

Rails3



RailsConf2009 - matt aimonetti

WHO AM I?

m a t t a i m o n e t t i

“morally loose, cheese eating surrender monkey”

WHO AM I?

m a t t a i m o n e t t i

“morally loose, cheese eating surrender monkey”

Ted Han

WHO AM I?

m a t t a i m o n e t t i

“morally loose, cheese eating surrender monkey”

Ted Han

Ruby Addict

WHO AM I?

m a t t a i m o n e t t i

“morally loose, cheese eating surrender monkey”

Ted Han

Ruby Addict

Merb

WHO AM I?

m a t t a i m o n e t t i

“morally loose, cheese eating surrender monkey”

Ted Han

Ruby Addict

Merb

Rails

WHO AM I?

m a t t a i m o n e t t i

“morally loose, cheese eating surrender monkey”

Ted Han

Ruby Addict

Merb

Rails

MacRuby

WHO AM I?

m a t t a i m o n e t t i

Rails3



RailsConf2009 - matt aimonetti



MERGE



THE HISTORY CHANNEL.

THE HISTORY CHANNEL







Theorem #1

Languages influence human thought,
more than you think

oscon 2003



Summary II

Good programming languages help you to

- program better
- think smarter
- finish your job quicker



How to Reduce Stress in Programming

- Principle of Least Surprise
- Principle of Succinctness
- Principle of Human Interface




Time is Money

We can approximate our stress by the time we spent.

- The quicker we program, the more we can program.
- The faster we finish our job, the better programmer we can be.
- Succinctness is Power
 - <http://www.paulgraham.com/power.html>
- Place your burden to machine's shoulders



37signals 



RUBY ON RAILS



Convention over Configuration

Don't repeat yourself

LARGE ADOPTION



SCHISM





Performance

Agnosticism

merb

Adobe

Wikimedia

sproutcore/Apple

AT&T/yellowpages.com

powerset/microsoft

...

SPLIT RUBY COMMUNITY



DRAMA



...

SAT DOWN & TALKED



...

DIFFERENT PERSPECTIVES



SAME GOAL





MERGE





WHAT DO YOU GET?





vaporware

WHAT ARE YOU GOING TO
GET

improved
performance

WHAT ARE YOU GOING TO
GET

increased modularity

WHAT ARE YOU GOING TO
GET

agnosticism

WHAT ARE YOU GOING TO GET



public API

WHAT ARE YOU GOING TO GET



WHAT DOES IT REALLY MEAN?



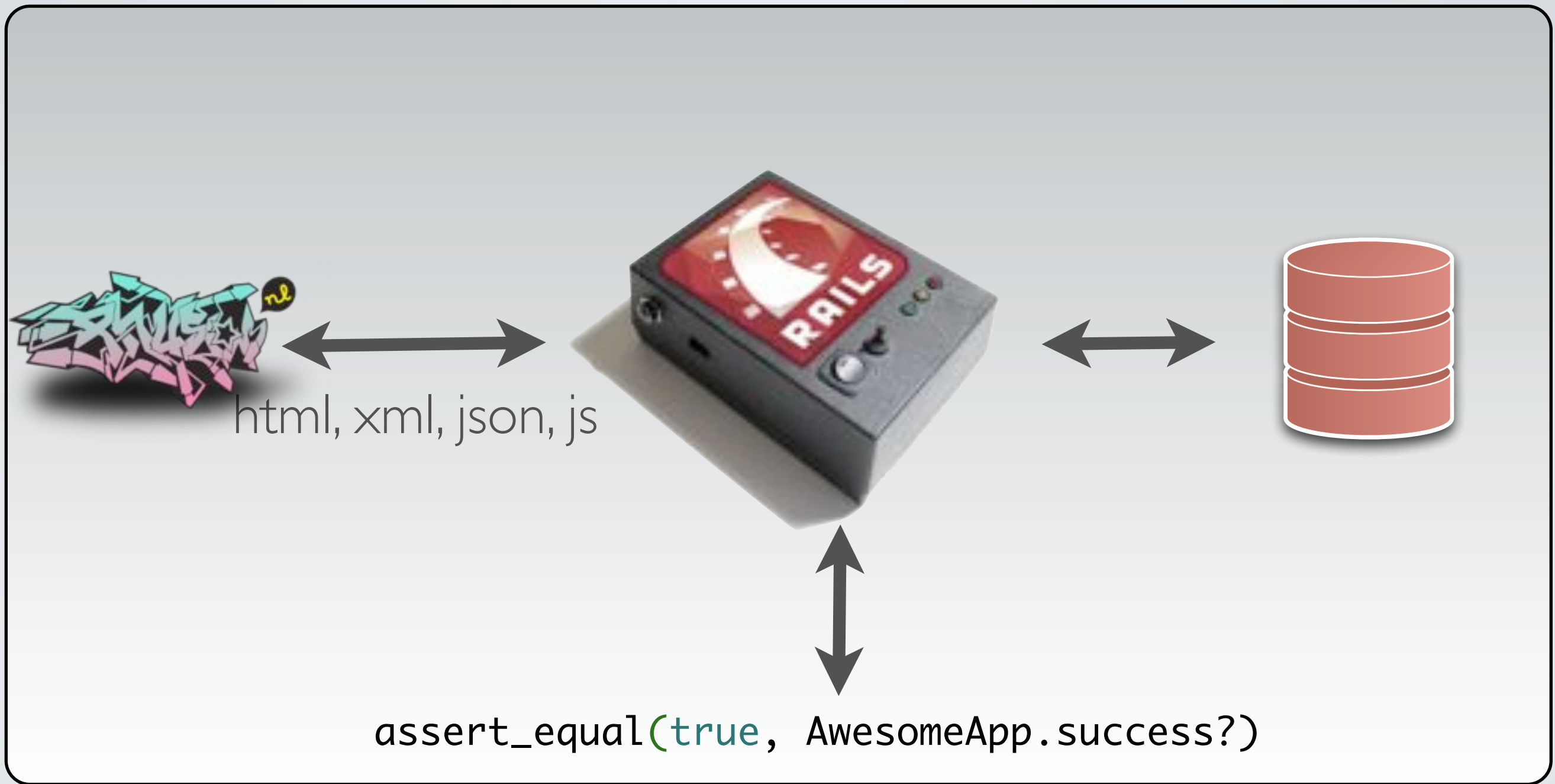
Rails 2

WHAT DOES IT REALLY MEAN?

Rails 3

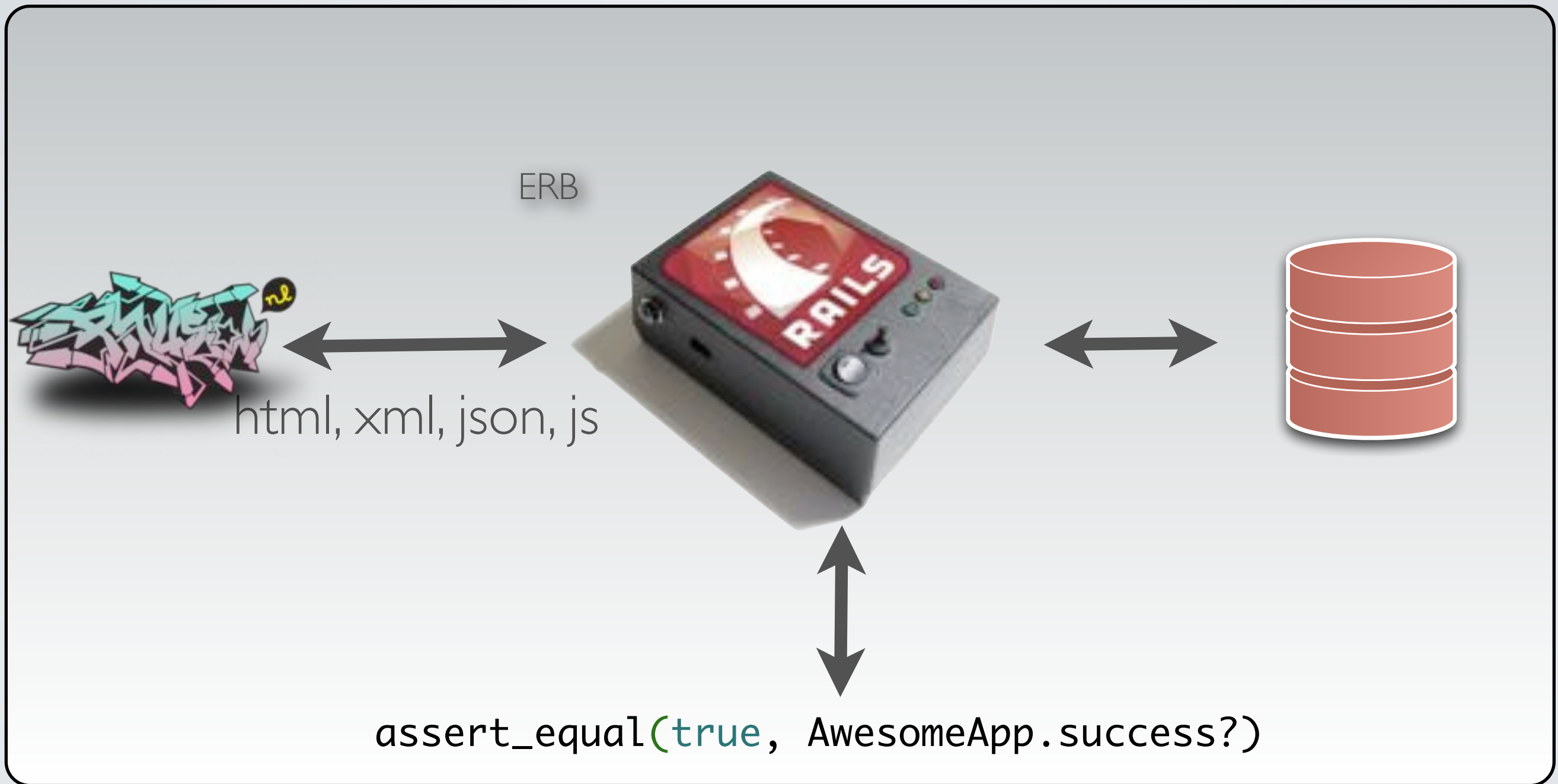


CECI N'EST PAS LA RÉALITÉ



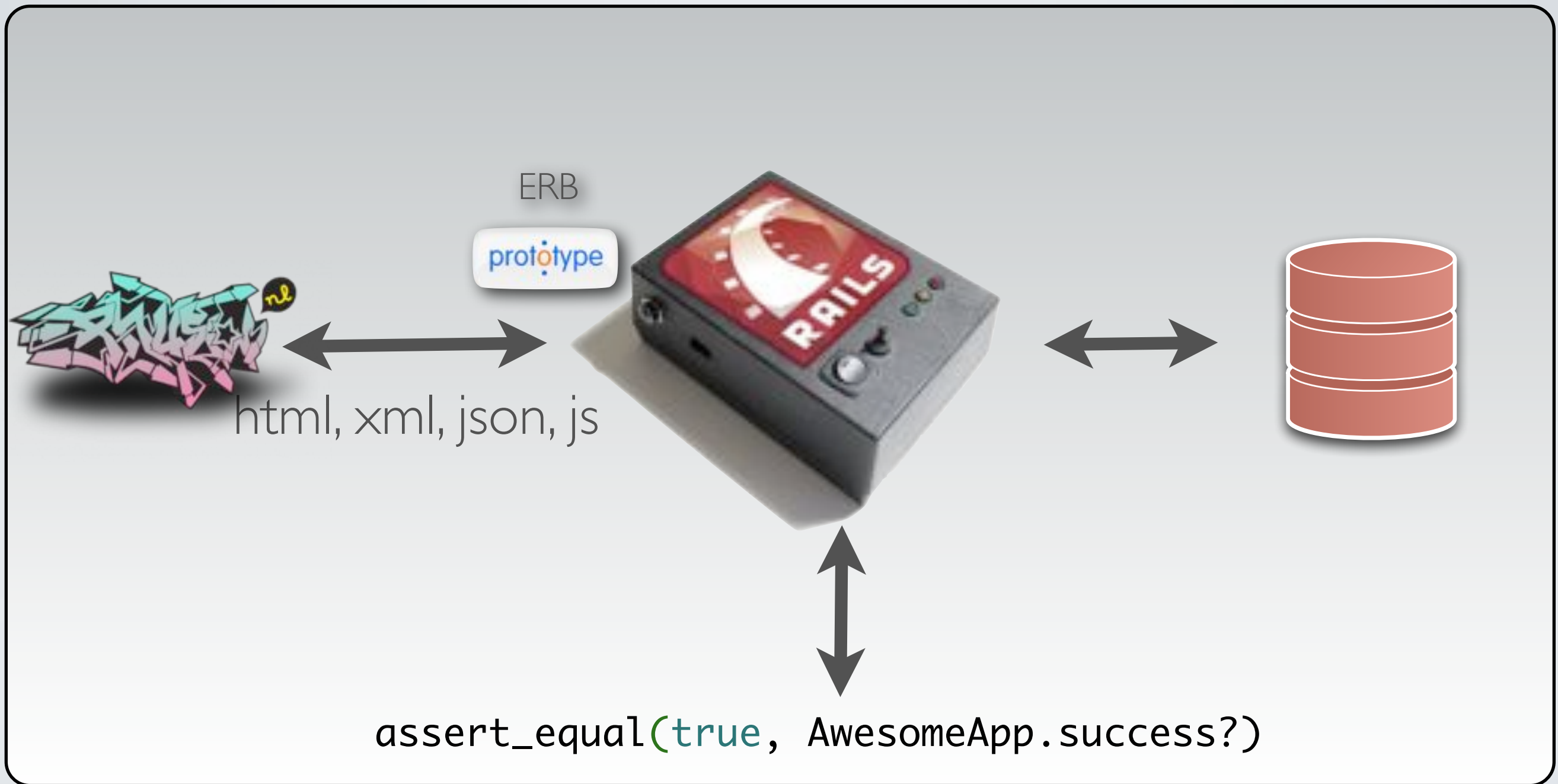
Rails 2

CECI N'EST PAS LA RÉALITÉ



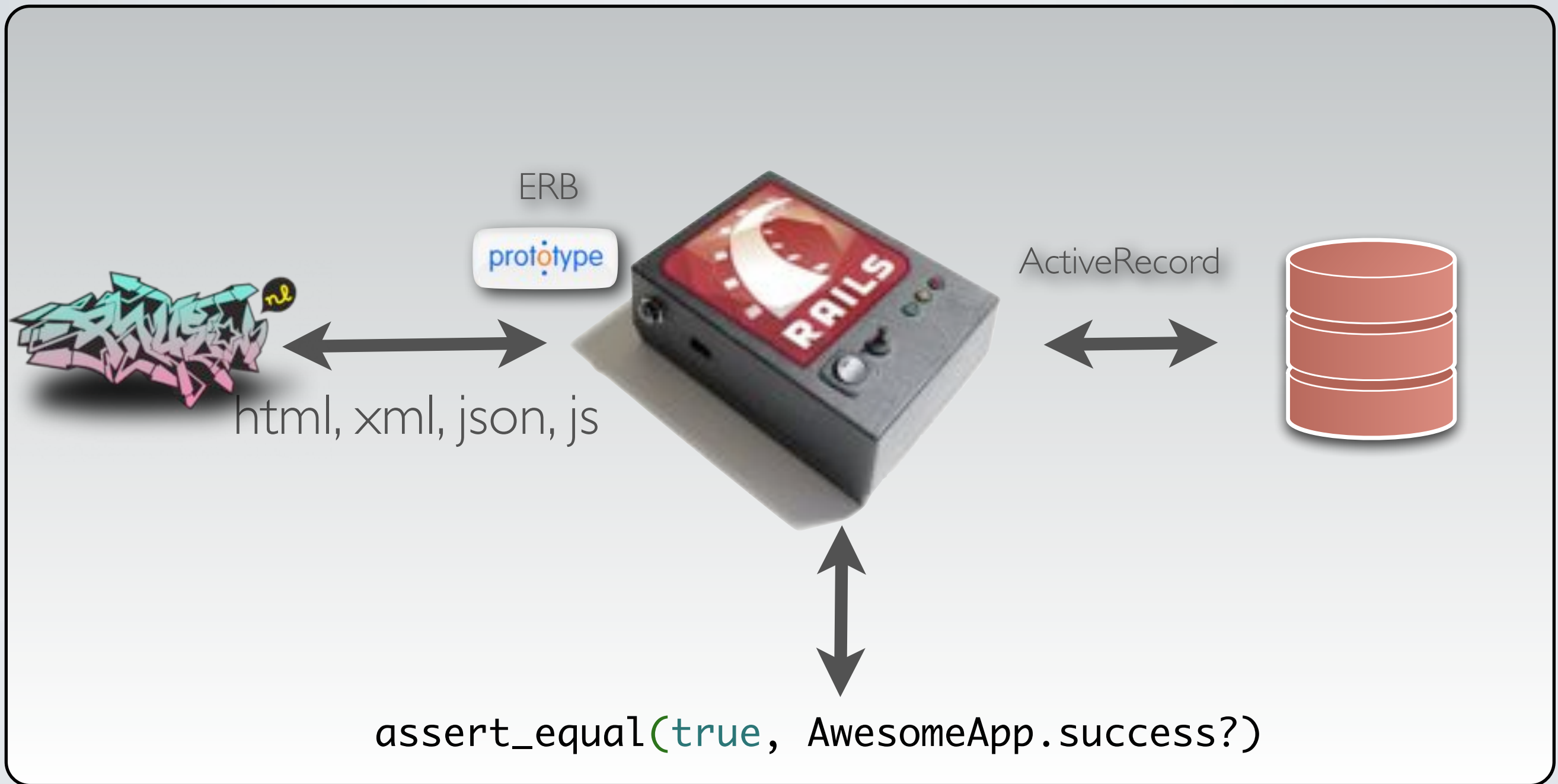
Rails 2

CECI N'EST PAS LA RÉALITÉ



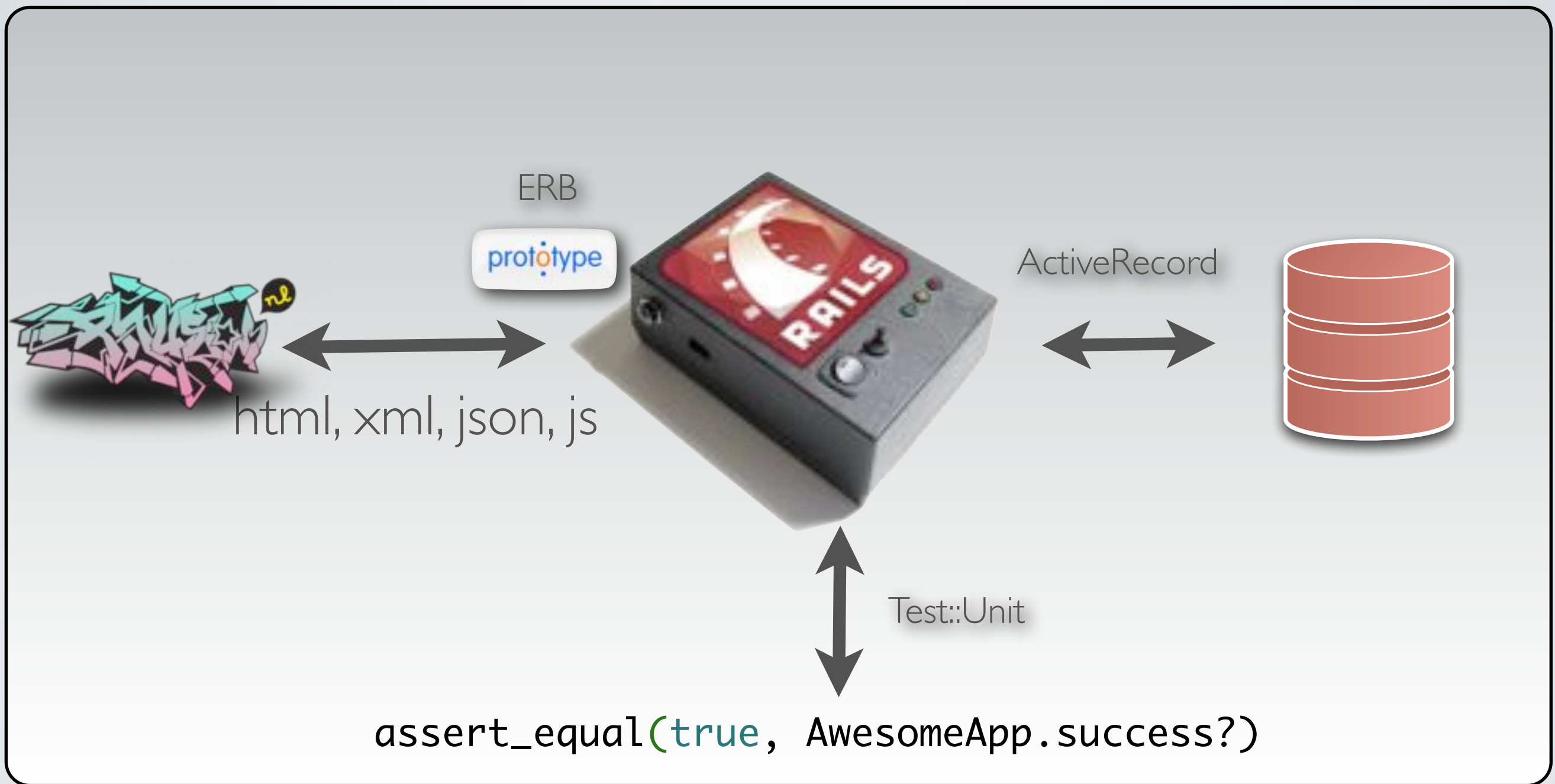
Rails 2

CECI N'EST PAS LA RÉALITÉ



Rails 2

CECI N'EST PAS LA RÉALITÉ

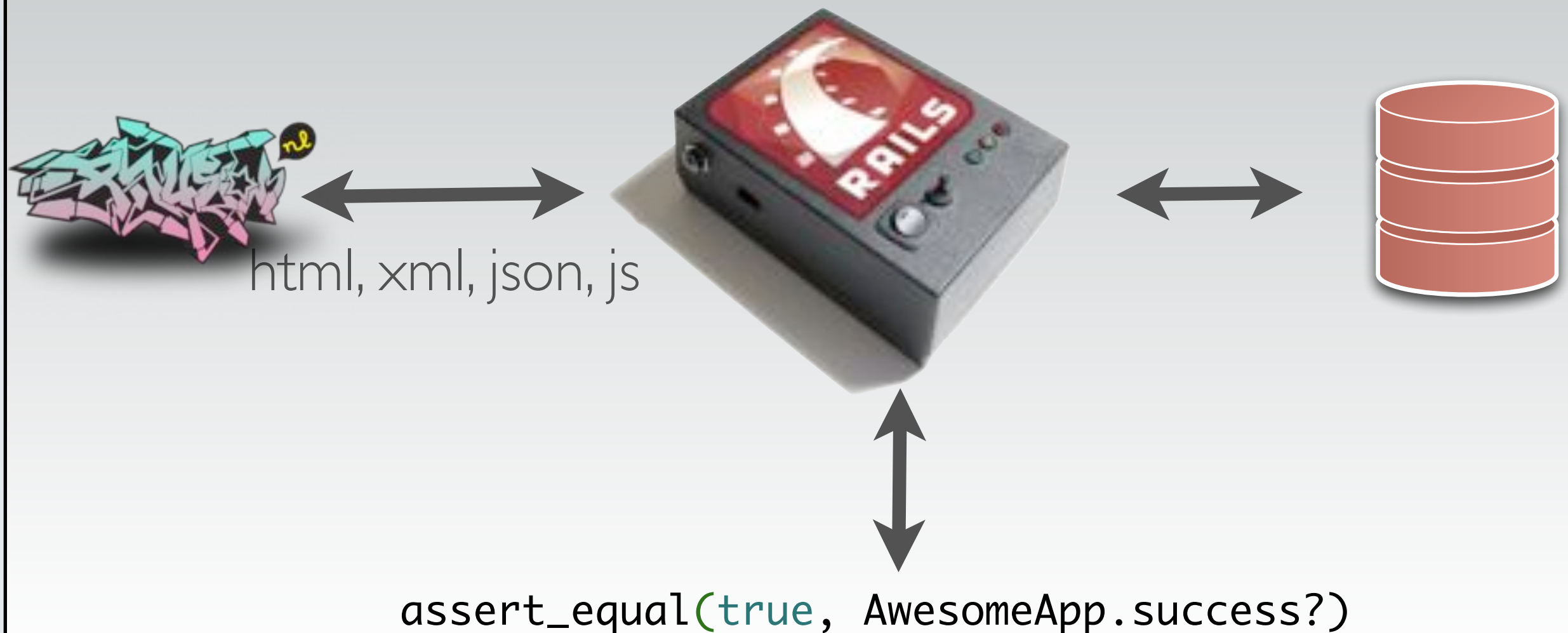


Rails 2

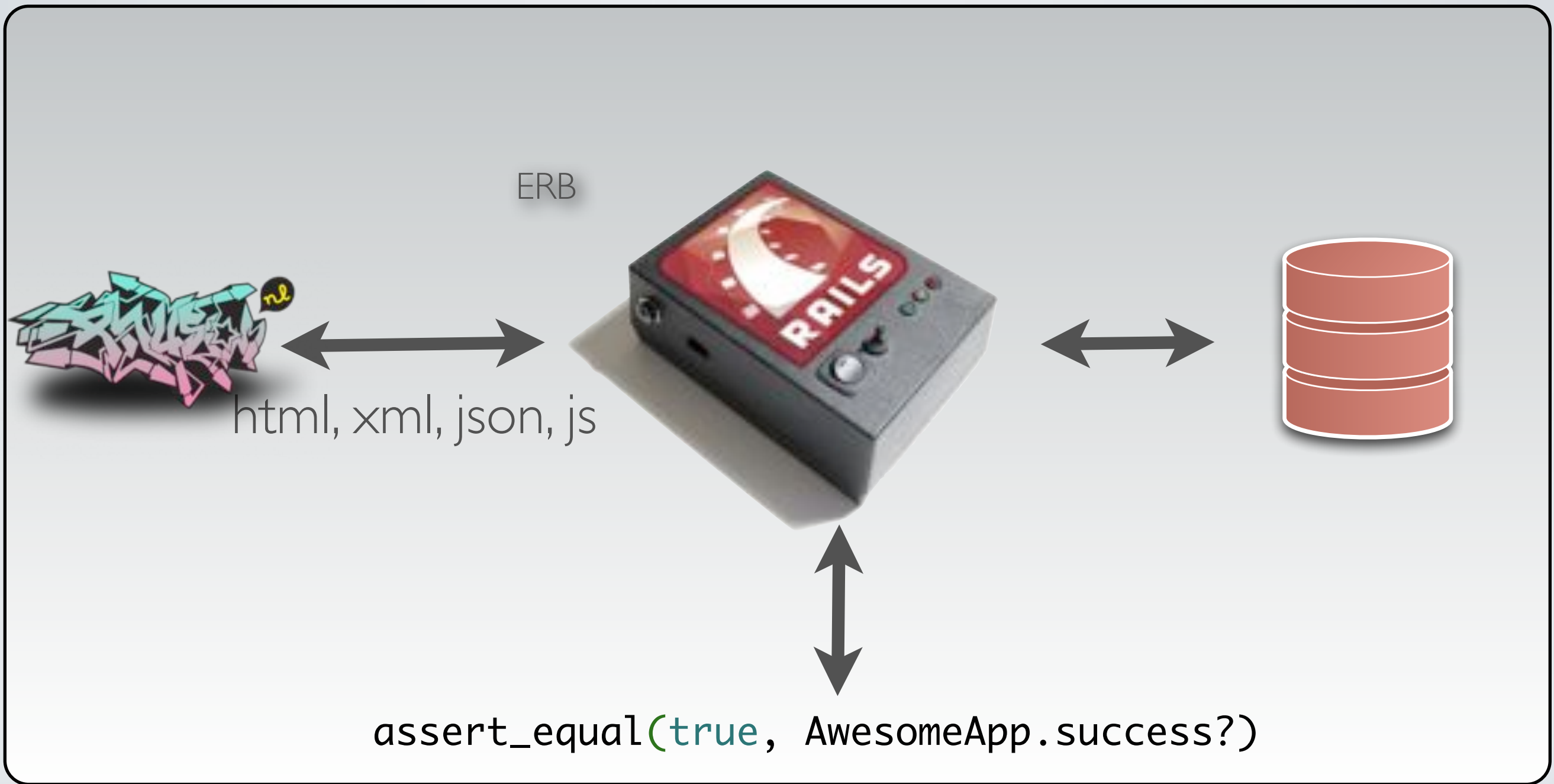
RAILS 3 DEFAULT STACK

- ORM: ActiveRecord
- JS: Prototype
- Templating: ERB
- Testing: Test::Unit

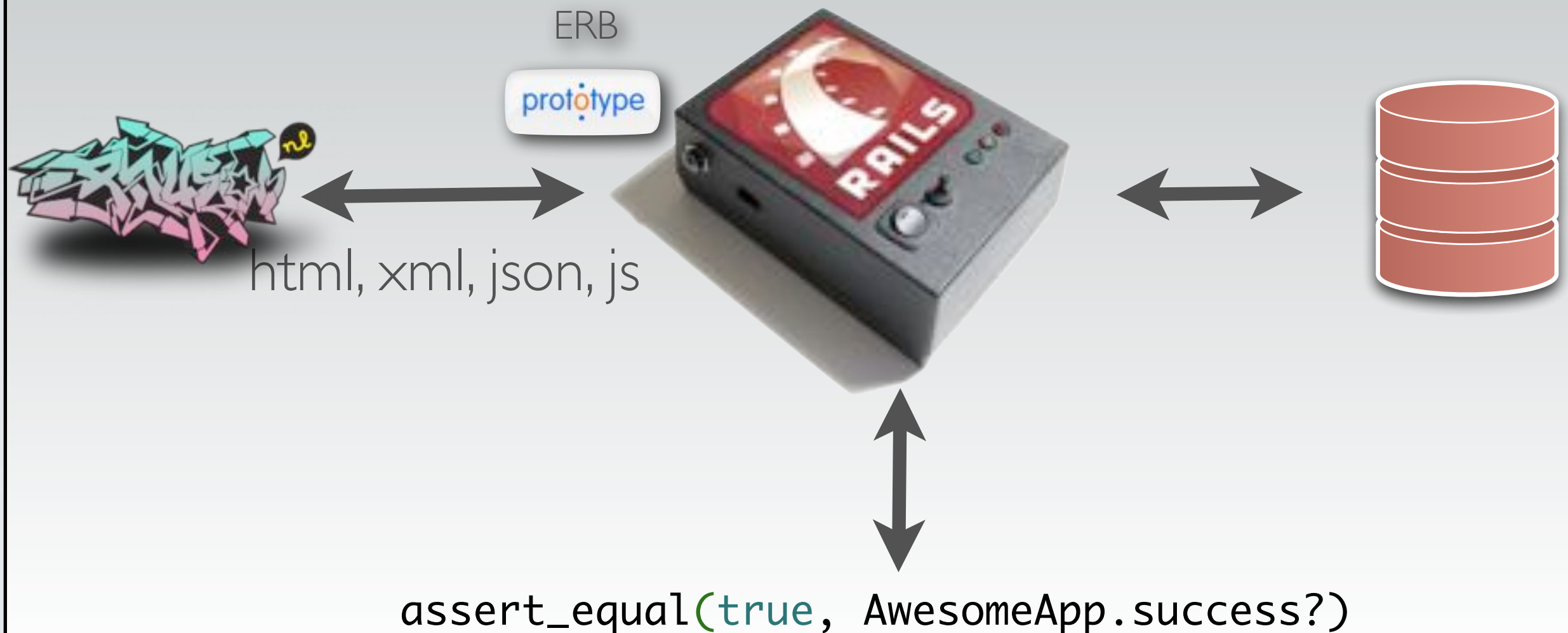
RAILS 3 - DEFAULT STACK



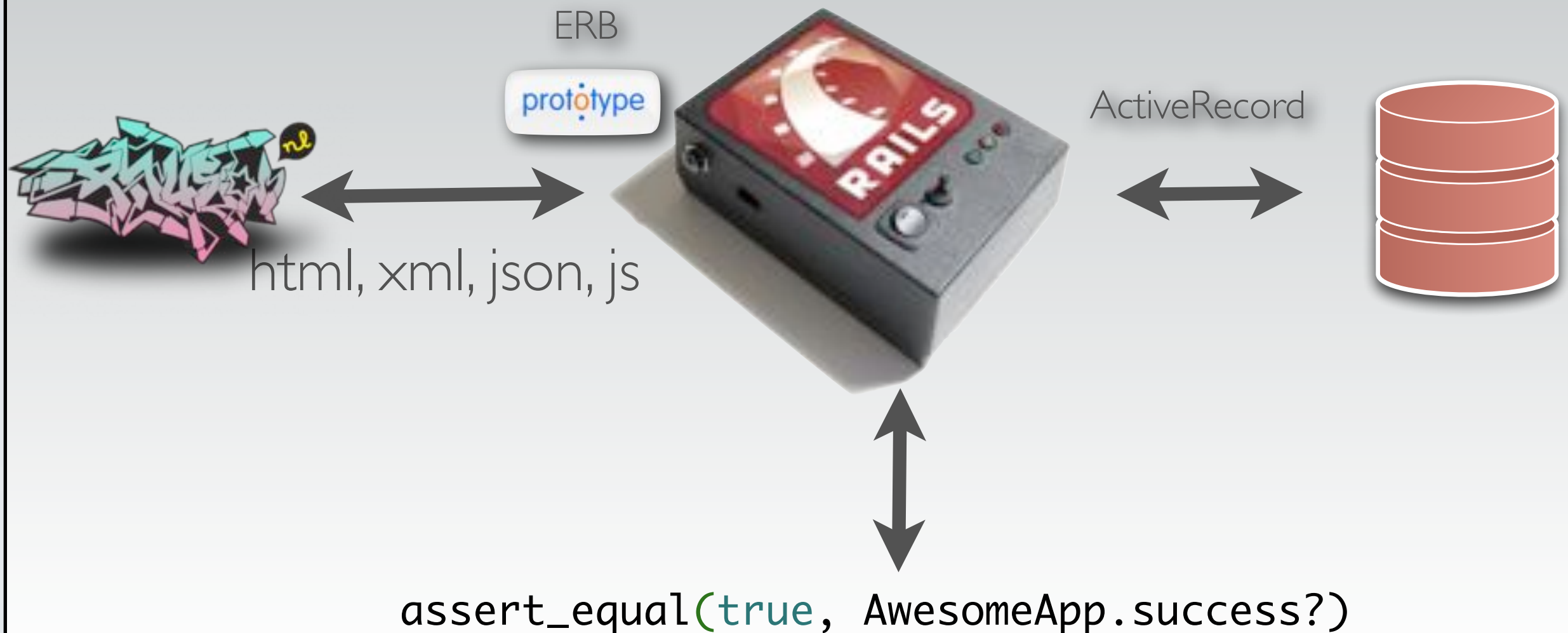
RAILS 3 - DEFAULT STACK



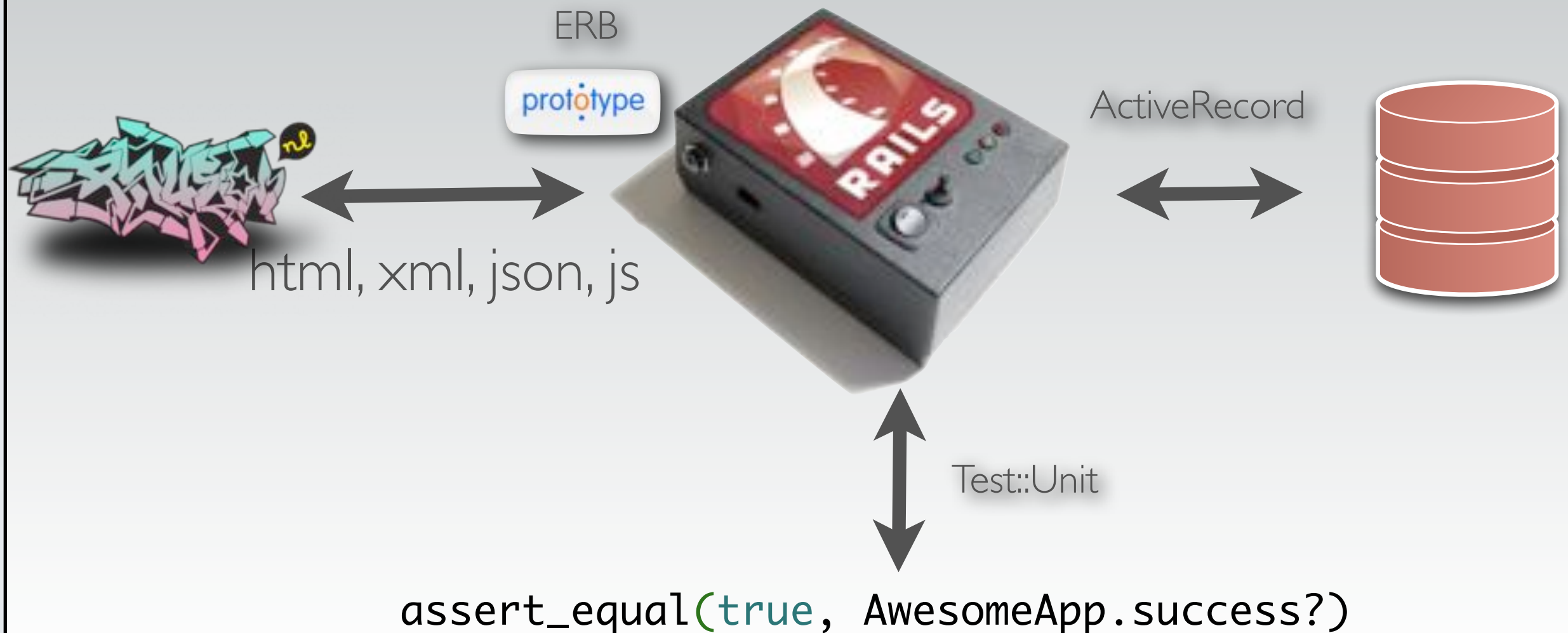
RAILS 3 - DEFAULT STACK



RAILS 3 - DEFAULT STACK



RAILS 3 - DEFAULT STACK



WHAAAT?

NO DRASTIC CHANGES FOR
THE END USERS

RAILS WAY

RAILS WAYS**S**









GAIN FLEXIBILTY



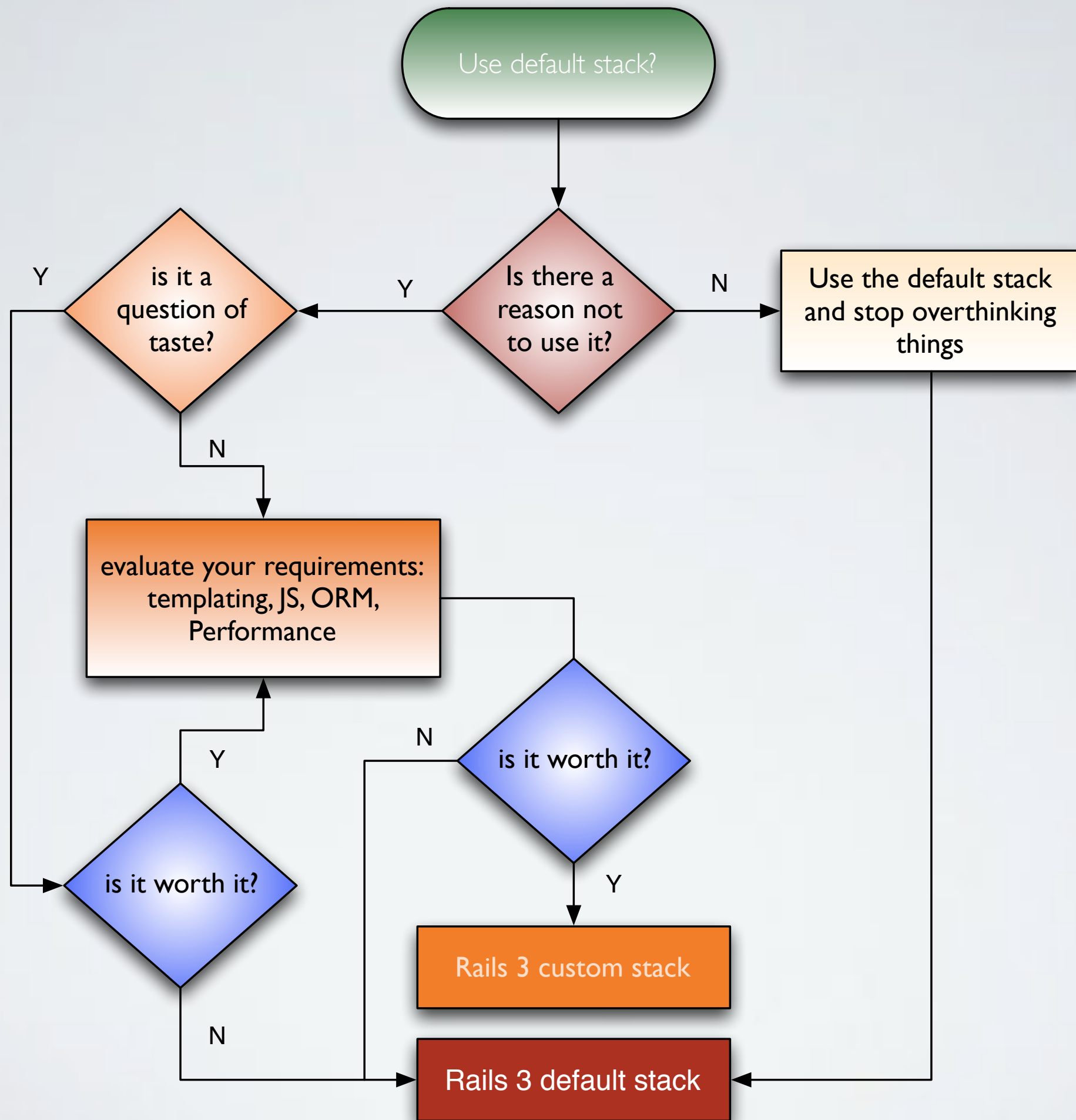
LESS OPINIONATED



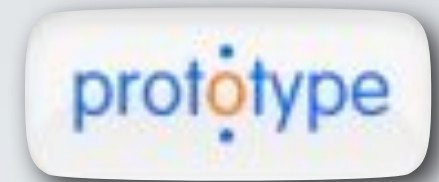
WE ARE NOT ALL THE SAME



¿WHEN?



OPTIONS



JAVASCRIPT FRAMEWORKS



TEMPLATING ENGINES

OBJECT RELATION MAPPER

ORM

Databases

scalar values

data



```
graph LR; Databases[Databases  
scalar values] <-->|data| OOL[objects  
Object Oriented  
Languages]; ORM[ORM];
```

objects

Object Oriented
Languages

RAW SQL

```
SELECT *  
FROM Book  
WHERE price > 100.00  
ORDER BY title;
```

ACTIVERECORD

```
Book.all \
  :conditions => ["price > ?", 100.00],
  :order      => 'title'
```

DATAMAPPER

```
Book.all \
  :price.gt => 100.00,
  :order    => [:title.asc]
```


SEQUEL

```
Book.find{|b| b.price > 100.00}.order(:title)
```

COUCHREST

```
Book.by_price(:startkey => 100.01)
```

MANY MORE ORMS/DSLS

RUBY IS AWESOME

DATAMAPPER

DATAMAPPER

```
@parent = Parent.find_by_name("Matt")

@parent.children.each do |child|
  @parent.object_id.should == child.parent.object_id
end
```

DATAMAPPER

```
@parent = Parent.find_by_name("Matt")

@parent.children.each do |child|
  @parent.object_id.should == child.parent.object_id
end
```

Fails with ActiveRecord

DATAMAPPER

```
@parent = Parent.find_by_name("Matt")

@parent.children.each do |child|
  @parent.object_id.should == child.parent.object_id
end
```

Fails with ActiveRecord

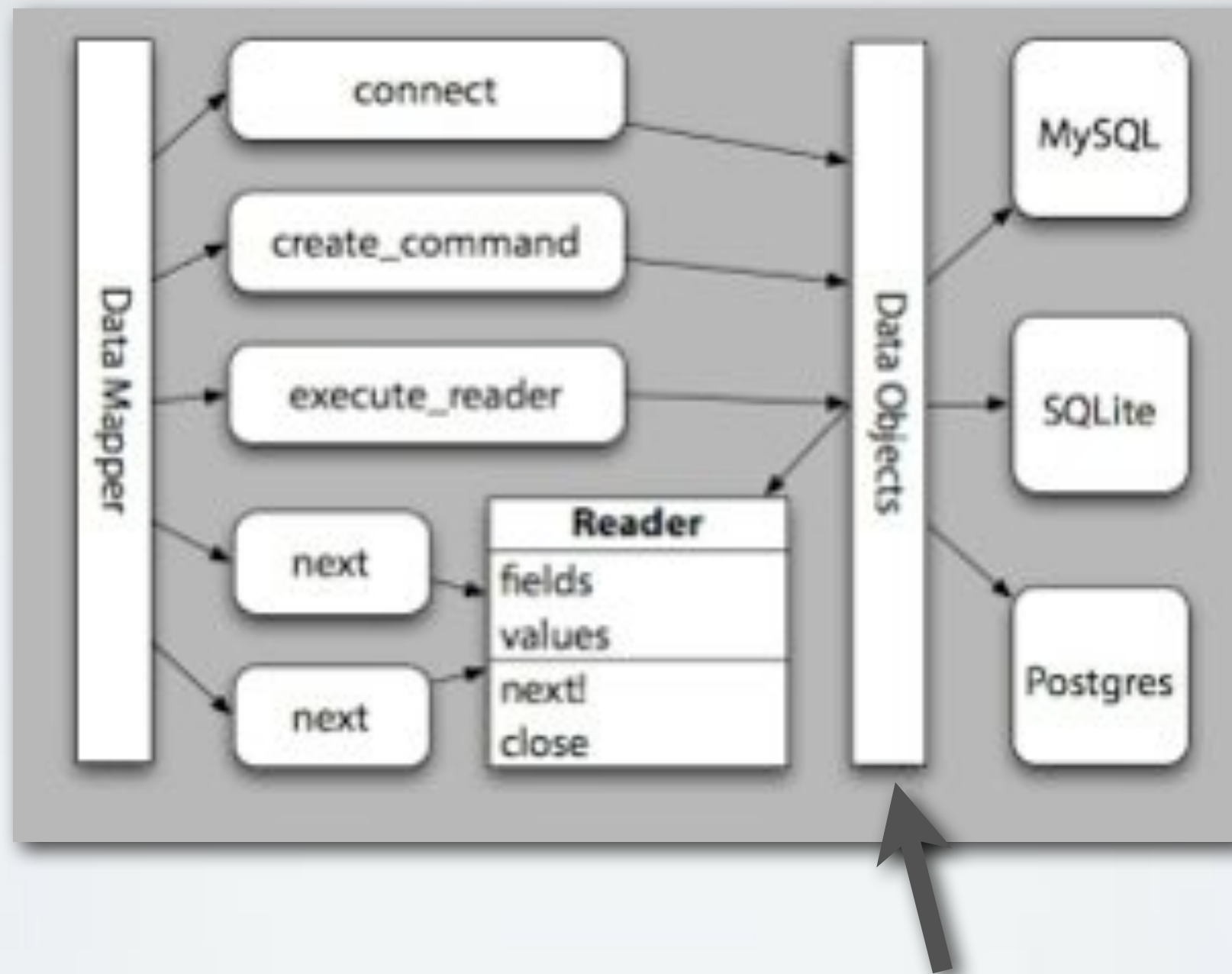
Passes with DataMapper

DATAMAPPER

does what ActiveRecord does...

but differently

DATAMAPPER



db drivers using | unified interface

DATAMAPPER

- procrastination as a value
- Lazy Loading
- Strategic Eager Loader

DATAMAPPER

```
students = Student.all
students.each do |student|
  student.books.map{|b| b.name}
end
```

ACTIVERECORD

```
students = Student.all
students.each do |student|
  student.books.map{|b| b.name}
end
```

```
SELECT * FROM "students"
SELECT * FROM "books" WHERE
      ("books".student_id = 1)
SELECT * FROM "books" WHERE
      ("books".student_id = 2)
SELECT * FROM "books" WHERE
      ("books".student_id = 3)
SELECT * FROM "books" WHERE
      ("books".student_id = X)
```

DATAMAPPER

```
students = Student.all
students.each do |student|
  student.books.map{|b| b.name}
end
```

```
SELECT "id", "name" FROM "students" ORDER BY "id"
SELECT "id", "name", "student_id" FROM "books" WHERE
  ("student_id" IN (1, 2, 3, X)) ORDER BY "id"
```

MULTIPLE REPOS

production:

```
adapter:  mysql
encoding: utf8
database: production-app
username: root
password: top-s3ckrit
host:     localhost
```

repositories:

nightly_backup:

```
adapter:  sqlite3
database: shared/nightly.db
```

weekly_backup:

```
adapter:  sqlite3
database: shared/weekly.db
```


MULTIPLE REPOS

```
Article.copy(:default, :nightly_backup,  
             :created.gt => 1.day.ago )
```

MULTIPLE REPOS

```
class Page
  include DataMapper::Resource

  property :id, Serial
  property :name, String

  repository(:legacy) do
    property :name, String,
              :field => "title"
  end
end
```

QUERY::PATH

Find all people
with an address that
has street in the street name

`Person.all("addresses.street.like" =>
"%street%")`

```
SELECT "people"."id", "people"."name" FROM "people"  
INNER JOIN "addresses" ON ("people"."id" = "addresses"."person_id")  
WHERE ("addresses"."street" LIKE '%street%')  
ORDER BY "people"."id"
```

ADAPTERS

RDBMS

file system

IMAP

Sales Force

YAML

REST APIs

custom APIs

...

SEQUEL

- Prepared Statements
- Sharding
- highly customizable SQL work
- high performance

HIBERNATE

- ActionORM
- JRuby

NON RDBMS SYSTEMS

- AppEngine::Datastore
- CouchDB with CouchRest or other DSL
- Redis, Tokyo Cabinet etc...

RAILS EXTREME

only for a few of you

CUSTOMIZE RAILS TO THE EXTREME



SproutCore
a javascript framework

YELLOWPAGES.COM™ yp.com



Adobe



**WIKIMEDIA
FOUNDATION**

CREATE YOUR OWN RAILS

- define your own file structure
- define your own router DSL
- define your own ways to deal with requests

EXTEND ACTIONCONTROLLER

```
class Presentation < ActionController::Http
  def index
    self.response_body = "Rails3 + MacRuby"
  end
end
```

```
Presentation.action(:index).call \
  Rack::MockRequest.env_for("/railsconf10")
```

EXTEND ACTIONCONTROLLER

```
def action(&blk)
  Class.new(ActionController::Http) do
    include ActionController::Renderer
    define_method(:xtreme, &blk)
  end.action(:xtreme)
end
```

```
run action do
  render "some_template"
end
```


¿FUTURE?

 **MacRuby**

+



+



CouchDB

credits:

<http://www.flickr.com/photos/dotdoubledot/>

Katsushika Hokusai

Delacroix

Benetton

<http://www.rubyist.net/~matz/slides/oscon2003>

Matt Groening

<http://www.flickr.com/photos/derricksphotos/>

<http://www.flickr.com/photos/bogdansuditu>

<http://www.flickr.com/photos/77096872@N00/2547160291/>

<http://www.flickr.com/photos/edvvc/1972546648/>

<http://www.flickr.com/photos/amagill/34762677/>

<http://www.flickr.com/photos/celinesphotographer/360918623/>

Thanks

Heidi Vega

Rails team

Yehuda Katz

Carl Lerche

Laurent Sansonetti

Rich Kilmer

Ninh Hernandez Bui

Hongli not-so-MacLovin' Lai

Dan Kubb