

The background of the slide is a blue-toned abstract image. It features a dense pattern of binary code (0s and 1s) overlaid with a network of glowing, intersecting lines that resemble fiber optic cables or data paths. The overall effect is a sense of digital connectivity and technology.

A Tour of Ruby on Rails

By David Keener

<http://www.keenertech.com>

But First, Who Am I?

David Keener

I'm a technical architect and writer with over 20 years of experience. Been doing web applications Since 1997, and Rails applications since version 1.1.

Technical Architect for Grab Networks, the company known for streaming the Beijing Olympics over the web and for distributing more news videos in the US than any other company except MSNBC.

- **Blog:** <http://www.keenertech.com>
(New Rails-based version in late June)
- **Email:** dkeener@keenertech.com



CareerBank.com

NEXTEL



THOMSON

NASDAQ

What Is Ruby on Rails?



Rails is an open-source, full-stack MVC framework for developing web-based, database-driven applications. Rails is implemented in the Ruby programming language.

- It's a tool
- It's optimized to create web sites quickly
- It's a productivity enhancer
- It's not magic
- It's not a silver bullet
- It's not going to replace Java for all applications

Open Source

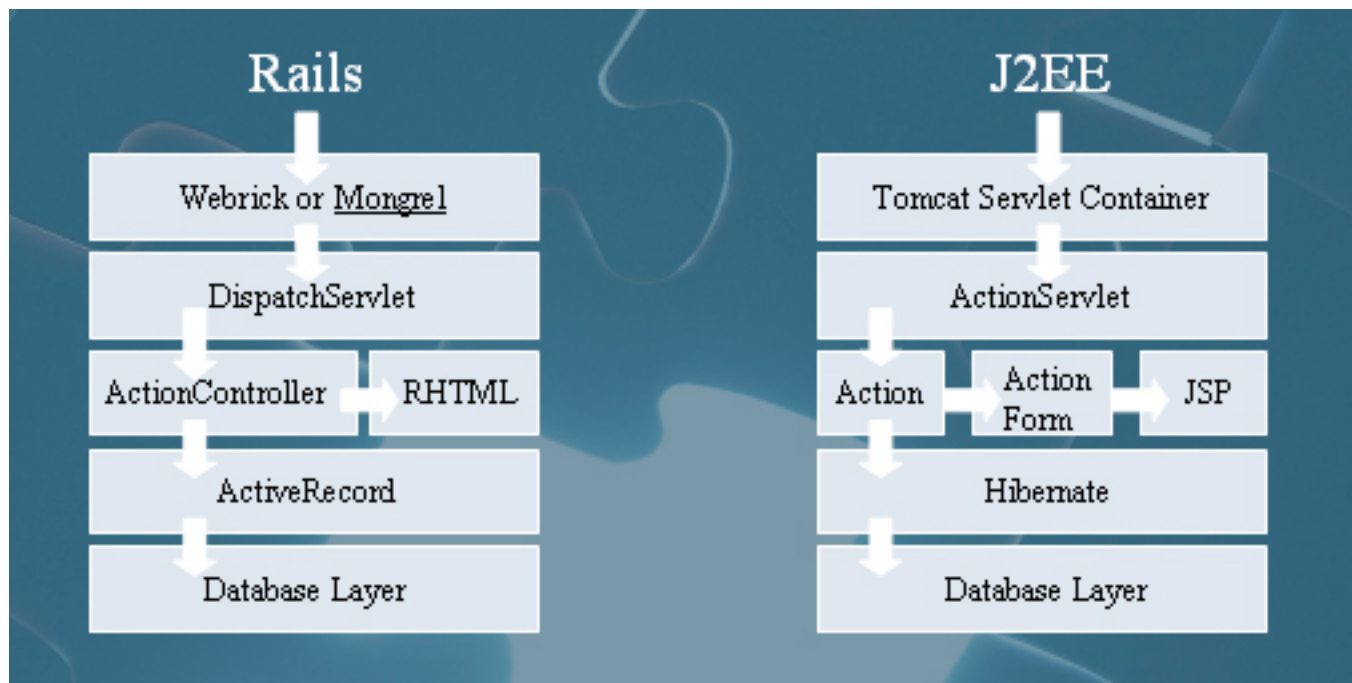
Rails is...



- Free
- Supported by a highly active community
- Easily hosted on UNIX/Linux platforms
- Integrated with popular web servers such as Apache and NGINX
- Supported by free development tools like NetBeans, Eclipse and Aptana Studio

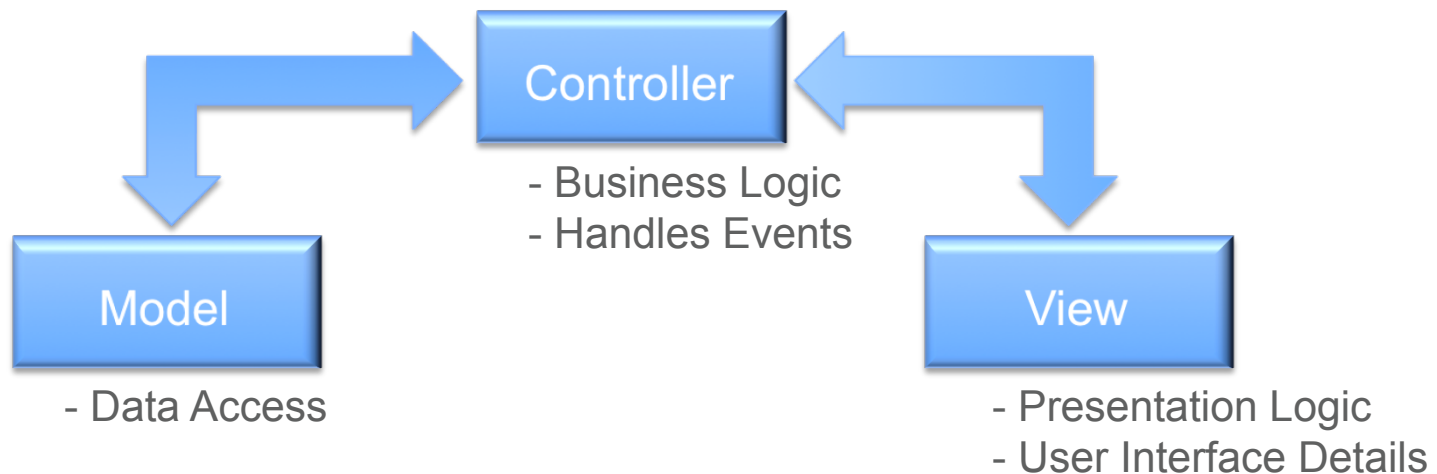
It's a Full-Stack Framework

Rails provides an integrated full stack framework. J2EE provides similar functionality with a variety of separate frameworks / tools (each with their own learning curves).



MVC

- Stands for “Model – View – Controller”
- Architectural pattern that organizes apps into distinct components with specific responsibilities





Web-Based, Database-Driven...

Rails is ideal for web applications that do CRUD operations....

- Create
- Read
- Update
- Delete

(Um, this generally sort of implies a database)

Example CRUD App

KEENERTECH

Channels Admin Home

[Home](#)
[Admin](#)
[Users](#)
[Channels](#)
[Campaigns](#)
[Videos](#)

Web 2.0
Hype Generator

Microsoft
Screen Cleaner

Entries

Entries for Articles

Search Create New

Search Reset X

Name	Status	Create Date	
Creating a Breadcrumb Trail in Rails, Part 2	Draft	01/13/2009 09:13 AM	Pages Posts Edit Delete Show
Harry Potter and the End	Draft	01/13/2009 09:13 AM	Pages Posts Edit Delete Show
Overriding Standard Delete Behavior in Rails	Draft	01/13/2009 09:13 AM	Pages Posts Edit Delete Show
Developing a Pure CSS Web Design	Draft	01/13/2009 09:13 AM	Pages Posts Edit Delete Show
Some HTML IS All Right	Published	11/09/2008 12:00 AM	Pages Posts Edit Delete Show
Managing Database Dumps Using MySQL	Published	10/09/2008 12:00 AM	Pages Posts Edit Delete Show
David Keener at RubyNation 2008	Published	10/02/2008 12:00 AM	Pages Posts Edit Delete Show
VMWare Fusion	Published	09/12/2008 12:00 AM	Pages Posts Edit Delete Show
Removing Browser Default CSS Styles	Published	08/17/2008 12:00 AM	Pages Posts Edit Delete Show
Introduction to RubyGems	Published	08/05/2008 07:25 PM	Pages Posts Edit Delete Show
Practical Tips for Rails Migrations	Published	07/20/2008 08:14 PM	Pages Posts Edit Delete Show
Creating a Breadcrumb Trail in Rails, Part 1	Published	06/18/2008 12:00 AM	Pages Posts Edit Delete Show
If You Can't Find the Oracle	Published	06/10/2008 12:00 AM	Pages Posts Edit Delete Show
AOL Exit Interview	Published	05/05/2008 12:00 AM	Pages Posts Edit Delete Show
Mac Office 2008 Product Key Issue Solved	Published	04/17/2008 06:03 AM	Pages Posts Edit Delete Show

88 Found 1 2 3 .. 6 Next

[Logout](#) Copyright © 2004 - 2009 KeenerTech.com [Site Map](#)



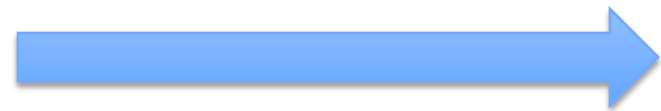
What's the Big Deal?

So, Rails is a framework.

It's not really doing anything new.

There are lots of frameworks....

What makes Rails so special?



One picture is worth....



Note: Graphic found on web.



The Goals of Rails

To solve any problem, first define the problem clearly. Here's what the creators of Rails set out to accomplish...

- Build web sites
- Build web sites quickly
- Increase productivity by eliminating or reducing common drudgery
- Make programming fun again by letting developers focus on features that matter



The Rails Philosophy

Philosophy separates Rails from other frameworks....

- Convention over configuration
- Do not repeat yourself (DRY)
- The 80/20 Rule: Rails implements 80% of the functionality that most users need; the other 20% is your application
- Corollary to the 80/20 Rule: You have the power to extend the framework yourself



“Opinionated Software”

Rails was developed by David Heinemeier Hansson in 2003 (released to the public in 2004). He and the members of the Rails Core Team had definite opinions about what makes a framework good.

- Philosophies are baked into the design of Rails
- Rails should make it hard not to do agile development
- Ruthless about what features are included in Rails...and which are not



Primary Rails Features

- **Migrations:** Facilitates DB maintenance
- **Code Generators:** Get up-and-running quickly with scaffolding (replace it later)
- **Object-Relational Mapping:** Uses ActiveRecord to automatically generate class methods based on database lookups
- **MVC:** No configuration required; just follow the Rails conventions
- **Integrated Features:** Database access, AJAX, Testing, Text/HTML emails, etc.



Migrations

- Automates building/rebuilding database
- Ensures all database-related components are stored in one location
- Simplifies a key area of drudgery in most projects
- Includes database-agnostic methods for manipulating a database
- Migrations are numbered, so there's no confusion about what order to run them in

An Example Migration

Name: 005_change_operations.rb

To run: rake db:migrate

```
class ChangeOperations < ActiveRecord::Migration
  def self.up
    add_column :operations, :org_id, :integer
  end

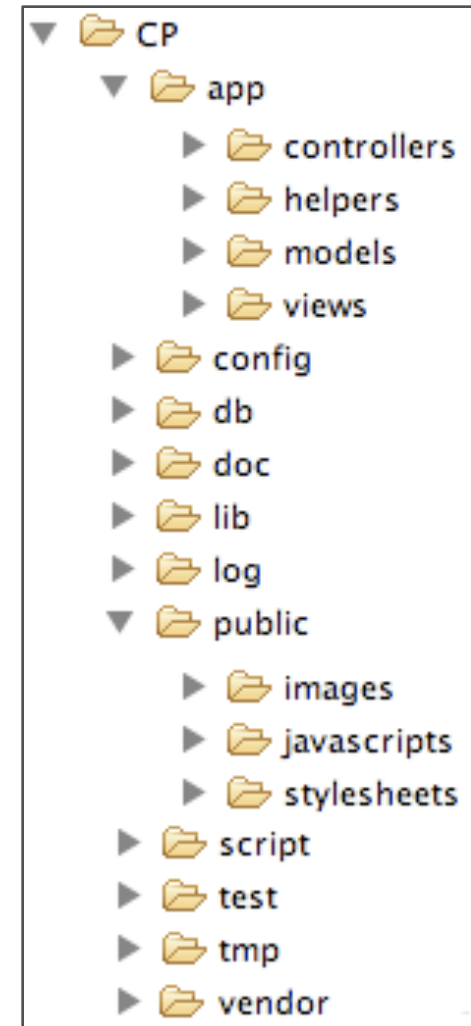
  def self.down
    remove_column :operations, :org_id
  end
end
```


Code Generator (1)

- Generate the structure of an app:

\$ rails appname

- Eliminates the drudgery of determining your app structure
- Makes maintenance easier because all Rails apps look similar





Code Generators (2)

You'll hear a lot about Scaffolding....

- Create a model, view, controller and migration with one command

```
$ ruby script/generate scaffold product  
name:string desc:text status:boolean
```

- Gets a new set of CRUD web pages up in minutes
- Replace the Scaffolding with more functional code as your development schedule allows



Convention Over Configuration

- Note that the Scaffolding set up an MVC feature with no configuration required
- If you follow standard Rails naming conventions, Rails automatically knows where to find the things it needs
- Some key files that got created:

`/app/controllers/products_controller.rb`

`/app/models/product.rb`

`/app/views/products/index.html.erb` (plus other CRUD views)

`/db/migrate/009_create_products.rb`

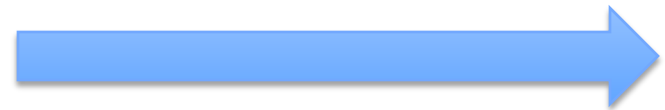
`/test/unit/product_test.rb`



Object-Relational Mapping

This is a fancy term for “how your objects are going to access the database.”

- ActiveRecord – A component of Rails that handles database access
- Eliminates most of the drudgery involved in working with a database
- The example model on the next page shows just how simple this is...



Example Model

```
class Product < ActiveRecord::Base  
end
```

- Two lines of code!
- Rails queries the database and determines the data types and fields for you
- Relationships between models and field validations can be added later

Example Controller Method

```
def index  
  @products = Product.find(:all)  
end
```

- Uses our Product model
- Uses the “find” capability provided by ActiveRecord to retrieve products from the database
- No SQL required!



But Wait, There's More!

- Built-in support for testing
- Integrated AJAX capabilities: Let's you make web pages function like Desktop apps
- Integrated JavaScript libraries: Prototype and Scriptaculous – Supports dynamic web page effect
- Email capabilities – Emails are just another type of view

About Productivity...



It's kind of like having a jet pack

- There's a learning curve...
- But once you learn Rails, you can really fly



Who's Using It?

- In the DC area, most startups are using Ruby on Rails
- Ideal for small businesses
- Ideal for internal corporate apps
- Slower penetration into large companies because they're inherently conservative
- Even slower penetration into government, although that is changing



How to Leverage Rails

- Rails is about doing more work in less time with fewer people
- Small, focused teams work well
- Frequent releases work well due to the quick development process
 - 2 week cycles at Voxant
 - Monthly cycles at Grab Networks
- Ideal for agile methodologies



Getting Started With Rails

- Agile Web Development With Rails, 3rd edition, by Sam Ruby, Dave Thomas, David Heinemeier Hansson – An award-winning technical book; the best way for developers to learn Rails
- <http://api.rubyonrails.org> - The Rails API documentation
- <http://rubyforge.org> - One of the main repositories for open source Rails add-ons



What About Hosting?

- **EngineYard** – Relatively expensive, but solid for corporate hosting
- **DreamHost** – Relatively low cost
- **WebFaction** – Low-cost, high quality...targeted for savvy developers
- Rails can be hosted on UNIX/Linux boxes easily
- Typically hosted using either Mongrel or Phusion Passenger (usually in combination with Apache or NGINX)