



# Rubinius

Entwicklungsprozesse in Open-Source-Projekten  
Hasso-Plattner-Institut  
09. Januar 2009



- Einführung in Rubinius
  - Was ist so schlecht an MatzRuby?
  - Ein Überblick über Ruby Implementierungen
  - Die Lösungsansätze von Rubinius
  - Status Quo
- Projektstruktur
  - Die Community
  - Behavior Driven Development
  - Meine Beteiligung



**MRI**

**=**

**MatzRuby**

**=**

**CRuby**

**=**

**Offizielle Ruby-Implementierung von  
Yukihiro „Matz“ Matsumoto**



## Allgemein:

- Schlechte Performance
- Schlechter Garbage Collector
- Speicherprobleme
- Sprache schwer erweiterbar
- Nicht Thread-Safe
- Feature XYZ fehlt

## MRI 1.8:

- Keine VM
- Nur ein nativer Thread
- Kein Unicode

## Behobene Probleme:

- Keine Spezifikation
- Kaum Tests



XRuby

Ruby.Net

**JRuby**

IronRuby


MacRuby

**MatzRuby**

MagLev

Cardinal

**Rubinius**




„MRI ist für C-Programmierer.  
JRuby ist für Java-Programmierer.  
IronRuby ist für CLI-Programmierer.  
Rubinius ist für Ruby-  
Programmierer.“

- Evan Pheonix



- Alles außer der VM ist in Ruby geschrieben
- Implementierung an Smalltalk-80 angelehnt
- Sauberer, gut lesbarer Code
- Solider, stabiler Coder (geprüft mit Valgrind)
- Generational GC (austauschbar)
- Thread Safe (mehrere VMs pro Prozess)
- Features von Erlang und Smalltalk
- Code Optimizer, LLVM Integration, Non-Blocking IO, Keinen C-Stack, ...

- 
- Kompatibel zu MRI 1.8
  - Zur Zeit langsamer Kaltstart
  - Rubinius führt Ruby am schnellsten aus, aber: JRuby implementiert Kernel in Java, MRI in C, ...
  - Sehr gut bei Micro-Benchmarks
  - Nicht so gut bei Macro-Benchmarks





## Drei Arten C einzubinden:

- Primitives:
  - Werden von der VM bereit gestellt
  - Werden vom Compiler erkannt
- Rubinius FFI:
  - Native Methoden und Structs ansprechen von Ruby aus
  - Wird jetzt auch von JRuby und via ruby-ffi von MRI unterstützt
- Subtent:
  - Täuscht MRI vor, C-Erweiterungen laufen unverändert
  - Wenn Erweiterung abstürzt (etwa durch Segmentation Fault), dann stürzt Rubinius nicht mit ab



- **Website:**  
<http://rubini.us>
- **Versionsverwaltung:**  
<http://github.com/evanphx/rubinius>
- **Bug-Tracker:**  
<http://rubinius.lighthouseapp.com>
- **Mailinglist:**  
<http://groups.google.com/group/rubinius-dev>
- **IRC:**  
#rubinius und #rubyspec auf freenode.net



- Vereinigt Spezifikation mit Tests
- RubySpec Projekt war erst Teil von Rubinius
- Heute hauptsächlich von Rubinius- und JRuby-Entwicklern entwickelt
- Wird von allen Implementierungen außer Ruby.Net und Cardinal verwendet und anerkannt
- MSpec zum Ausführen der Tests und Anzeigen der Spezifikation



Folgende Patches habe ich eingeschickt:

- Fixnum:
  - + raises a TypeError when given a non-Integer
  - \*\* returns Infinity for  $0^{**-1}$
  - == calls 'other == self' if the given argument is not a Fixnum
- Bignum:
  - div returns a result of integer division of self by a float argument
  - div raises FloatDomainError if the argument is a float zero
- Singleton:
  - `_instantiate?` returns nil until it is instantiated
  - `_instantiate?` is private

Danach: Commit-Rechte, seit dem 26 Commits



- **Rubinius**

Evan Phoenix, RubyConf 2008

<http://rubyconf2008.confreaks.com/rubinius.html>

- **Strengthening the Ruby Ecosystem Part I: Rubinius**

Evan Phoenix, MountainWest RubyConf 2008

<http://mwrc2008.confreaks.com/01phoenix.html>

- **Merb, Rubinius and the Engine Yard Stack**

Ezra Zygmuntowicz, Google Tech Talk, 20.10.2008

<http://www.youtube.com/watch?v=TcMklv40YMY>

- **Introducing BDD**

Dan North, Better Software magazine, März 2006

<http://dannorth.net/introducing-bdds>