

Ruby On Rails

web development that doesn't hurt



astrails

Boris Nadion

astrails.com

Who am I?

Boris Nadion

19 years of experience in software engineering
working with Ruby on Rails since 2005

founder of **Astrails**

web development that
doesn't hurt

doesn't hurt businesses

doesn't hurt developers

software

software development
background

organic

you always find things you
want to change

great ideas

adaptive development

agile, scrum, xp, etc.

small iterations

continuous delivery

simplicity

tools?

what programmers need?

inspiration

challenge

compensation

do programmers
like programming?

programmers don't like
programming

we would be programming assembler still

solving complex problems in
beautiful ways

programming language
is a tool

happy

easy to learn

make problems easier to
solve

increase chance of success

that's why **Ruby on Rails**
programmers are **happy**

Ruby on Rails born to be
agile

extremely flexible

rapid development

focus on productivity

maintainable code

free

what is Ruby on Rails?

Ruby is a language



Yakihiro Matsumoto “matz”

1993

1995

free to use, copy, modify, and
distribute

POLS

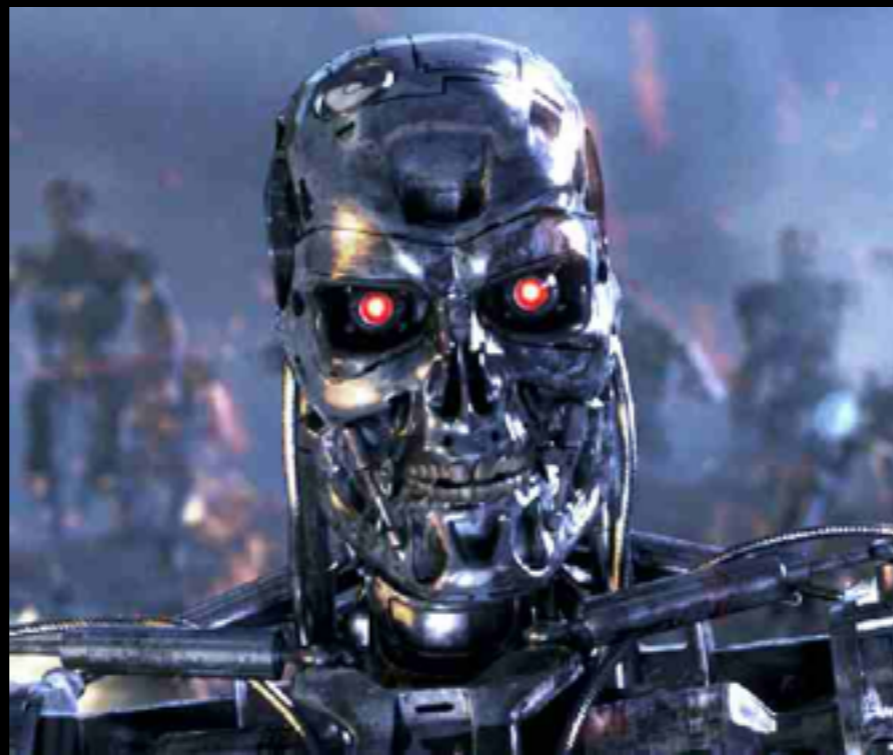
principle of least surprise

easy to read a code

guess how it works -
guess correctly

humans should easily
understand the code, **not**
computers

are you a human?



```
3.times { say "Hello" }
```

```
return if presentation.looks?(:boring)
```

```
["espresso", "cappuccino", "latte"].each do |drink|  
  print drink.capitalize  
end
```

almost pure English

less code

less code - less bugs

high maintainability

```

import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Date;

public class TimeServer {
    private static class TellTime extends Thread {
        private Socket soc;

        public TellTime(Socket soc) {
            super();
            this.soc = soc;
        }

        public void run() {
            try {
                this.soc.getOutputStream().write(
                    new Date().toString().getBytes());
            } catch (Exception e) {
            } finally {
                try {
                    this.soc.close();
                } catch (IOException e1) {
                }
            }
        }
    }

    public static void main(String args[]) throws Exception {
        ServerSocket server = new ServerSocket(12345);
        while (true) {
            new TellTime(server.accept()).start();
        }
    }
}

```

```

require "socket"

server = TCPServer.new(12345)

while (session = server.accept)
    Thread.new(session) do |my_session|
        my_session.puts Time.new
        my_session.close
    end
end

```

high signal to noise ratio

```

import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Date;

public class TimeServer {
    private static class TellTime extends Thread {
        private Socket soc;

        public TellTime(Socket soc) {
            super();
            this.soc = soc;
        }

        public void run() {
            try {
                this.soc.getOutputStream().write(
                    new Date().toString().getBytes());
            } catch (Exception e) {
            } finally {
                try {
                    this.soc.close();
                } catch (IOException e1) {
                }
            }
        }
    }

    public static void main(String args[]) throws Exception {
        ServerSocket server = new ServerSocket(12345);
        while (true) {
            new TellTime(server.accept()).start();
        }
    }
}

```

```

require "socket"

server = TCPServer.new(12345)

while (session = server.accept)
    Thread.new(session) do |my_session|
        my_session.puts Time.new
        my_session.close
    end
end

```

less code - less effort

high productivity



credit: <http://www.flickr.com/photos/slworking/>

Rails is a **framework**



David Heinemeier Hansson

2003

MIT license

opinionated

convention over
configuration

no XML configuration files

default places and names for
files, tables, etc

you can override everything

```
c = Company.first  
c.name
```

```
app/models/company.rb
```

```
class Company < ActiveRecord::Base  
end
```

```
select * from companies limit 1;
```

MVC

Model - View - Controller

concerns separation

MVC

business logic in models

fetch the data



credit: <http://www.flickr.com/photos/simonpais/>

adapters

MySQL, SQLite, PostgreSQL, MS SQL, Oracle, DB2,
MongoDB, Cassandra, ...

display logic in **views**

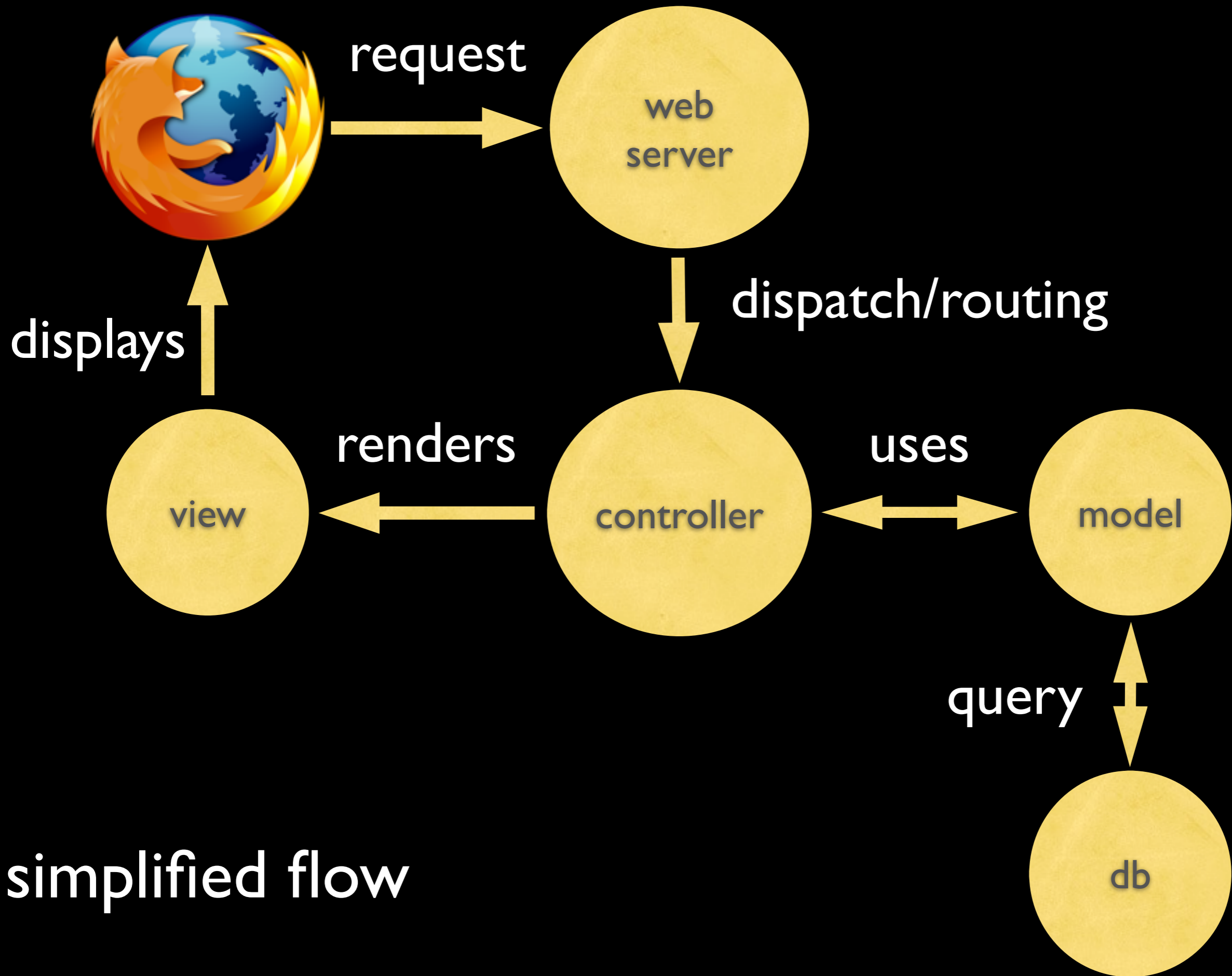
render the data

templates

ERB, HAML, Liquid, Amrita, Markaby, ...

user logic in controllers

handle requests



testing

testing frameworks

test::unit, rspec, micronaut, cucumber, shoulda, ...

TDD

http://en.wikipedia.org/wiki/Test-driven_development

BDD

http://en.wikipedia.org/wiki/Behavior_Driven_Development

TAFT

test all the f\$cking time

ruby gems and rails plugins

bottom line

what do you **do next?**

<http://groups.google.com/group/israelrb>

learn *Ruby* on Rails

learn the best practices

apply it to your
next killer project

it won't hurt

web development that
doesn't hurt

Ruby on Rails

perfect for web startups

optimized for programmer
happiness and sustainable
productivity

Q&A

does it scale?

yes

Ruby On Rails

web development that doesn't hurt

Thank you!

Q&A



astrails

Boris Nadion

astrails.com