

# Restructuring Ruby on Rails

John Wang

# Table of Contents

- Background
- Problem: Configuration and Singleton
- Related/Previous Work
- Proposed Solution: Application Configurations

# Background

**Web Frameworks and Ruby on Rails**

# Web Development Frameworks

- Web development is complicated:
  - Databases
  - HTTP Protocols
  - Templates and CSS
  - Servers
  - Asset compilation
- Frameworks help limit complexity

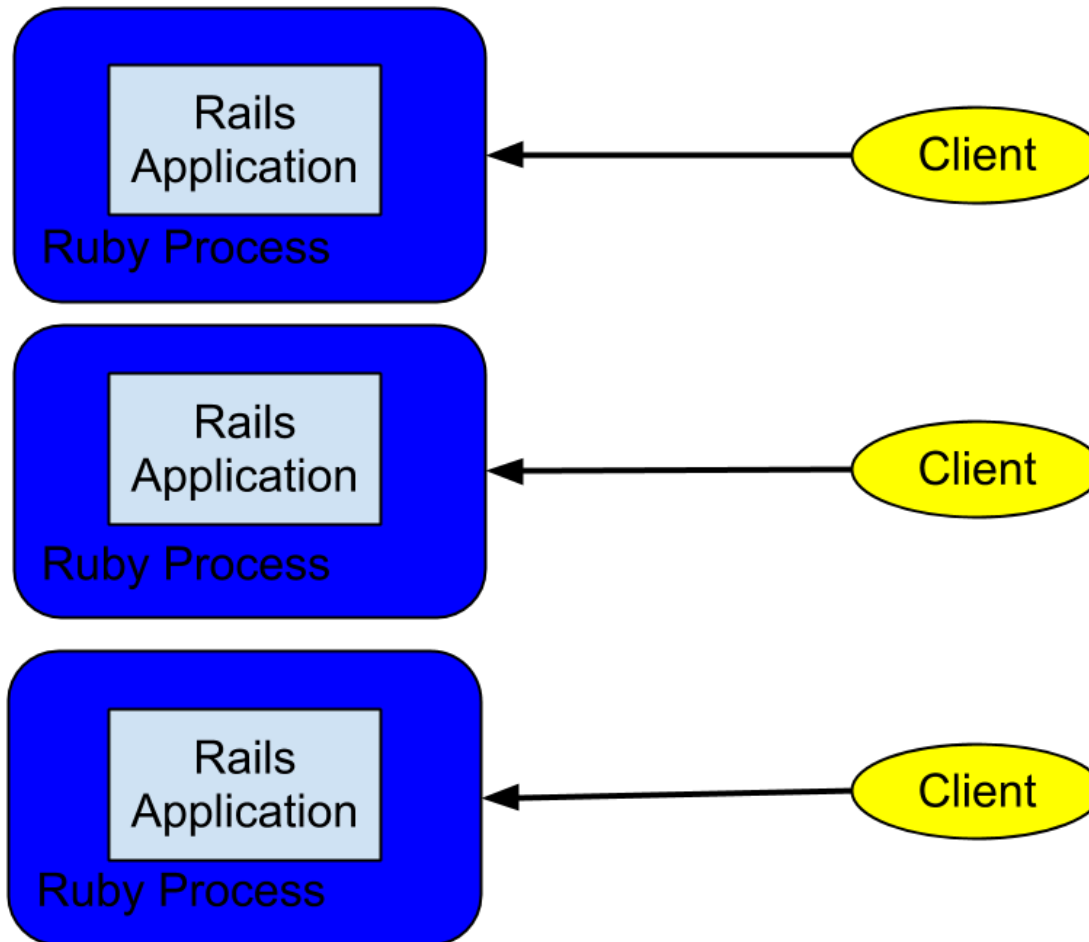
# Ruby on Rails

- Web development framework
- Built on Ruby (dynamic, interpreted language)
- Complete out-of-box solution
  - Database adapters
  - Web server handling
  - Rendering of templates

# The Problem

**Singleton and Global Configuration**

# Singleton Rails Application



# Whats Wrong With Singletons?

- Memory Usage
  - New process per application
- Global configurations
  - Prevents easy manipulation
- Prevents new applications
  - Tests require extra startup time



# Singletons and Testing

- Singleton necessitates persistent state
  - Clearing state is hard/error prone
  - Restarting the application is expensive
- Configuration before initialization is hard
  - Rails expects to be initialized in one way
  - Prevents initialization based on configuration
  - Test environment is loaded after main environment

# Who Cares?

- Ruby on Rails Developers
  - Singletons significantly slow down Rails applications
- Users of Ruby on Rails based websites
  - Twitter, Hulu, etc.
- Startups deciding between web frameworks
  - Singletons make it more difficult for Ruby on Rails to be deployed and scaled

# Related Work



# Removing Thread Locals

- Thread locals:
  - Thread level global variables
  - Cannot be accessed across other threads
- Project in the Rails Community to remove global configuration on each thread
- Involves refactoring the way configurations are passed

# Decoupling Controllers and Views

- Huge push in 2009 to decouple components
- No change to public API, but many changes to internal methods
- Rewired how controllers rendered views
  - Controller and view specific classes created
  - Only rendering methods that connected these classes

# Previous Work on Singletons

- No previous attempts to make the rails application a non-singleton
- Relatively few people attempting to change the structure
  - Most pull requests assume singleton structure, adding more technical debt
  - Singleton removal proposed by Rails Core Team, but never begun

# Proposed Solution

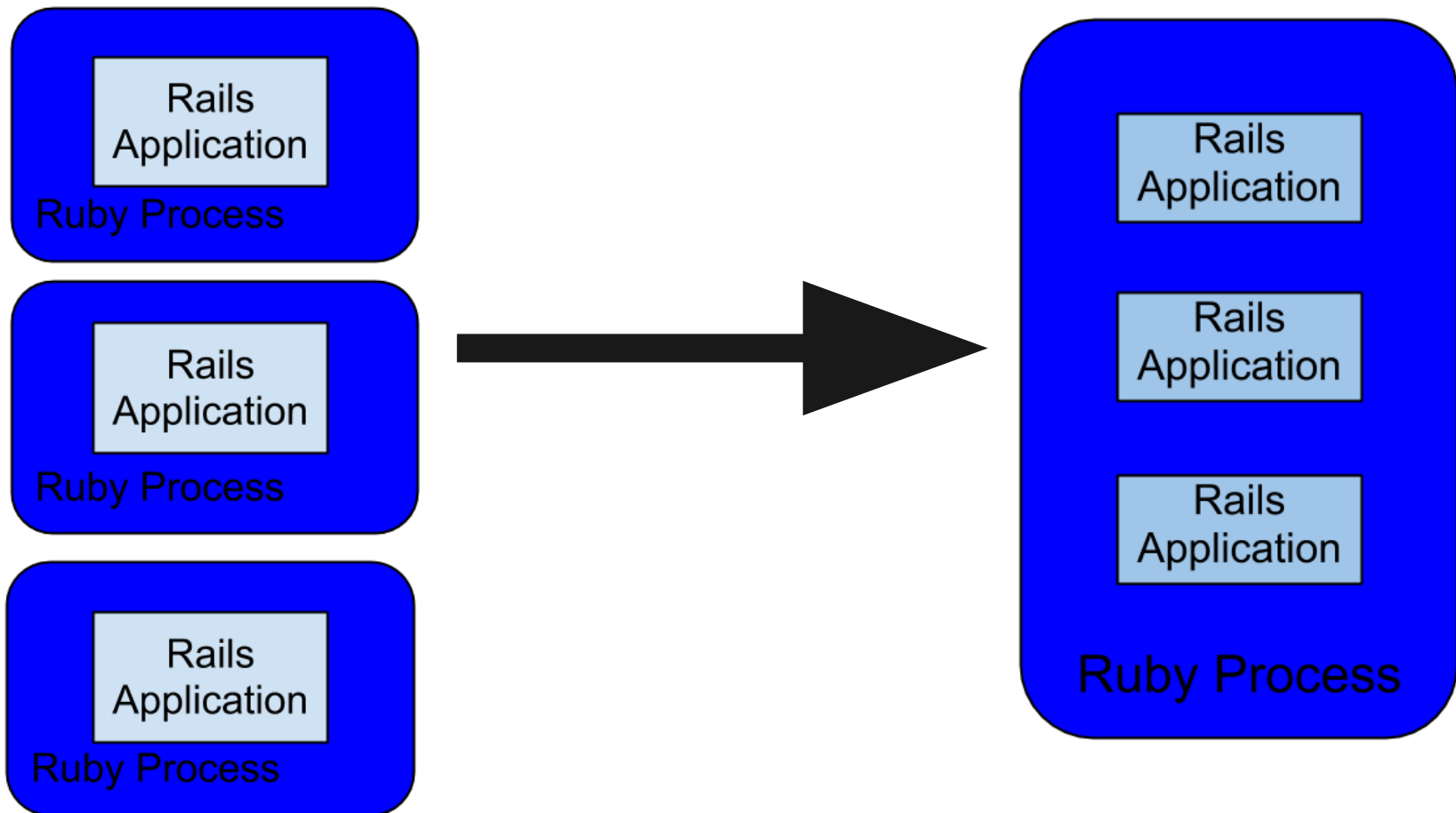
**Application Specific Configuration**

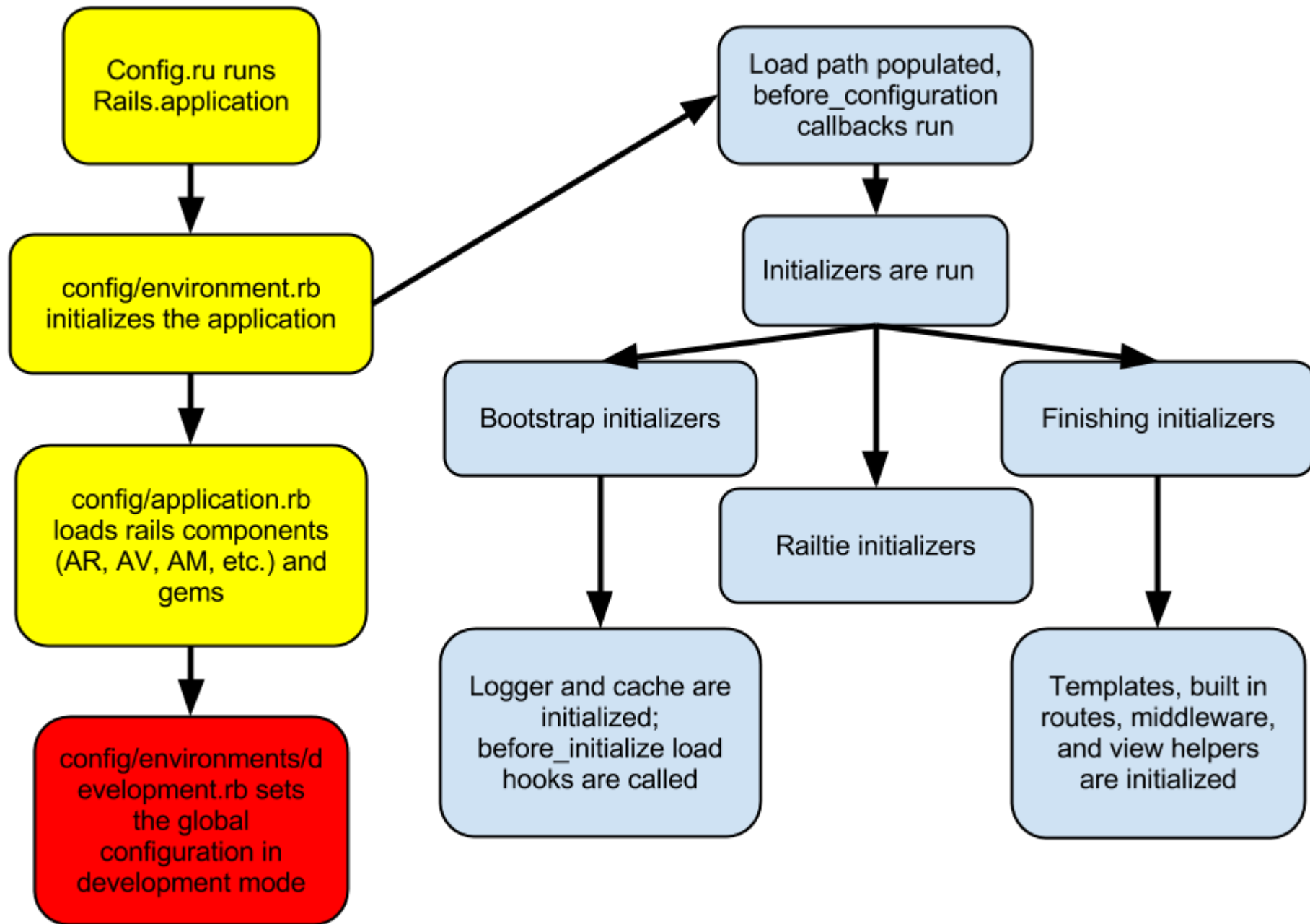
# Removal of Singleton

Overall Plan: Remove singleton and replace with application specific configuration



# Overall Plan





Config.ru runs  
Rails.application

Load path populated,  
before\_configuration  
callbacks run

## Application Factory

config/environment.rb  
initializes the application

config/application.rb  
loads rails components  
(AR, AV, AM, etc.) and  
gems

config/environments/  
development.rb sets  
the application  
configuration in  
development mode

Initializers are run

Bootstrap initializers

Finishing initializers

Railtie initializers

Logger and cache are  
initialized;  
before\_initialize load  
hooks are called

Templates, built in  
routes, middleware,  
and view helpers  
are initialized

# Phase I: Removal

- Allow multiple applications with single global configuration
- Applications are clones of each other
- Technically no longer singleton, but still global configuration remains problematic

# Phase II: Configuration

- Each Rails Application has its own configuration
- Create new applications by instantiating configuration and application objects.
- Refactor initialization procedure so it doesn't create a singleton application

# Schedule and Timeline

- **Early May:**
  - Complete Phase I
  - Have detailed architecture for completing Phase II
- **Late May:**
  - Create Application Factory
- **Early June**
  - Move configuration to before application initialization
  - Create method to duplicate applications
- **July**
  - Test and iterate on design

# Risks and Backup Plans

- Incompatibility with previous Rails versions
  - Revert to previous methods
- Introducing or uncovering a serious bug
  - Spend time working on the bugfix
- Inability to finish
  - Complete the project in phases
  - Move to a less ambitious project: removing thread locals