developers with experience

But concurrency is the new sexy

# Concurrency is worth it

Huge performance improvements

Some features would be much easier with threads

Threads is the future

# The Decision



**JRuby is a non-  
standard choice.**

# Should you use JRuby?

Do you want to use Java libraries?

# Should you use JRuby?

**Do you want to leverage Java-based deployment infrastructure?**

# Should you use JRuby?

Do you need good performance?

# Should you use JRuby?

Otherwise, no



# Should you use JRuby?

**Not to worry – you can always switch to JRuby later**  
**Lots of people switch for performance reasons**

# Getting started with JRuby

**jruby-lint** checks how ready your code is to run on JRuby

# Acknowledgments

**Pivotal Labs**

**JRuby team**



# Thanks!

@MsFionaTay