

Building
maintainable
command-line
tools with MRuby

MRuby & mruby-cti

Eric Hodel — @drbrain

About Me

- Ruby committer
- RubyGems committer
- Rake committer
- Author of many gems
- On vacation





PRODUCTS

CUSTOMERS

PRICING

Support Contact Login



SIGN UP

Customers are moving to Fastly. Find out why.

WATCH THE VIDEO ▶



WE'RE REDEFINING REAL-TIME CONTENT DELIVERY. TRY US NOW

Fastly + Ruby

www.ruby-lang.org

rubygems.org

@FastlyJapan

Part I

Web API Wrapper

Why not Ruby?

- Install Ruby
- Installing gems
- Expensive support

MRuby $\approx$ Ruby

1. Prototype in CRuby
2. Port to MRuby
3. Maintain in MRuby

MRuby

MRuby

- Lightweight ruby
- Embeddable
 - Hardware
 - Software

mrbgems

- CRuby stdlib parts
- mruby libraries
- C extensions

Other mruby Talks

2016: Welcome to haconywa

2016: Recent Advances in HTTP

2016: System calls hijacking with (m)ruby

2015: Making robot with mruby

2015: Building CLI Apps for Everyone

2014: MRuby as Development Platform for Payments

2014: MRuby on LEGO Mindstorms EV3 ®

2014: Resource Control Architecture scripting with mruby

Using MRuby

build_config.rb

- Which mrbgems to use
- Which compiler to use
- Library & include path
- Can have many builds

mrbgem.rake

- Gem metadata
- Gem dependencies
- Gem binaries

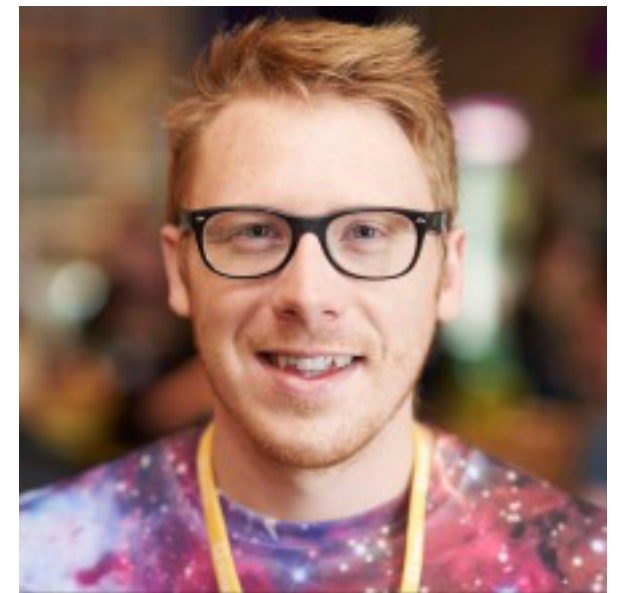
rake

rake Results

- `bin/your-cli`
- `bin/mruby`
- `bin/mirb`
- `bin/mrbtest`

mruby-cli

Building CLI Apps for Everyone



Write Ruby

release on

Linux, OS X, Windows

Using mruby-cli

1. Write MRuby
2. Compile
3. `GOTO 1 unless tests pass`
4. Release executables

MRuby Challenges

MRuby

!=

CRuby

Modular stdlib

- Minimal core library
- `mruby-array-ext`
- Easy to forget for `mrbgems`

Language Issues

- No pre-defined globals
- Some syntax bugs
- No keyword arguments for `def`
- Regexp replacements

Exceptions

- Inaccurate backtrace
- More C exploration

Porting Libraries

- Easy overall
- Beware restricted syntax
- Porting tests is hard

mrbgems

- Like Ruby 1.6.x
- Libraries have narrow focus
- Expect to submit patches

Patches

mruby-onig-regexp

mruby-curl

mruby-io

mruby-httpsclient

mruby-cli

Build Information

- Needed for cross-compile
- Hard to read without rake

mruby-cli

Challenges

Docker

Docker

- Ready to cross-compile
- Clumsy to use
- Best option available

`docker-compose run compile`

- Too long to type
- `run shell` to debug
- Unfamiliar environment

Build System

Task Organization

- MRuby and mruby-cli mixed
- Hard to add new tasks

Improvements

- Build MRuby separately
- Run docker from a task

Outer

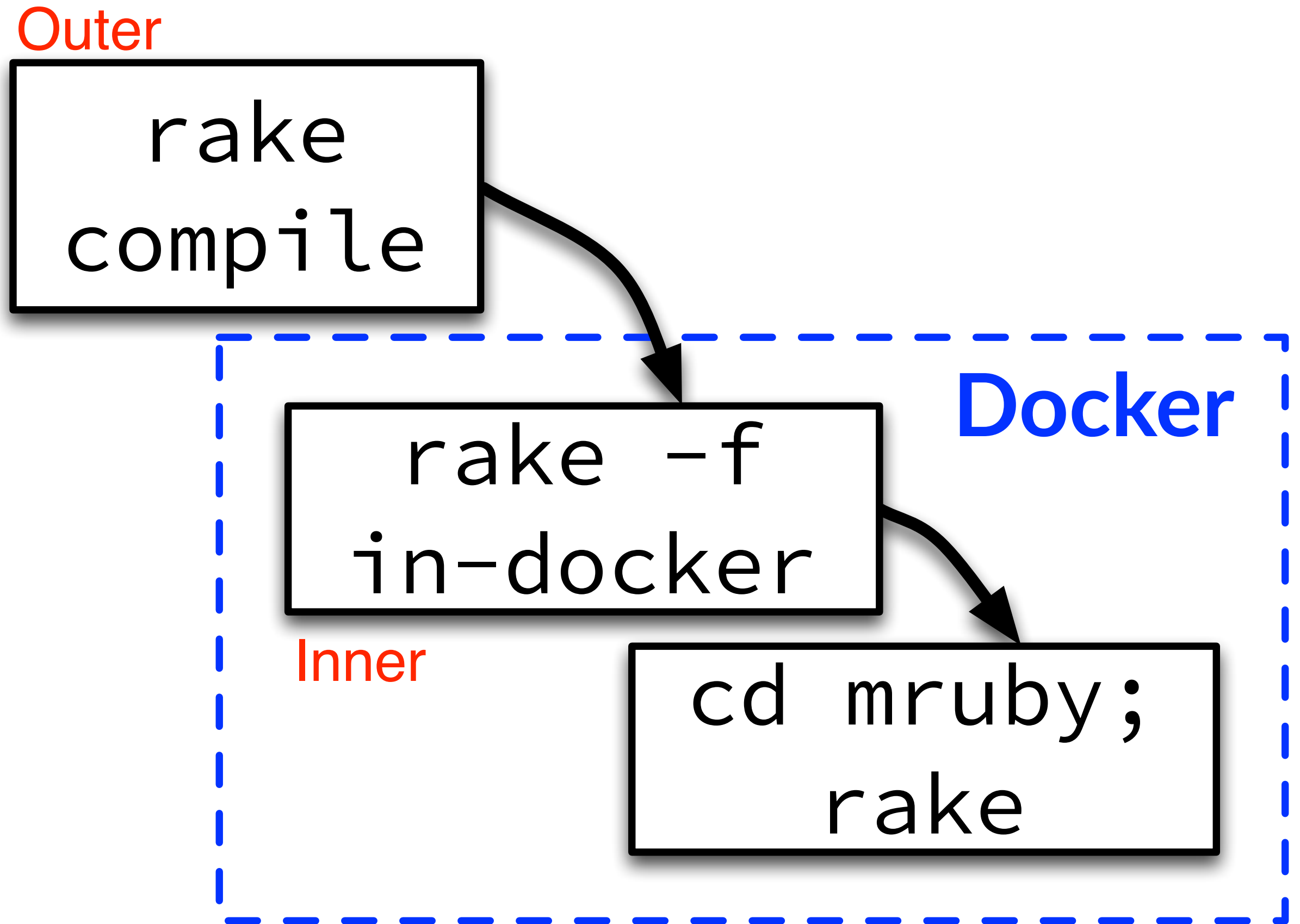
```
rake  
compile
```

Docker

```
rake -f  
in-docker
```

Inner

```
cd mruby;  
rake
```



"Outer" Tasks

- Download MRuby
- Global test setup
- Release tasks

"Inner" Tasks

- Cross-compile libraries
- Compile MRuby
- Run tests

Additional Benefits

- Hooks for custom tasks
- Faster rake test

C Libraries

Static Linking

- Single file
- Re-release on vulnerabilities
- Larger

Dynamic Linking

- Easy vulnerability management
- Adds a dependency

Cross Compiling

Cross Compiling

- Download release
- Cross-compile
- MRuby compiler configuration

CrossLibrary

```
CrossLibrary.new 'curl' do |cross|  
  cross.release_name = 'curl-7.28.0'  
  
  cross.url = "https://curl.haxx.se/..."  
  
  # TODO: cross-compile OpenSSL too.  
  cross.configure_flags << '--without-ssl'  
end
```

CrossLibrary

```
CrossLibrary.new 'curl' do |cross|  
  cross.release_name = 'curl-7.28.0'  
  
  cross.url = "https://curl.haxx.se/..."  
  
  # TODO: cross-compile OpenSSL too.  
  cross.configure_flags << '--without-ssl'  
end
```

CrossLibrary

```
CrossLibrary.new 'curl' do |cross|  
  cross.release_name = 'curl-7.28.0'  
  
  cross.url = "https://curl.haxx.se/..."  
  
  # TODO: cross-compile OpenSSL too.  
  cross.configure_flags << '--without-ssl'  
end
```

CrossLibrary

```
CrossLibrary.new 'curl' do |cross|  
  cross.release_name = 'curl-7.28.0'  
  
  cross.url = "https://curl.haxx.se/..."  
  
  # TODO: cross-compile OpenSSL too.  
  cross.configure_flags << '--without-ssl'  
end
```

CrossLibrary

```
CrossLibrary.new 'curl' do |cross|  
  cross.release_name = 'curl-7.28.0'  
  
  cross.url = "https://curl.haxx.se/..."  
  
  # TODO: cross-compile OpenSSL too.  
  cross.configure_flags << '--without-ssl'  
end
```

Part II

Customer Support Tool

Problem

- API-only to start
- Complicated setup
- Support burden

Solution

- Automate
- Validate inputs
- Reduce errors

Prototypes

Single File Script

- Easy to write
- Inflexible

Multiple Scripts

- One purpose per script
- More flexible
- Well documented

Extract classes

- JSON-API
- Argument parsing

Lessons

Document

- No API docs
- 50% comments
- Users can suggest changes

Simple Scripts

- Minimum dependencies
- Minimum options
- One script per task
- Very few large scripts

Hard to Maintain

- CRuby version
- Clone and edit by support

Switch to
mruby-cli

RubyGems Pattern

- Subclass per subcommand
- `mruby-optparse`
- `setup` for options
- `execute` to run

Example Command

```
class ConditionList < Command
  def setup
    option_service_id required: true
    option_version      required: true
  end
end
```

Example Command

```
class ConditionList < Command
  def execute
    results = api.send_request ...
    results.each do |result|
      puts "..."
```

end

end

mruby-optparse

- Familiar
- Type checking
- Argument completion
- Shell completion

mruby-curl

- Most mature HTTP library
- HTTP + HTTPS
- Persistent connections
- `libcurl` dependency

Testing

- mruby-mtest for assertions
- mruby-test as a runner
- Test case setup is painful

Testing

mruby-mtest

- Like minitest
- Test subclass
- `test_*` methods
- `setup` / `teardown`
- Remember: `MTest::Unit.new.run`

mruby-test

- Test runner
- One MRuby VM per test file
- Complete isolation

Complications

- Global setup hard
- `docker-compose` documentation
- Slow startup

Workarounds

- Setup before all tests
- Pass data out-of-band
- Compile only host for testing

rake bintest

- Integration tests
- Assert output from input

Part III

Future Work

mruby-cli

- Improved build system
- Documentation
- Easier upgrades

MRuby

- Accurate backtraces
- `build_config.rb` library
- Keyword args in methods

CRuby

- `mrb_get_args()`
 - Automatic type check
 - More expressive than `rb_scan_args()`

mruby-test

- Automatic test_preload
- --name option

Thank You

Questions?