

Ruby On Rails

James Reynolds

- * What is a Ruby on Rails
- * Why is it so cool
- * Major Rails features

Web framework

- * Code and tools for web development
- * A webapp skeleton
- * Developers plug in their unique code

Platforms

- * Windows
- * Mac OS X
- * Linux

Installation

- * Mac OS X 10.5 will include Rails
- * Mac OS X 10.4 includes Ruby
 - * Most people reinstall it anyway
- * From scratch
- * Drag and drop Locomotive

Databases

- * Mysql

- * SQLite

- * PostgreSQL

- * DB2

- * Oracle

- * Firebird

- * SQL Server

- * more

Webservers

- * Apache w/ FastCGI or Mongrel
- * LightTPD
- * WEBrick

"IDE's"

- * TextMate and Terminal (preferred)
- * RadRails
- * jEdit
- * Komodo
- * Arachno Ruby

Has "inspired"

Grails

CakePHP

Trails

PHP on TRAX

Sails

MonoRail

Catalyst

TrimPath Junction

Pylons

WASP

ColdFusion on Wheels

And perhaps more...

Why is it so cool?

- * Using the right tool for the job

- * $y = x^2$ vs $y = x^{0.5}$

Right tool

























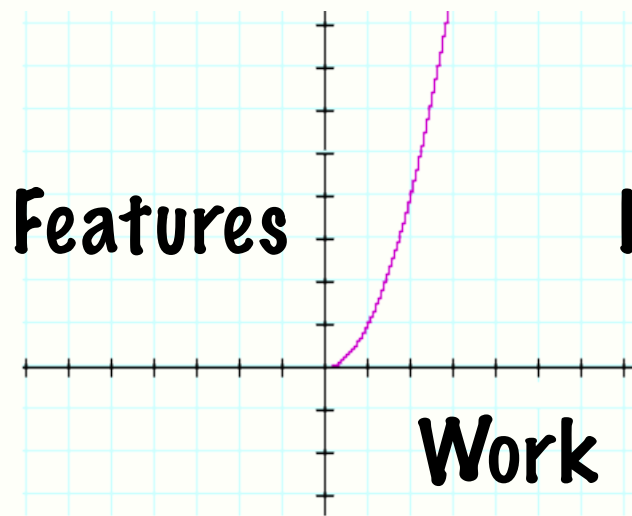




Rails is the most well thought-out web development framework I've ever used. And that's in a decade of doing web applications for a living. I've built my own frameworks, helped develop the Servlet API, and have created more than a few web servers from scratch. Nobody has done it like this before.

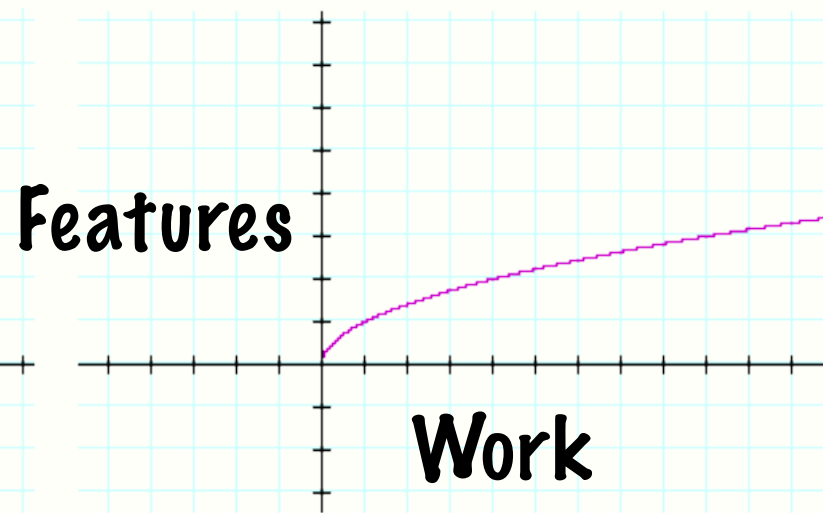
James Duncan Davidson, Creator of Tomcat and Ant

$$y = x^2 \text{ vs } y = x^{0.5}$$



$$y = x^2$$

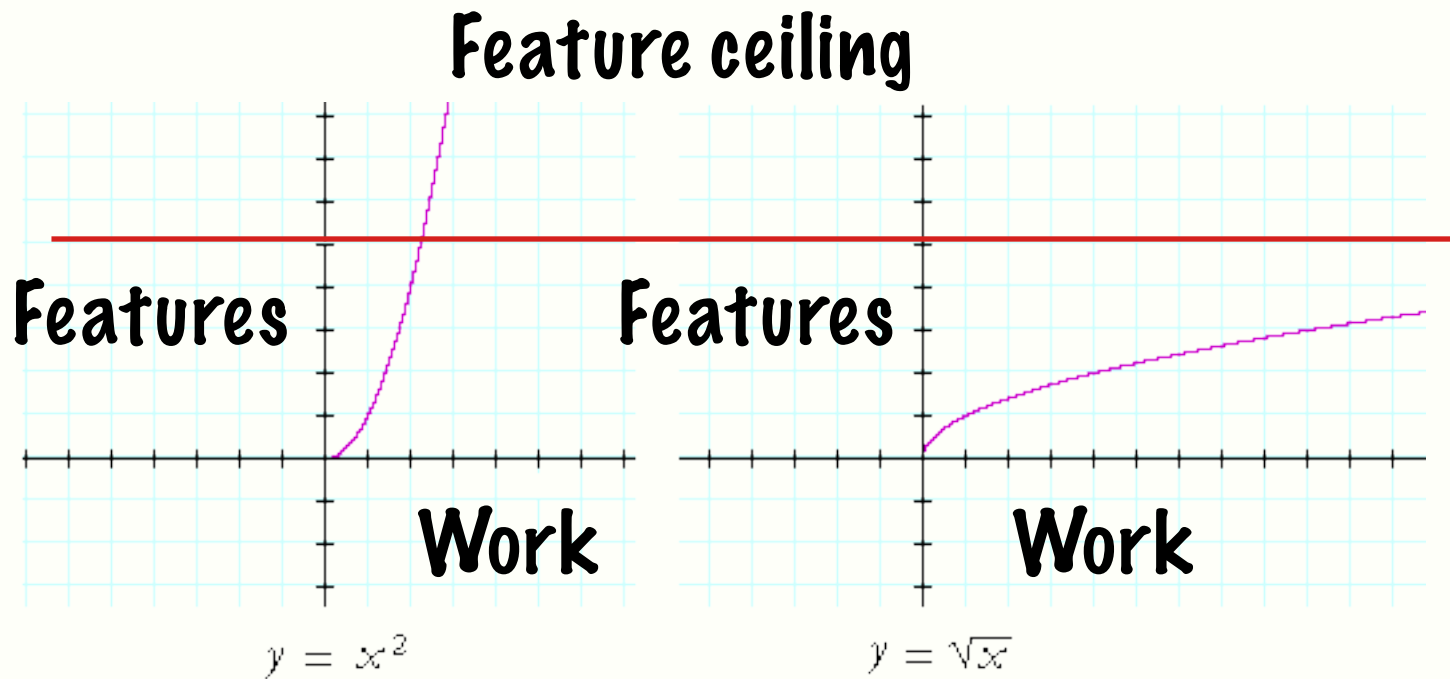
Typical



$$y = \sqrt{x}$$

Rare

$$y = x^2 \text{ vs } y = x^{0.5}$$



This is a no-brainer...

Ruby on Rails is a breakthrough in lowering the barriers of entry to programming. Powerful web applications that formerly might have taken weeks or months to develop can be produced in a matter of days.

- Tim O'Reilly, Founder of O'Reilly Media

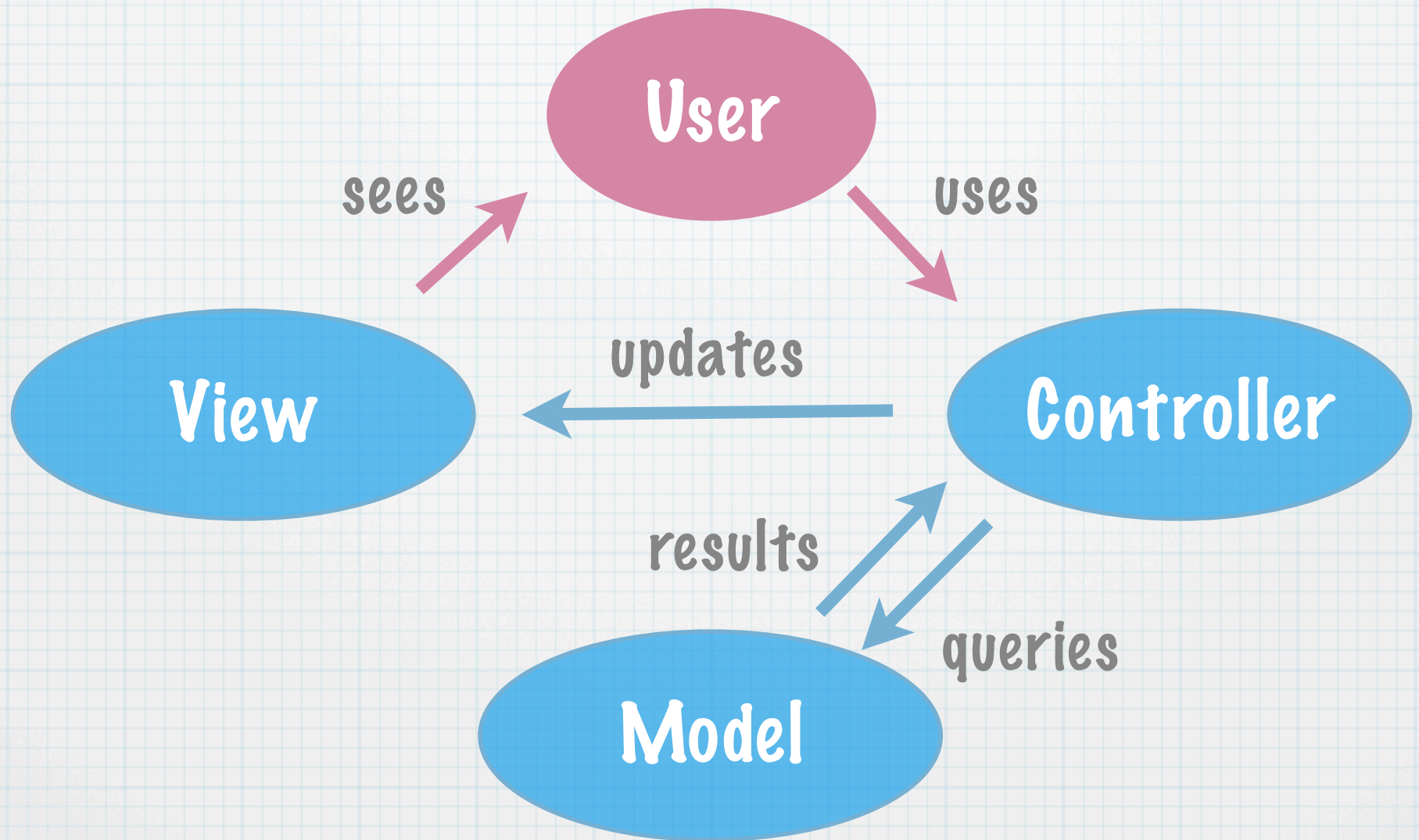
Major Rails features

- * Model View Controller
- * Run Environments
- * Unit testing
- * Migrations
- * Scripts
- * ActiveSupport

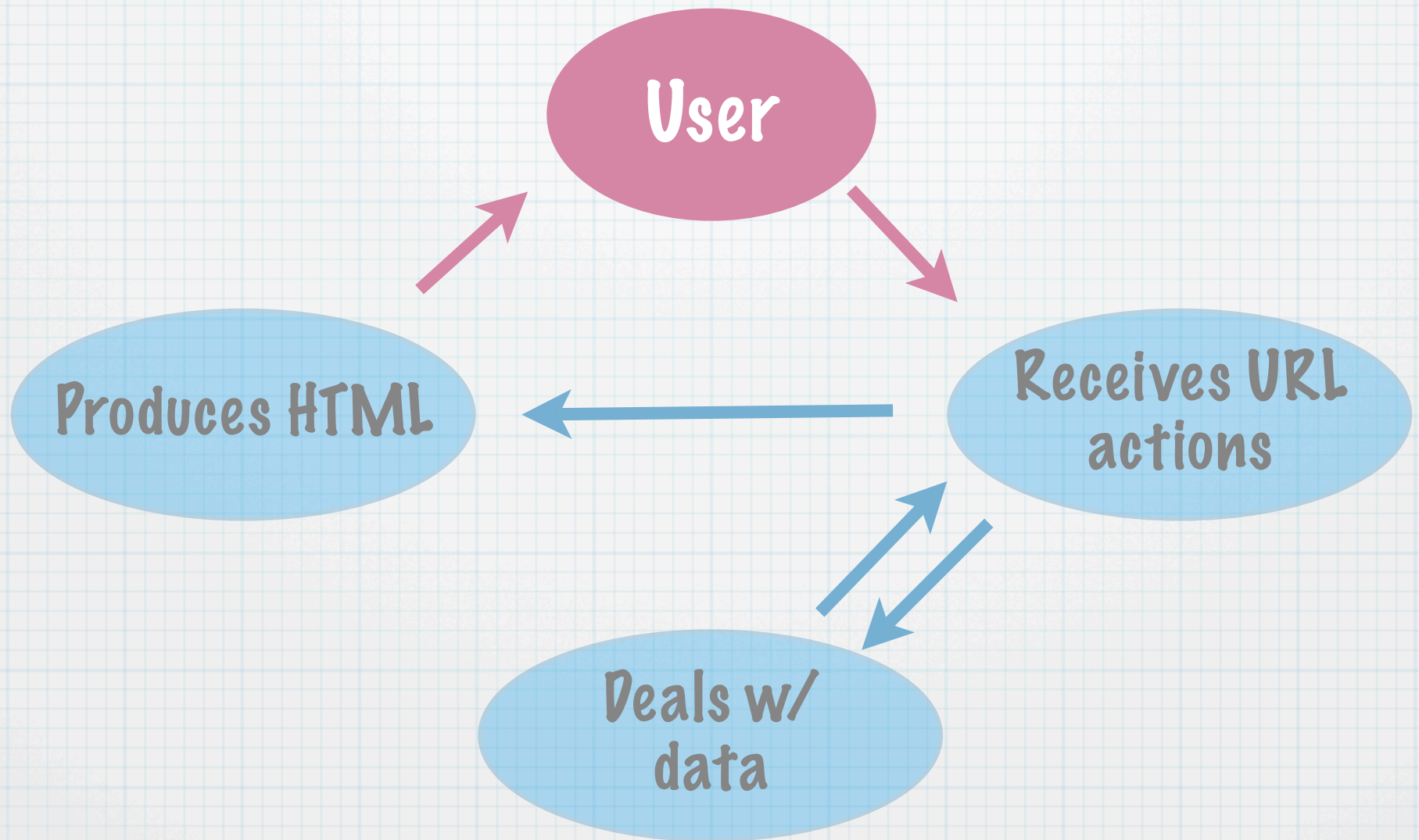
MVC

- * Described by Trygve Reenskaug in 1979
- * Totally ignored in web dev
 - * Except
 - * WebObjects
 - * Struts
 - * JavaServer Faces

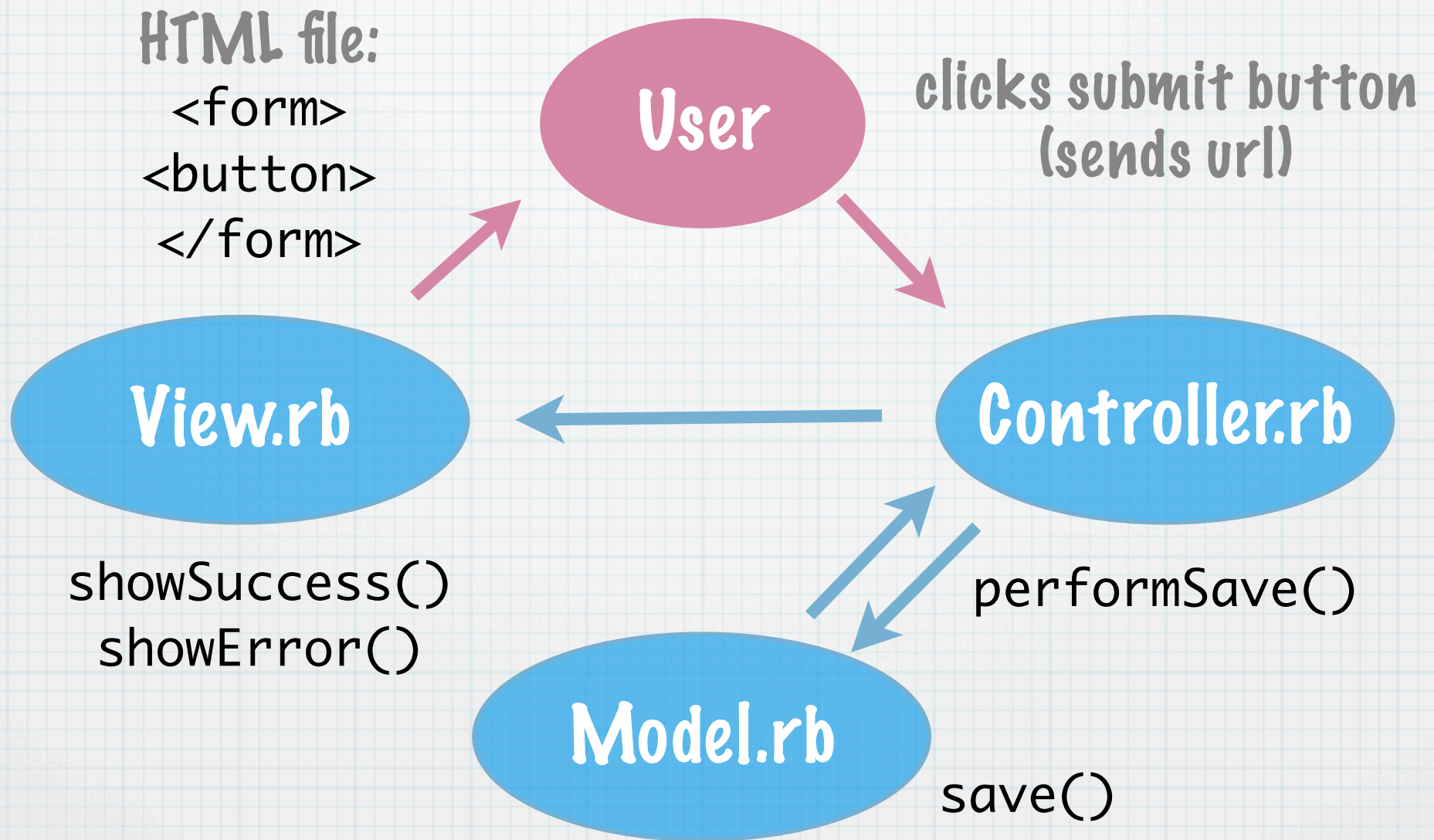
Model-View-Controller



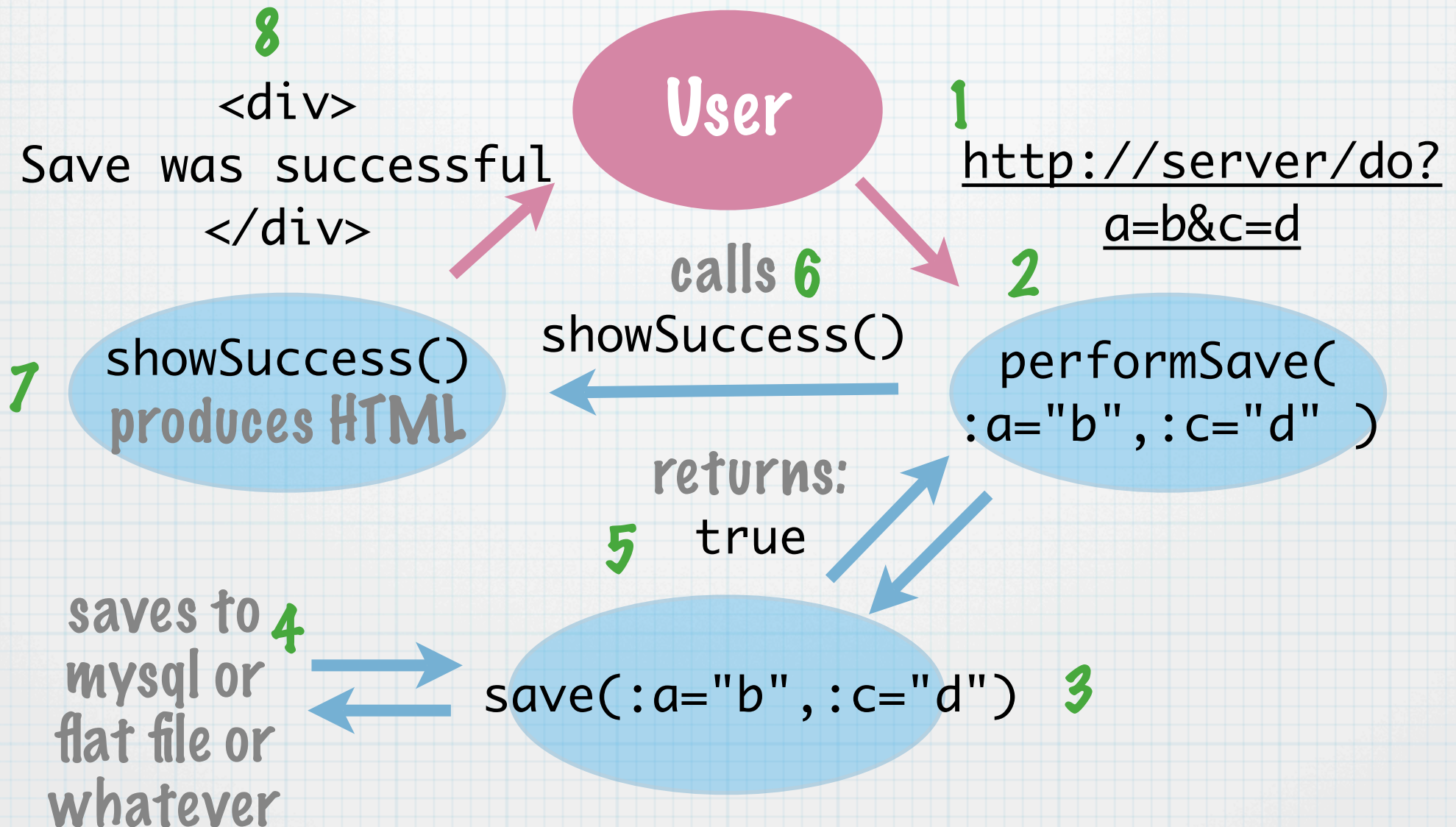
How it works



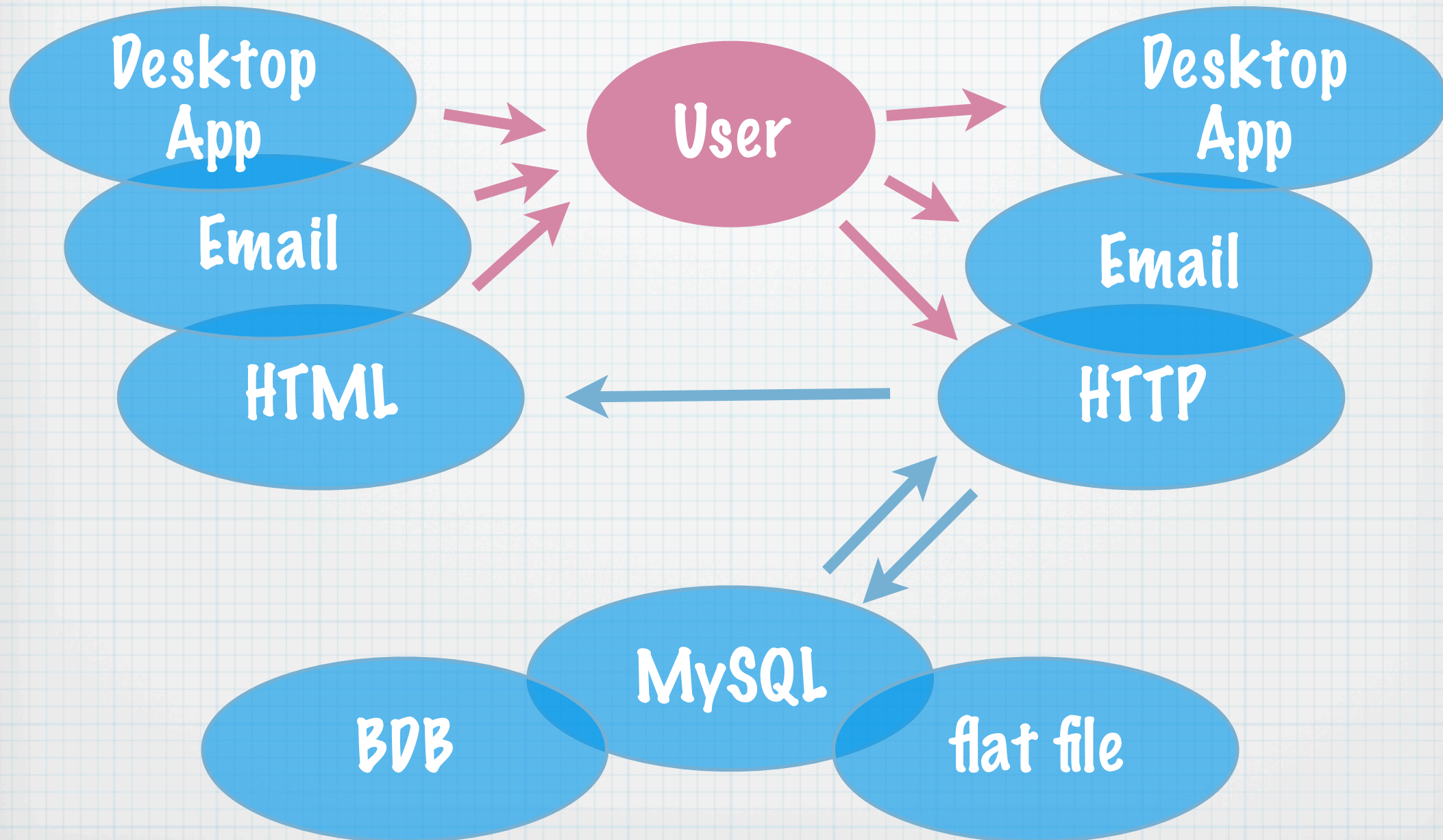
How it works



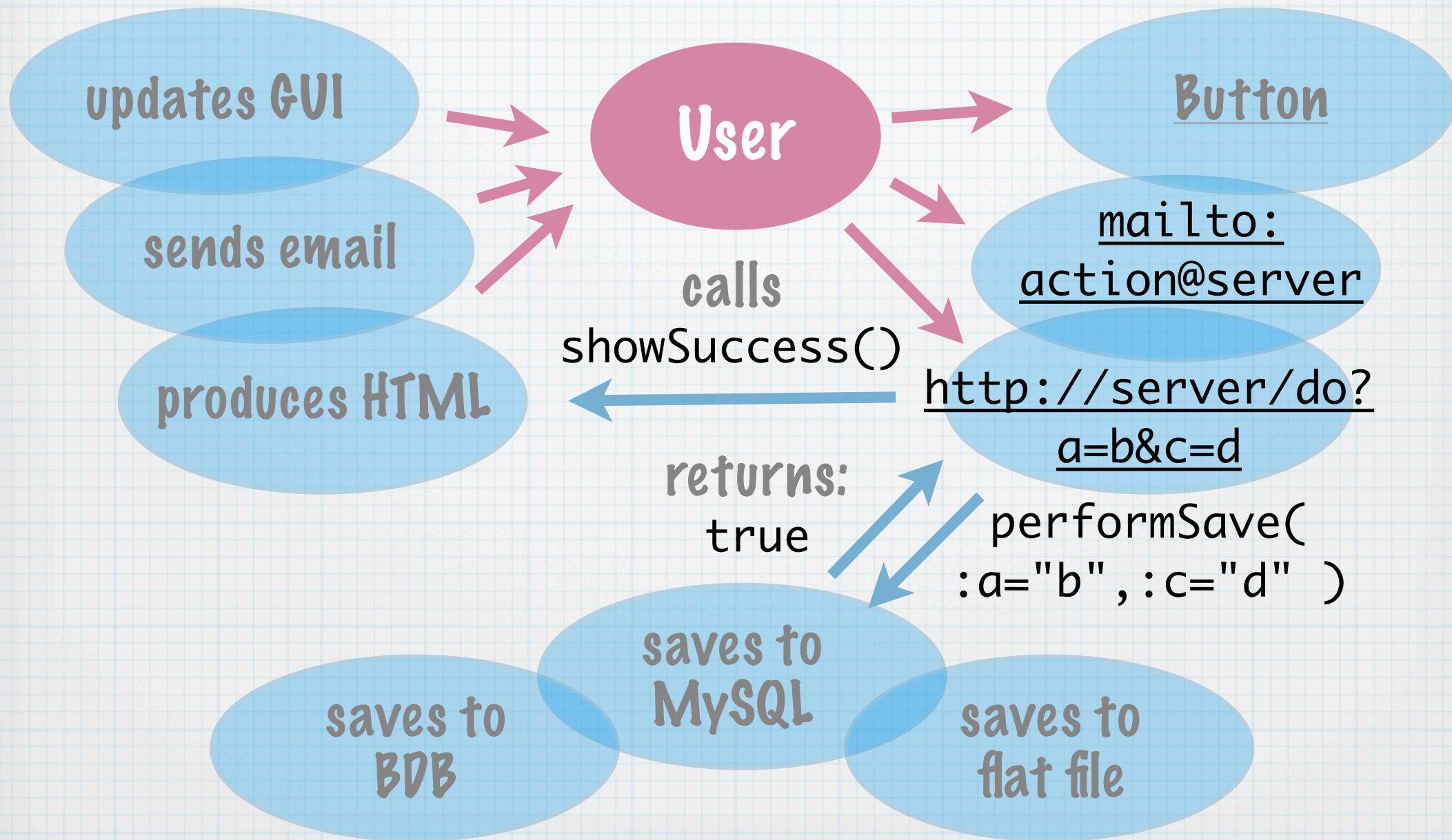
How it works



Why use it?



Why use it?



Models

- * DB backed
 - * Easy to switch db's
- * Stored on server as files
- * Stored in session objects
 - * useful for one server
 - * manage with a db for many servers

Models

- * Object relational mapping
 - * tables = classes
 - * rows = arrays of objects
 - * columns = attributes/variables
- * Maps form submissions to classes
- * No need to write SQL

Models

- * Naming conventions
 - * Table names are plural
 - * A Class is singular
 - * Table attributes auto discovered
 - * Comes with find method
 - * etc

Models - Validation

- * Fields are not empty
- * Numerically of fields
- * Uniqueness of text
- * Matches text to regex
- * Acceptance (checkbox)
- * 2 password fields same
- * Length of text
- * Custom validators
- * Etc

Models - Validation

```
class Person < ActiveRecord::Base

  validates_acceptance_of :terms, :message
  => "Please accept the terms to proceed"

  validates_exclusion_of :age, :in
  13..19, :message => "Cannot be a teenager"

  validates_format_of :height, :with
  => /^\\d+(in|cm)/

end
```


Models - Connecting

- * One to one relationships
 - * has_one
 - * belongs_to
- * One to many relationships
 - * has_many
- * Many to many relationships
 - * has_and_belongs_to_many

Models - Connecting

```
class Project < ActiveRecord::Base  
  belongs_to :portfolio  
  has_one :manager  
  has_many :milestones  
  has_and_belongs_to_many :categories  
end
```


Views

- * Create html, xml, and js
- * Flash
 - * `flash[:notice] = "Message here"`
- * Ajax built in and dumb simple

Tangent - AJAX

- * Asynchronous JavaScript and XML
- * How it works
 - * Mouse click runs a JavaScript
 - * JS contacts server
 - * Server replies with XML
 - * JS parses XML and updates HTML
- * NO PAGE RELOADS

Tangent - AJAX

- * Google uses AJAX heavily
 - * Google Maps
 - * Google Calendar
 - * The MS killing Word Processor
 - * Etc

Controllers

- * Handles external requests to internal methods
- * Pretty friendly urls
 - * NO: <http://example.com/?node=34>
 - * Yes: <http://example.com/blog/view/34>
- * Caches

Run Environments

- * Production
- * Development
- * Testing

Run Environments

- * Production
 - * Cached
 - * Freeze Rails
 - * Ship Rails with your app
 - * etc

Run Environments

- * Development
 - * Reloads source files every time
 - * Scaffold

Run Environments

- * Testing
 - * Connect debugger to running webapp
 - * Stop at breakpoints
- * Unit testing
 - * Integration testing
 - * Functional testing
 - * DB is reloaded w/ each test
 - * Mock and stub code

Unit testing

```
require 'test/unit'

def mult param1, param2
  param1 * param2
end

class TestMath < Test::Unit::TestCase
  def test_simple
    assert_equal( mult( 4, 4 ), 16 )
  end
end
```

Other killer features

- * Migrations
 - * Add/remove database scheme changes
- * script/runner
 - * cron for your db
- * script/console
- * profiler

Sessions

- * Cookie based
- * You just add stuff to the session hash
 - * `session[:something] = "value"`
- * Hash stored on the server on disk or in db
 - * Not stored in cookie
 - * Easy to use a db

ActiveSupport

- * `number_to_currency`
- * `5.months + 30.minutes`
- * `"cat".pluralize => "cats"`
- * `240.megabytes + 2.petabytes`
- * `20.minutes.ago`

You can recognize truth by its beauty and simplicity. When you get it right, it is obvious that it is right.

Richard Feynman, Genius Scientist

Demo