

FANTASTIC JVM LANGUAGES

AND WHERE TO FIND THEM

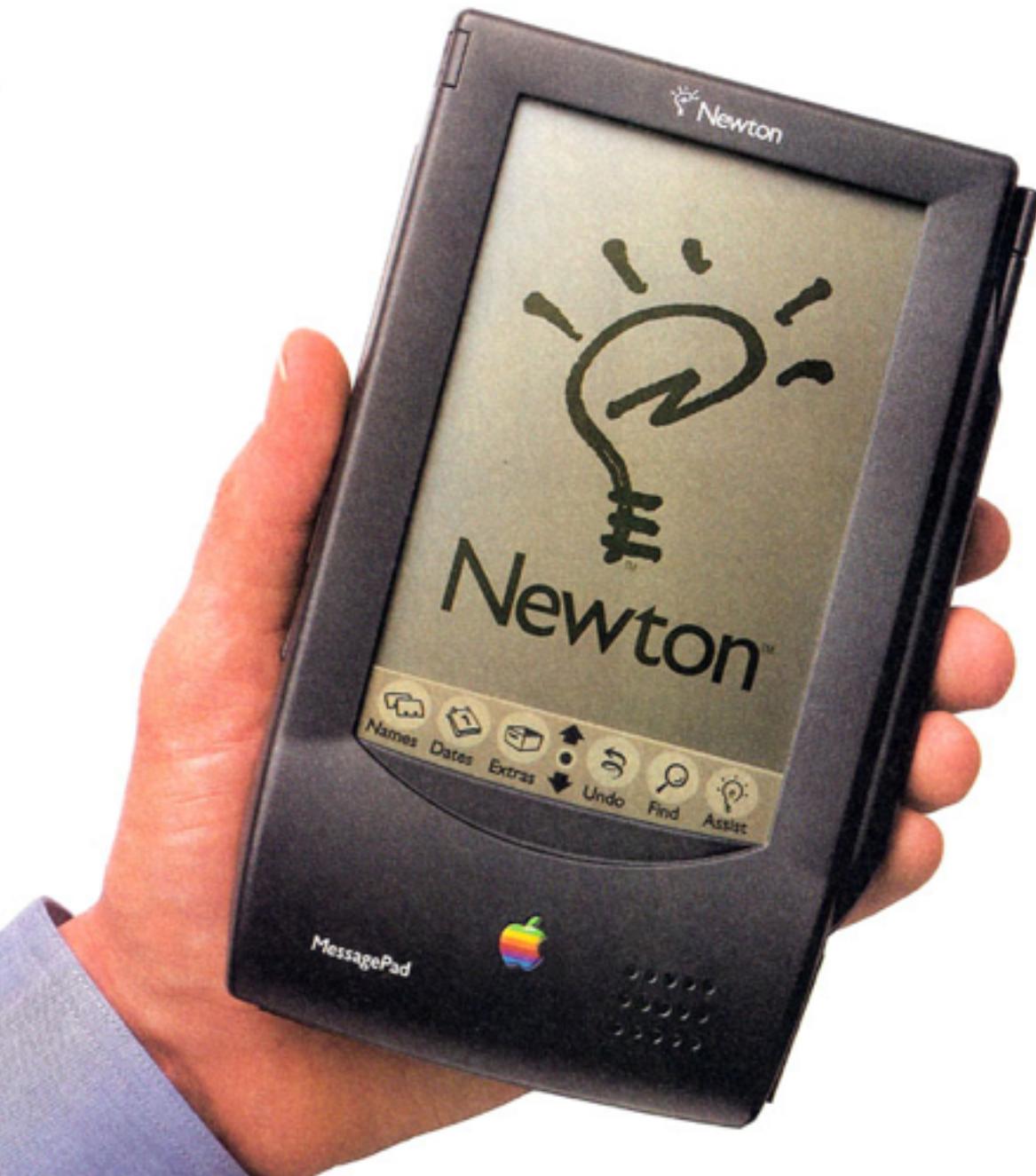
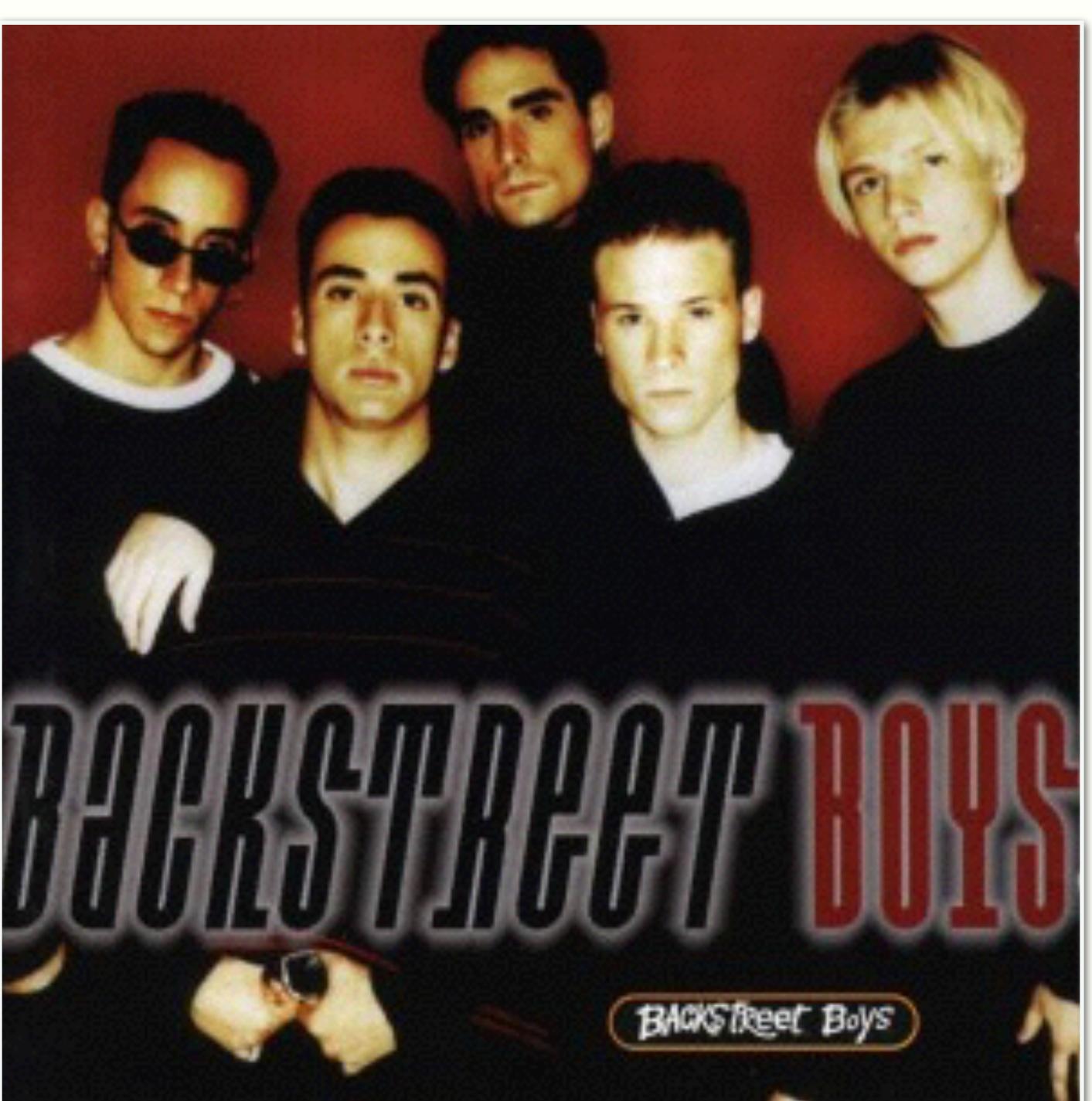
BRANDEN BELLANCA **ABOUT ME**

TECHNICAL QUALITY
ASSURANCE ANALYST AT
NETJETS

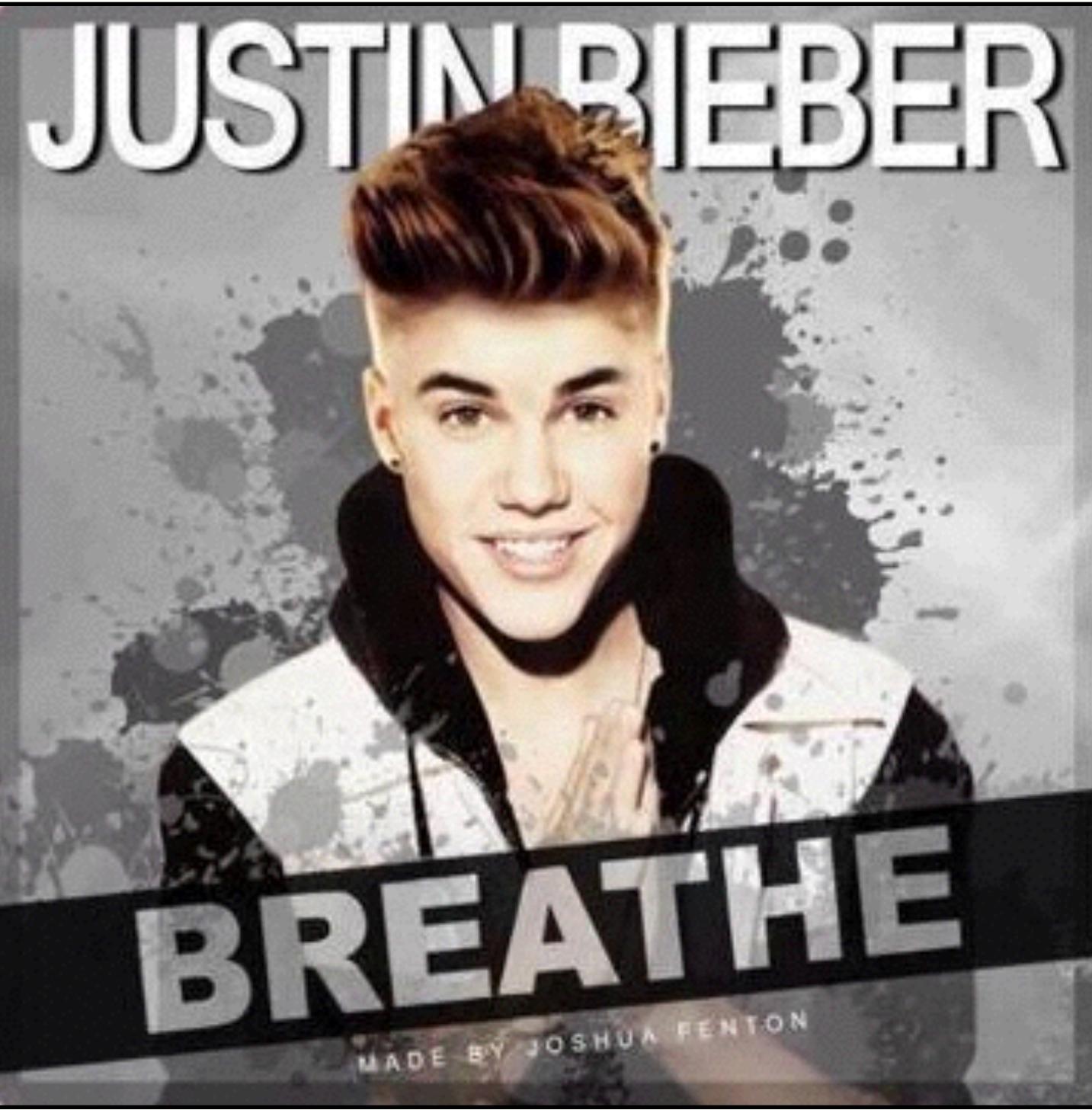
@BSBELLANCA
/BELLANC

CIRCA 1996

(WHEN JAVA WAS BORN)



**BUT THE WORLD
HAS CHANGED**



Yes, this is
the guy
from
Beverly
Hills 90210
in
Sharknado



MOST PEOPLE TALK ABOUT **JAVA** THE LANGUAGE,
AND THIS MAY SOUND ODD COMING FROM ME, BUT I
COULD HARDLY CARE LESS. **AT THE CORE OF THE**
JAVA ECOSYSTEM IS THE JVM.

JAMES GOSLING

CREATOR OF THE JAVA PROGRAMMING LANGUAGE (2011, THE SERVERSIDE)

THE JAVA VIRTUAL MACHINE [JVM]

- **PORTABLE**
 - Runs programs compiled into java byte code
 - Allows for unprecedented portability
 - WORA (Write Once, Run Anywhere)
- **FERTILE PLATFORM**
 - People who want to build their own JVM or build a language on the existing JVM standard can do so
- **VERSATILE**

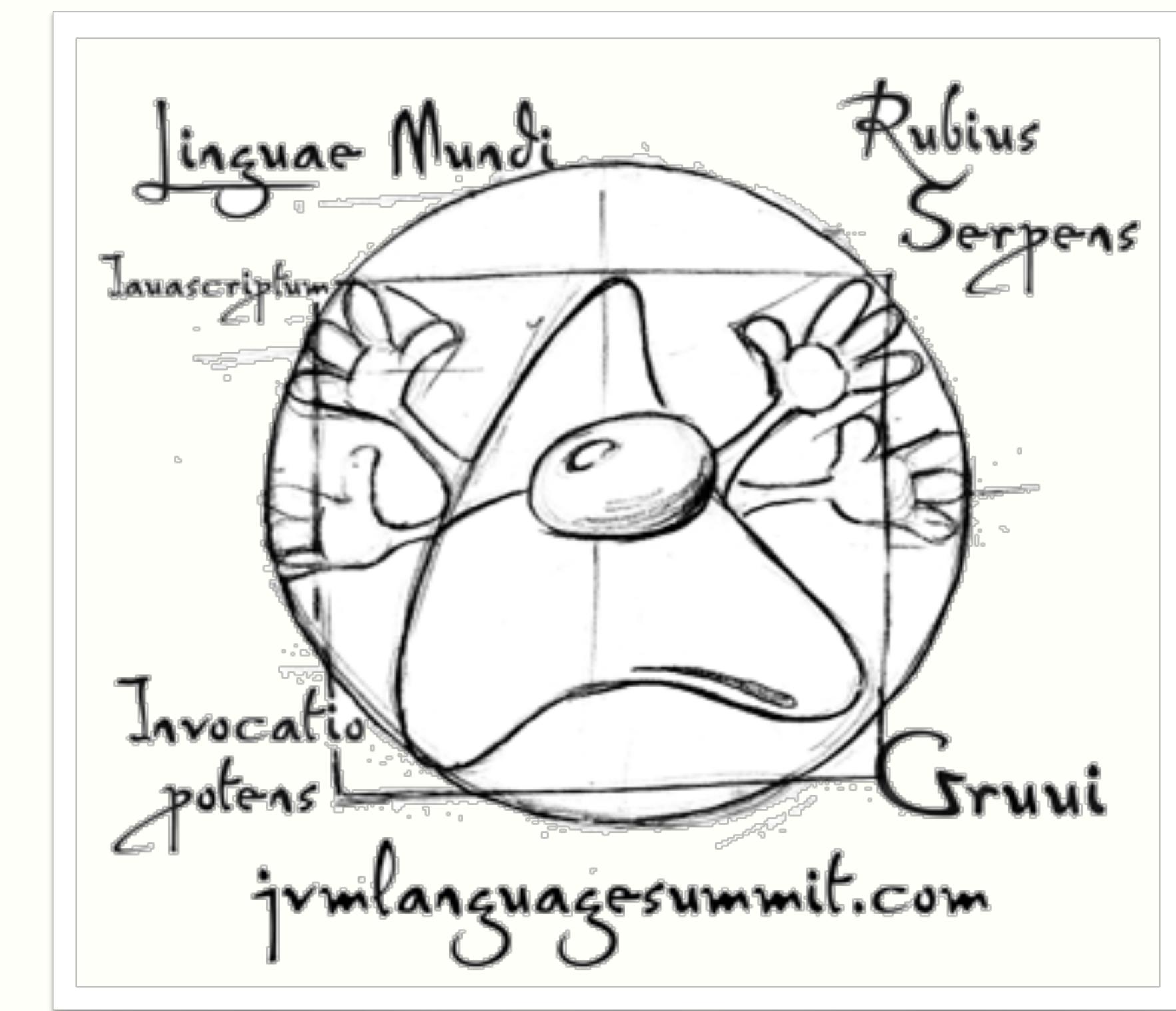
THE SHIFTING TIDE

- ▶ After 20 years, Java is starting to show its age
- ▶ The industry has realized the need, and potential, for more options on the JVM
- ▶ Now over **200(!)** languages for the JVM
- ▶ With so many choices, how do developers decide what's the right tool for the job?



5 LANGAUGES IN 60 MINS(ISSH)

- JRUBY
- CLOJURE
- SCALA
- GROOVY... ISH
- JAVASCRIPT (NASHORN)





ELEGANT | 'ELəGəNT |

ADJECTIVE

PLEASINGLY GRACEFUL AND

STYLISH IN APPEARANCE OR

MANNER:

she will look elegant in black | an elegant, comfortable house

JRUBY: THE GEM OF THE JVM

- ▶ JRuby == Ruby++
- ▶ All the elegance of Ruby, with Java Libraries and APIs
- ▶ Gain the agility of using a dynamic scripting language
- ▶ Full two-way access between the Java and the JRuby code
- ▶ RAILS!



JAVA

```
class Animal{  
    private String name;  
    public Animal(String name){  
        this.name = name;  
    }  
    public String getName(){  
        return name;  
    }  
    public String setName(String name){  
        this.name = name;  
    }  
    public void saySomething(){  
        System.out.println("I am " + name);  
    }  
}  
  
class Dog extends Animal{  
    public Dog(String name) {  
        super(name);  
    }  
    public void saySomething(){  
        System.out.println("I can bark");  
    }  
}  
  
public class Main {  
    public static void main(String[ ] args) {  
        Dog dog = new Dog("Beagle");  
        dog.saySomething();  
    }  
}
```

JRUBY

```
class Animal  
  attr_accessor :name  
  
  def initialize(name)  
    self.name = name  
  end  
  def say_something  
    puts "I am #{self.name}"  
  end  
end  
  
class Dog < Animal  
  def say_something  
    puts "I am #{self.name}, and I can bark"  
  end  
end  
  
dog = Dog.new("Beagle")  
dog.say_something
```

**Less ceremony,
more elegant code**

USING JAVA WITH JRUBY

WRITE A CLASS WITH JAVA...

```
public class Beach {  
  
    private String name;  
    private String city;  
  
    public Beach(String name, String city){  
        this.name = name;  
        this.city = city;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getCity() {  
        return city;  
    }  
  
    public void setCity(String city) {  
        this.city = city;  
    }  
}
```

EXECUTE WITH JRUBY

```
require 'java'  
java_import 'Beach'  
beach = Beach.new('ClearwaterBeach','Clearwater')  
puts beach.getName()
```

WHO'S USING JRUBY?



Oslo Airport Gardermoen - Fleet Control
(monitors and controls the passenger
busses and fuel trucks)



Best Buy - Best Buy Developers' Portal



Danish Technological University - Library



University of Minnesota - Minnesota
Population Center

INTO THE WILD: **LEGACY JAVA CODE AND JRUBY**

JAVA SWEETENED WITH JRUBY

- ▶ Java is everywhere, and isn't going away anytime soon
- ▶ Many Enterprise applications exist already written in Java
- ▶ JRuby allows you to use your... *cough* favorite Java EE applications with all the joy of Ruby
- ▶ require complied Java classes individually or just require whole jars





Clojure

I SET OUT TO CREATE A LANGUAGE TO ONLY DEAL WITH THE PROBLEMS I WAS CONTENDING WITH IN JAVA AND C# FOR THE KINDS OF APPLICATIONS THAT I WRITE, WHICH ARE BROADCAST AUTOMATION AND SCHEDULING AND ELECTIONS SYSTEMS AND THINGS LIKE THAT, WHERE THERE IS A LOT OF CONCURRENCY. I FOUND JUST OBJECT ORIENTED PROGRAMMING AND THEIR APPROACHES TO CONCURRENCY OF THOSE LANGUAGES IS JUST NOT GOOD ENOUGH FOR THOSE KINDS OF PROBLEMS - THEY ARE TOO HARD. I'M A BIG FAN OF LISP AND THE OTHER FUNCTIONAL LANGUAGES AND WHAT I WANT TO DO IS SOLVE THOSE PROBLEMS, MAKE A PRACTICAL LANGUAGE, NOT TO HAVE TO PROGRAM IN JAVA ANYMORE...

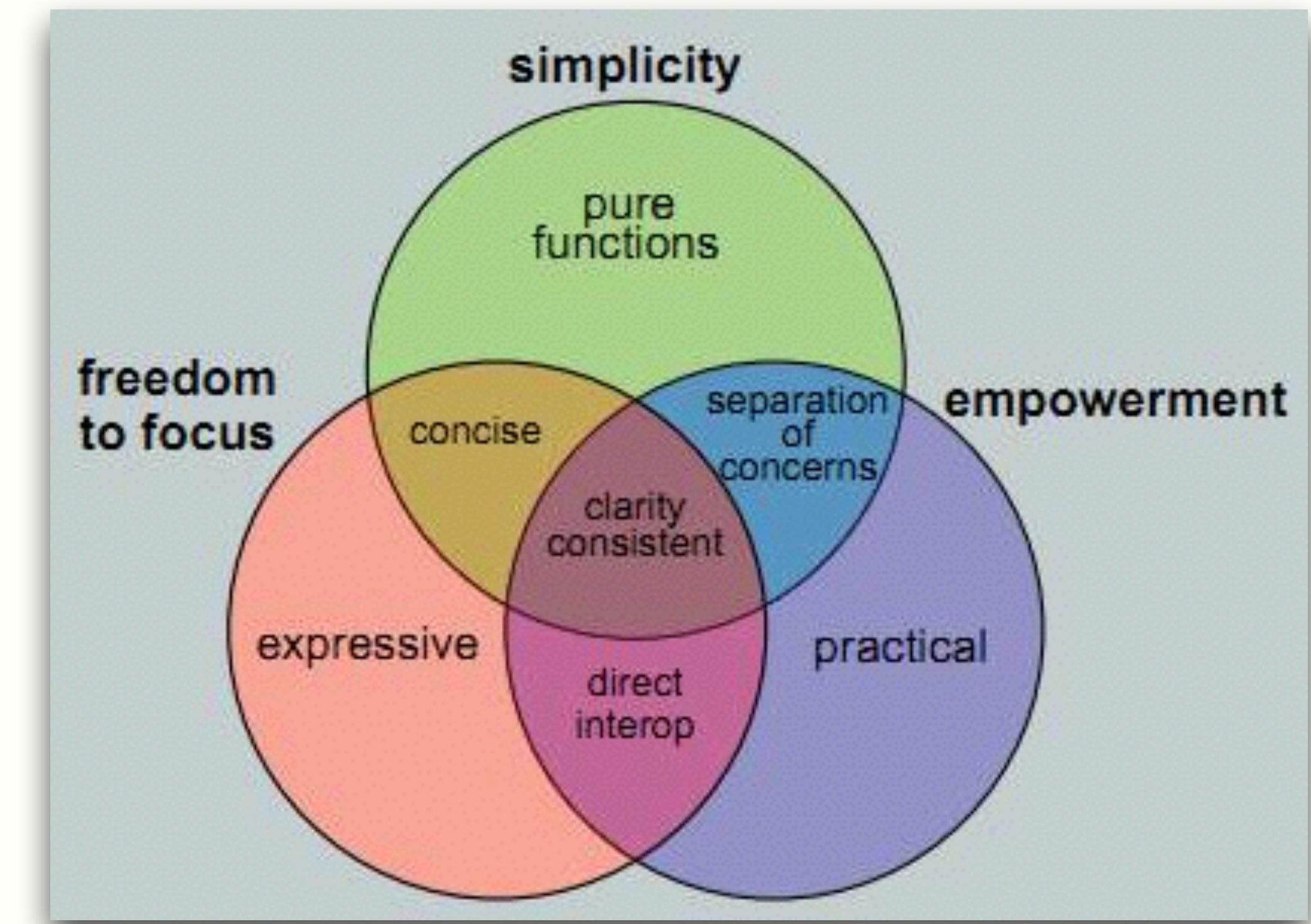
BASICALLY BECAUSE I WANTED A LISP FOR FUNCTIONAL PROGRAMMING, SYMBIOTIC WITH AN ESTABLISHED PLATFORM, DESIGNED FOR CONCURRENCY AND COULDN'T FIND ONE

RICH HICKEY

CREATOR OF CLOJURE, IN A 2009 INFOQ INTERVIEW

CLOJURE: A MODERN LISP

- ▶ Dynamically typed, purely functional language
- ▶ Addresses the issues that held Lisp back (libraries, readability, performance) while preserving its virtues
- ▶ Provides a rich set of immutable, persistent data structures (including hashmap, sets and lists).
- ▶ Inter-operate with Java with extreme ease; Java programs can call Clojure, and Clojure can call Java



WHY CLOJURE?

- **PURELY FUNCTIONAL**
- **CONCISE**
- **HOMOICONIC**

- More concise, Less code
- Homoiconicity
- Everything is a data structure
 - syntax is much simpler and more abstract constructs can be built in term of primitives. It also make it easier for macro programming
- Concurrency
- REPL (Read–eval–print loop)

SOME BASICS

- ▶ Hello World!

```
(println "Hello world!")
```

- ▶ Defining a function

```
(defn square [x]  
  (* x x))
```

- ▶ Printing first item in a list

```
(first '("one" "two" "three")) => "one"
```

- ▶ Conditional

```
(= :a (if (false? (= 4 5)) :a :b))
```

HOW MUCH MORE CONCISE IS CLOJURE THAN JAVA?

CAN YOU QUICKLY TELL ME WHAT THIS METHOD IS SUPPOSE TO DO?

```
public static boolean isAllWhitespace(String str) {  
    int strLen;  
    if (str == null || (strLen = str.length()) == 0) {  
        return true;  
    }  
    for (int i = 0; i < strLen; i++) {  
        if (((Character.isWhitespace(str.charAt(i)) == false)) {  
            return false;  
        }  
    }  
    return true;  
}
```

```
(defn is-all-whitespace [input-str]  
  (every? (fn [char] (= " " char)) input-str))
```

HOW DOES JAVA HANDLE PROBLEMS LIKE CONWAY'S GAME OF LIFE?

CONWAY'S GAME OF LIFE: CLOJURE

```
(defn neighbours [[x y]]
  (for [dx [-1 0 1] dy (if (zero? dx) [-1 1] [-1 0 1])]
    [(+ dx x) (+ dy y)]))

(defn step [cells]
  (set (for [[loc n] (frequencies (mapcat neighbours cells))
             :when (or (= n 3) (and (= n 2) (cells loc)))]
         loc)))

(def board #{[1 0] [1 1] [1 2]})
; #'user/board
(take 5 (iterate step board))
; #{{[1 0] [1 1] [1 2]} #{[2 1] [1 1] [0 1]} #{[1 0] [1 1] [1 2]} #{[2 1] [1 1] [0 1]} #{[1 0] [1 1] [1 2]}})
```

WHO'S USING CLOJURE?



Citibank is using Clojure to build front-office trading systems in its global credit group



The language is helping **Akamai** to adapt its global content delivery network (CDN) to the burgeoning mobile market



An outfit called **FlightCaster** is writing algorithms in Clojure that "scour data on every domestic flight for the past 10 years and match it to real-time conditions" as a way to predict flight delays hours before the airlines bother to alert travel

INTO THE WILD: HOPLON & CLOJURESCRIPT

HOPLON: CODE THE WEB IN CLOJURE

- ▶ Web framework for Clojure
- ▶ Everything is written in Clojure and ClojureScript, clientside and serverside
- ▶ “Single-page applications, not documents”
- ▶ Decouples business logic from HTTP-related concerns

HTML

```
<script type="text/hoplon">
(page "foo/bar.html"
  (:require [my.lib :as lib]))

(defc clicks 0)
</script>

<html>
<head>
<title>Hello World</title>
<link rel="stylesheet" href="css/main.css">
</head>
<body>
<lib.heading-large>Hello, world!</lib.heading-large>
<p><text>You've clicked ~{clicks} times.</text></p>
<button on-click="{{ #(swap! clicks inc) }}">
Click me!
</button>
</body>
</html>
```

CLOJURESCRIPT

```
(page "foo/bar.html"
      (:require [my.lib :as lib]))

(defc clicks 0)

(html
  (head
    (title "Hello World")
    (link :rel "stylesheet" :href "css/main.css"))
  (body
    (lib/heading-large "Hello, world!")
    (p (text "You've clicked ~{clicks} times."))
    (button :on-click #(swap! clicks inc) "Click me!")))
```

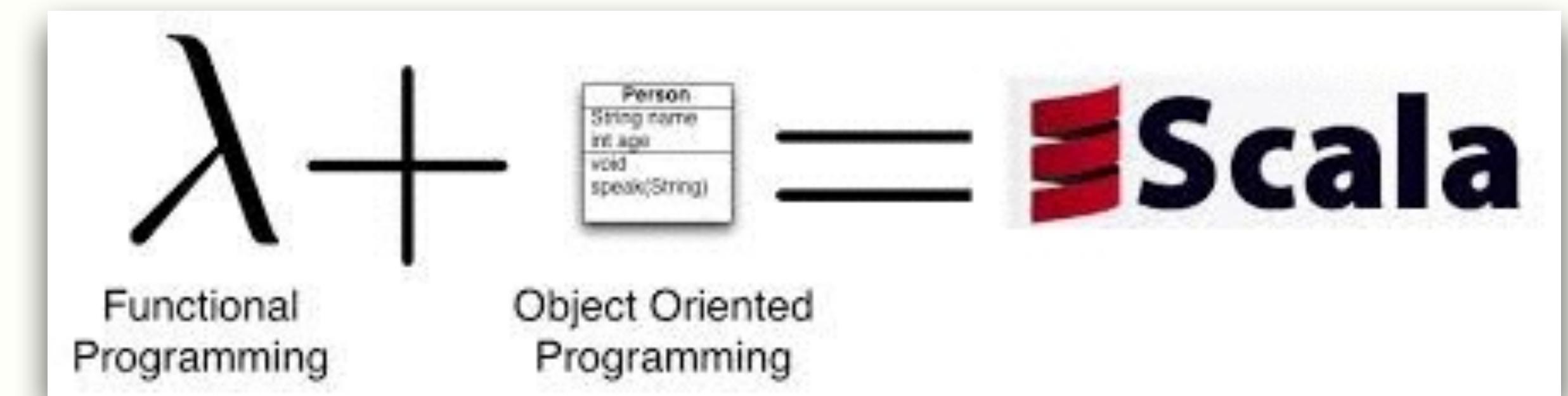


KEEP
CALM
AND
(USE
'CLOJURE)



SCALA: A “SCALABLE”, CONCURRENT JAVA

- ▶ Designed by Martin Odersky, designer of Sun’s Java compiler
- ▶ statically typed programming language that fuses the object-oriented model and functional programming ideas
- ▶ Support for fully functional programming (anonymous functions, currying, lazy initialization, etc) allowing for smaller and more concise programs
- ▶ Focus on concurrency
- ▶ Can be interpreted or compiled



WHY SCALA?

- **FUNCTIONAL & OBJ ORIENTED**
- **JAVA-LIKE**
- **ACTORS**

- Allows a gradual, easy migration to a more functional style
- Less verbose
- Better concurrency & immutable data structures
- Java and Scala classes can be freely mixed
- Scala projects -> are just jars
- Great Web Support

Play

Lift

Scalatra

SOME BASICS

- ▶ Hello World(ish)!

```
println("Hello, Scala!");
```

- ▶ Iteration through a list this way...

```
var myArray = Array('a', 'b', 'c', 'd', 'e')
for(myString <- myArray) {
    println(myString);
}
```

- ▶ ... or this way

```
List(1, 2, 3) foreach println _
```

- ▶ Hashes

```
val map = scala.collection.mutable.HashMap.empty[Int, String]
map += (1 -> "make a website")
map += (3 -> "???")
map += (4 -> "profit!")
map(1)
res44: String = make a web site
map contains 2
res46: Boolean = false
```

- ▶ Placeholder Syntax

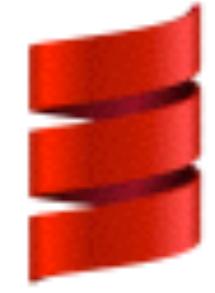
```
List(1, 2, 3) map (_ + 2)
```

- ▶ Default Params

```
def myfunct(elems: List[Int], x: Int = 0, cond: Boolean = true) {
    ...
}
myfunct(List(1))
f(Nil, cond = false)
```

- ▶ Loops

```
for(i <- 0 until myArray.length){
    println("i is: " + i);
    println("i'th element is: " + myArray(i));
}
```



Scala

```
def products = orders.flatMap(o => o.products)
```

```
public List<Product>getProducts() {
    List<Product> products = new ArrayList<Product>();
    for (Order order : orders) {
        products.addAll(order.getProducts());
    }
    return products;
}
```



JAVA

```
import java.net.*;
import java.io.*;
import java.util.*;

public class WordsOfEqChars {
    public static void main(String[] args) throws IOException {
        URL url = new URL("http://www.puzzlers.org/pub/wordlists/unixdict.txt");
        InputStreamReader isr = new InputStreamReader(url.openStream());
        BufferedReader reader = new BufferedReader(isr);

        Map<String, Collection<String>> anagrams = new HashMap<String, Collection<String>>();
        String word;
        int count = 0;
        while ((word = reader.readLine()) != null) {
            char[] chars = word.toCharArray();
            Arrays.sort(chars);
            String key = new String(chars);
            if (!anagrams.containsKey(key))
                anagrams.put(key, new ArrayList<String>());
            anagrams.get(key).add(word);
            count = Math.max(count, anagrams.get(key).size());
        }

        reader.close();

        for (Collection<String> ana : anagrams.values())
            if (ana.size() >= count)
                System.out.println(ana);
    }
}
```

SCALA

```
val src = io.Source.fromURL("http://www.puzzlers.org/pub/wordlists/unixdict.txt")
val vls = src.getLines.toList.groupBy(_.sorted).values
val max = vls.map(_.size).max
vls filter (_.size == max) map (_ mkString " ") mkString "\n"
```

JAVA

```
public class Person{
    private final String firstName;
    private final String lastName;

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public Person withFirstName(String firstName) {
        return new Person(firstName, lastName);
    }

    public Person withLastName(String lastName) {
        return new Person(firstName, lastName);
    }

    public String toString() {
        return "Person(" + firstName + "," + lastName + ")";
    }
}
```

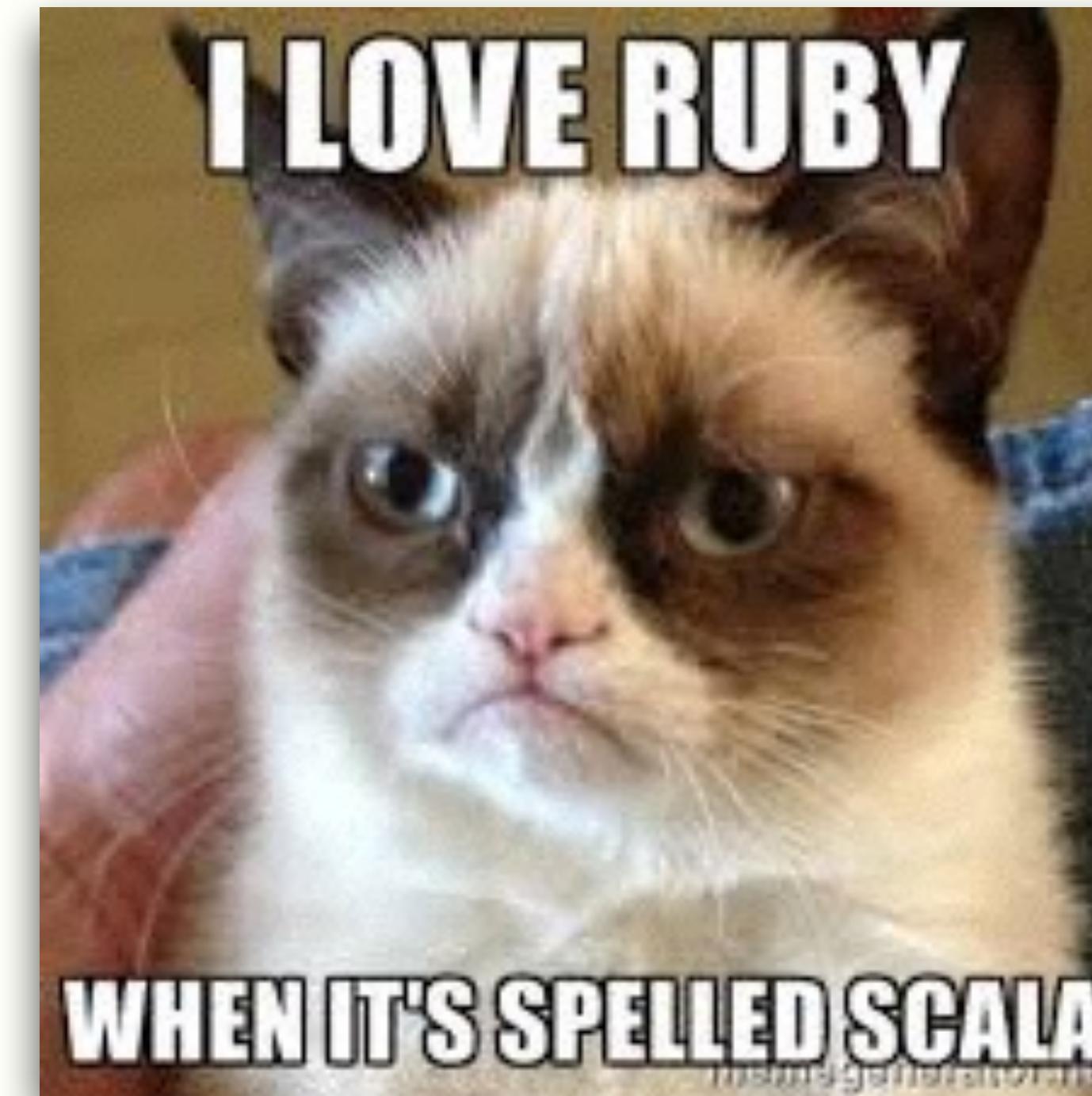
```
Person mr = new Person("Bob", "Dobbelina");
Person miss = new Person("Roberta", "MacSweeney");
Person mrs = miss.withLastName(mr.getLastName());
```

SCALA

```
case class Person(firstName: String, lastName: String)
```

```
val mr = Person("Bob", "Dobbelina")
val miss = Person("Roberta", "MacSweeney")
val mrs = miss copy (lastName = mr.lastName)
```

**DON'T TAKE MY WORD FOR IT,
HERE'S WHAT THE EXPERTS SAY ABOUT SCALA:**





Jan Schulte
@janschultecom

 Follow

After using [#scala](#) for a month touching [#java](#)
feels like repairing a Swiss watch with a drill



...

RETWEETS

8

FAVORITE

1



6:17 AM - 2 Feb 2014

I CAN HONESTLY SAY IF SOMEONE HAD SHOWN ME
THE PROGRAMMING IN **SCALA** BOOK BY MARTIN
ODERSKY, BACK IN 2003 I'D PROBABLY HAVE
NEVER CREATED GROOVY

JAMES STRACHAN

CREATOR OF THE GROOVY PROGRAMMING LANGUAGE

SCALA, IT MUST BE STATED, IS THE CURRENT HEIR APPARENT TO THE JAVA THRONE. NO OTHER LANGUAGE ON THE JVM SEEKS AS CAPABLE OF BEING A REPLACEMENT FOR JAVA.

CHARLES NUTTER

CREATOR OF THE JRUBY

WHO'S USING SCALA?



Much of **Twitter's** infrastructure is written in Scala



Many of **PayPal's** internal frameworks are Scala



Most of **LinkedIn** is written in Scala too!



The Guardian uses Scala to develop their Content API for faster application delivery



Scala Migrations is developed at **Sony Pictures Imageworks** to manage database versioning for internal applications

INTO THE WILD: LOAD TESTING WITH GATLING



Gatling
STRESS TOOL

WE'LL BE LOOKING AT EXAMPLES
OF THIS THIS LATER...



GROOVY: JAVA++

- ▶ Expressive, dynamic, object oriented language
- ▶ It **IS** Java without the ceremony, boilerplate code and, of course, semicolons
- ▶ Blocks, closures and other awesome constructs from more modern languages
- ▶ Good choice when Java verbosity hurts and dynamic typing isn't an issue



WHY GROOVY?

- **DYNAMIC**
- **SCRIPTABLE**
- **IT'S JAVA**

- Natural for Java developers; smooth transition into the language
- Both Static and Dynamic Typing
- Can be executed as a script using all your legacy Java code and your favorite libraries
- Reflection in hand
- Grails!

SOME BASICS

- ▶ Hello World!

```
println "Hello World"
```

- ▶ Iteration through an array

```
[a, b, c, d, e].each { |x| print x }
```

- ▶ Hashes

```
def animalSounds = [dog: 'woof', cat: 'meow']
animalSounds['dog'] => woof
```

- ▶ Ranges

```
for (i in 1..10) {
    println "Hello ${i}"
}
```

- ▶ Closure

```
def printSum = { a, b -> print a+b }
printSum( 5, 7 ) //prints "12"
```

- ▶ Loops

```
def x = 0
def y = 5
while ( y-- > 0 ) {
    x++
}
```

ORIGINAL EXAMPLE

```
public class HelloWorld {  
    private String name;  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String greet() {  
        return "Hello " + name;  
    }  
  
    public static void main(String[] args) {  
        HelloWorld helloWorld = new HelloWorld();  
        helloWorld.setName("Groovy");  
        System.out.println( helloWorld.greet() );  
    }  
}
```

GROOVY-FIED EXAMPLE:

```
class HelloWorld {  
    def name  
    def greet() { "Hello ${name}" }  
}  
  
println new HelloWorld(name: "Groovy").greet()
```

WHO'S USING GROOVY?

NETFLIX



Mutual of Omaha



Netflix offers various levels of filtering through scripts.
Also used Groovy and Grails to build Asgard, a cloud management platform

Mutual of Omaha (Fortune 500 insurance company in the US) uses Groovy as an embedded business language

National Cancer Institute uses it to do scientific simulations

INTO THE WILD: SCRIPTING & GRADLE



SCRIPTING IN THE JVM WITH GROOVY

Provides a way automate mundane, repeatable tasks

“Quick and Dirty”

No ceremonial setup and boilerplate code

GRADLE: BUILD AUTOMATION EVOLVED

- ▶ Automated build tool
- ▶ Combines the power and flexibility of **Ant** with the dependency management and conventions of **Maven** into a more effective way to build
- ▶ “build scripts are code” - Groovy DSL



Flexibility
Full control
Chaining of targets



Dependency management



Convention over configuration
Multimodule projects
Extensibility via plugins



Groovy DSL on top of Ant



TASK EXAMPLES

- ▶ Required Hello World Example

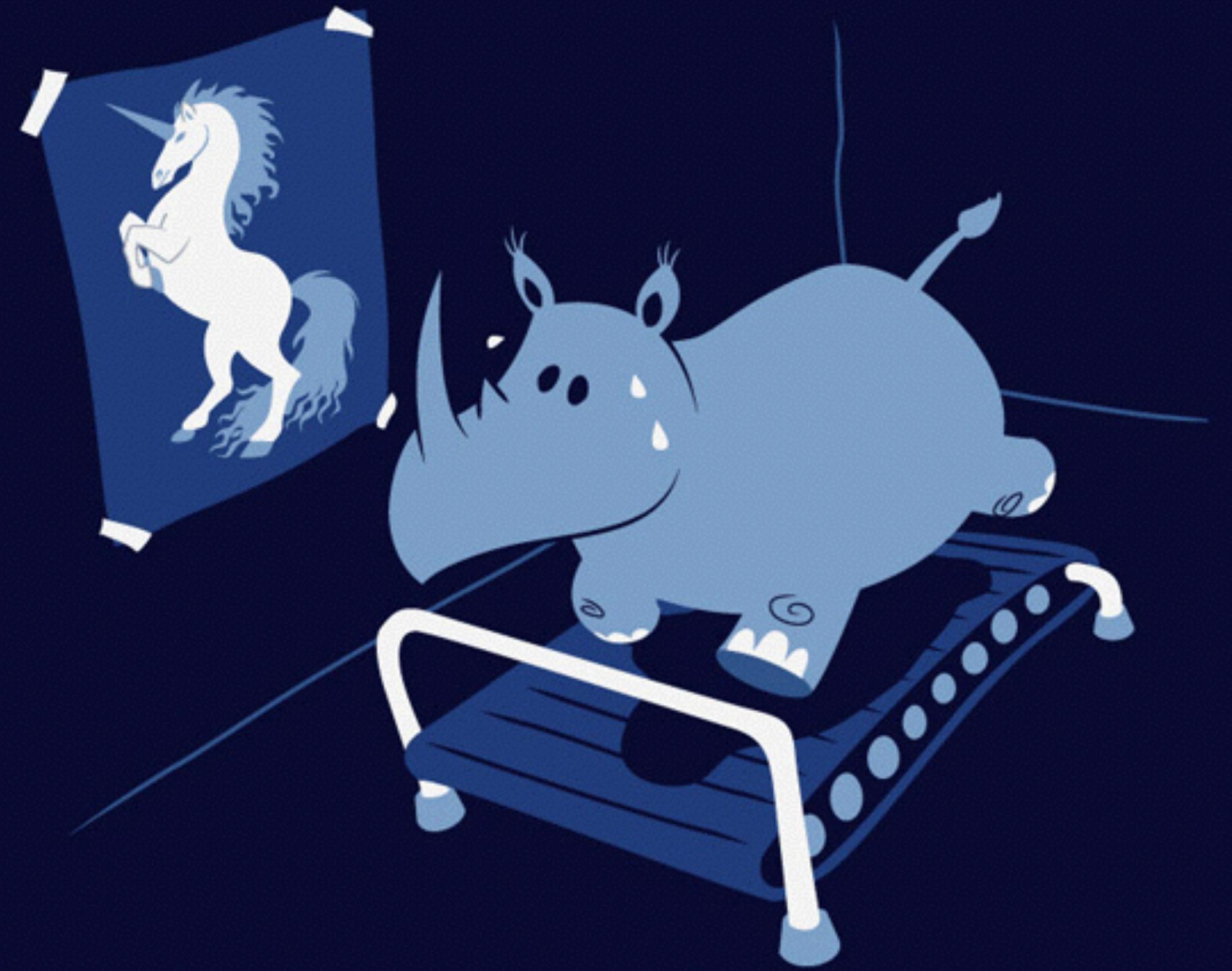
```
task hello << {
    println 'Hello world!'
}
```

- ▶ Dynamically Created Tasks

```
4.times { counter ->
    task "task$counter" << {
        println "I'm task number $counter"
    }
}
```

- ▶ Tasks Dependent on Each Other

```
task hello << {
    println 'Hello Earth'
}
hello.doFirst {
    println 'Hello Venus'
}
hello.doLast {
    println 'Hello Mars'
}
hello << {
    println 'Hello Jupiter'
}
```



JAVASCRIPT WITH NASHORN

ATWOOD'S LAW | 'AT, WOODS LÔ I

NOUN

**ANY APPLICATION THAT CAN BE
WRITTEN IN JAVASCRIPT, WILL
EVENTUALLY BE WRITTEN IN
JAVASCRIPT.**

NASHORN: THE JVM'S OWN JAVASCRIPT ENGINE

- ▶ (Almost) Everything you love about JavaScript on the JVM
- ▶ Started in July 2011/ Released March 2014 (with Java 8)
- ▶ Replaced existing Rhino Engine (Slow, slow, slooowww...)
- ▶ Compiled to bytecode (Not interpreted like Rhino)
- ▶ Uses magic of invokeDynamic to help create a truly practical JavaScript on the JVM



WHY NASHORN?

- JS MAGIC
- INVOKEDYNAMIC
- ATWOOD'S LAW

- 100% ECMAScript 5.1 compliance
- Platform independent (Unlike Google's V8)
- 2x - 10x faster than Rhino
- SMALL - only 1.5MB Jar
- End-user scripting in Java applications
 - Rule Engines
 - Reporting Engines
- Simply access Java code from JavaScript and vice-versa
- Built into Java 8 (So you probably already have it!)

SOME BASICS

▶ Hello World!

```
var hello = function() {  
    print("Hello Nashorn!");  
};  
  
hello();
```

▶ Iteration through an array

```
var myStringArray = ["Hello", "World"];  
for (var str in myStringArray) {  
    print(str);  
}
```

▶ Tertiary Fun

```
var foo = bar ? 1 : 0;
```

▶ Loops

```
for (var i=0; i<5; i++) {  
    print('try ' + i);  
}
```

▶ Browser Functions

```
alert("hello, InfoWorld");  
=> script error: ReferenceError: "alert" is not  
defined in <STDIN> at line
```

THIS DOESN'T WORK BECAUSE **ALERT()**
IS A **BROWSER/DOM FUNCTION**. D'OH!

INVOKING JS FUNCTION FROM JAVA

```
import javax.script.*;
public class EvalScript {
    public static void main(String[] args) throws Exception {
        // create a script engine manager
        ScriptEngineManager factory = new ScriptEngineManager();
        // create a JavaScript engine
        ScriptEngine engine = factory.getEngineByName("JavaScript");
        // evaluate JavaScript code from String
        engine.eval("print('Hello, World')");
    }
}
```

GET REFERENCE OF JAVA TYPE

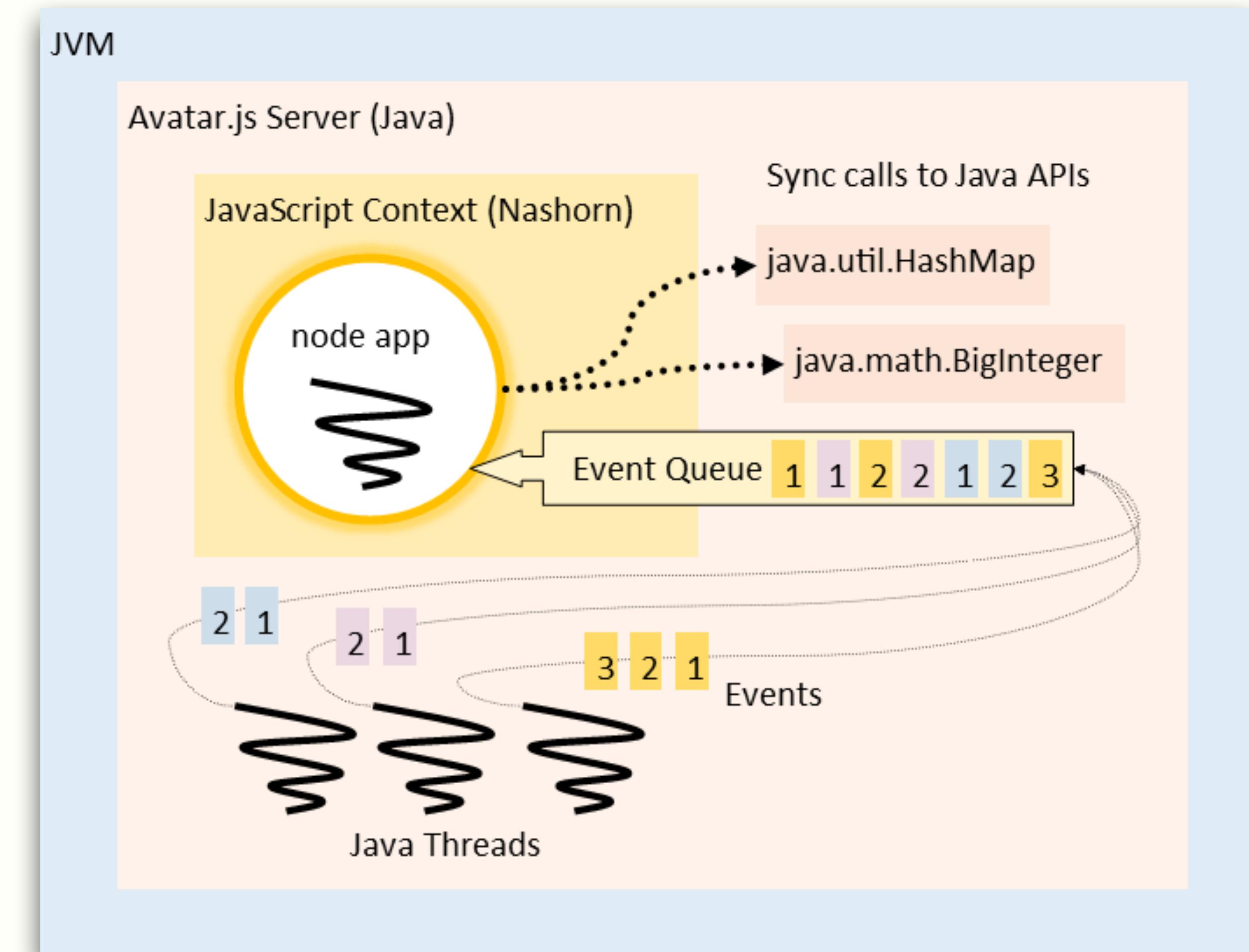
```
var HashMap = Java.type('java.util.HashMap');
var hashmap = new HashMap();
```

INTO THE WILD:

AVATAR.JS & PROJECT AVATAR

AVATAR.JS

- ▶ Open source framework for server side JavaScript applications
- ▶ focused on bringing “the node programming model, APIs and module ecosystem to the Java platform”
- ▶ 95% Node compatibility
- ▶ A mashup between:
 - ▶ Node.js
 - ▶ Nashorn
 - ▶ Java APIs

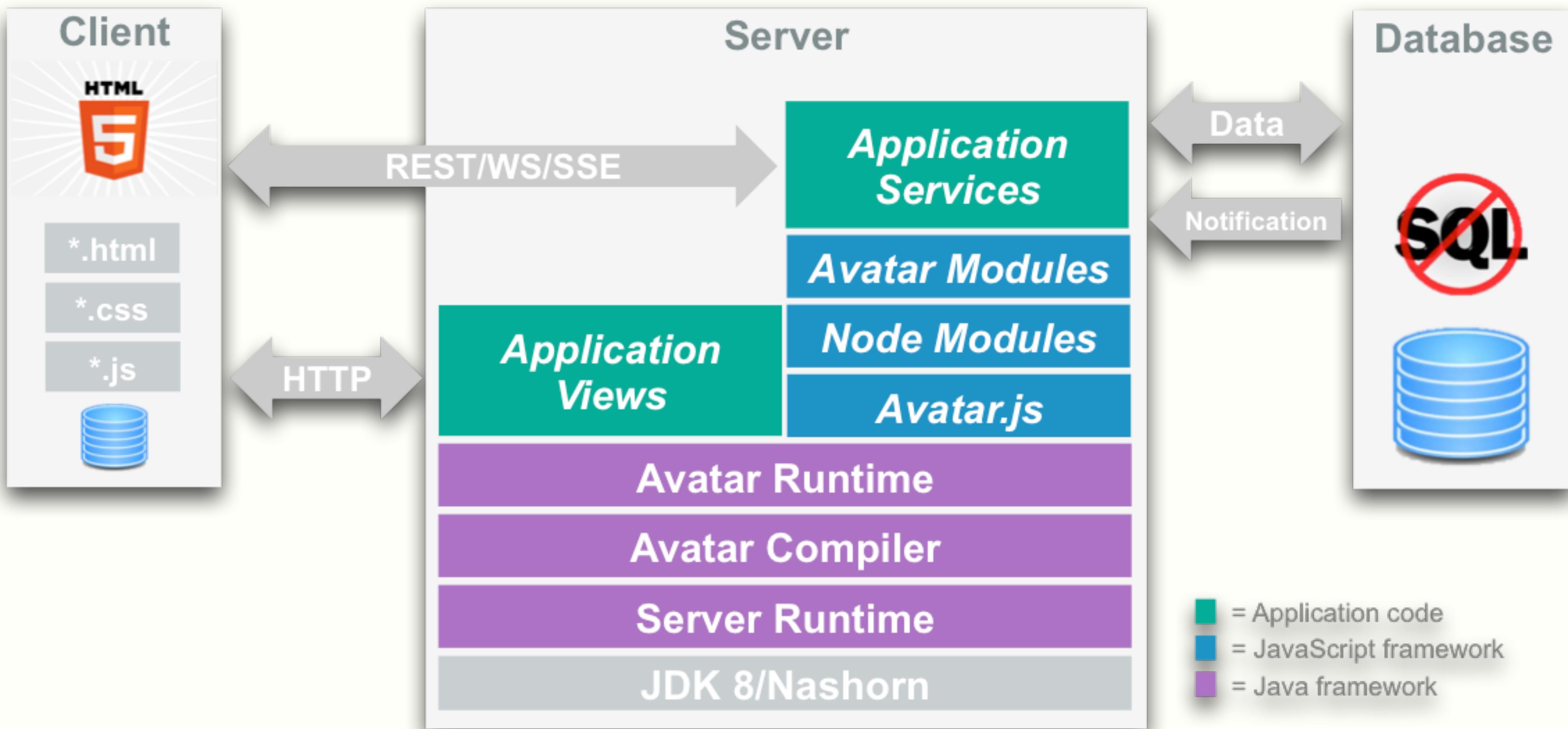


PROJECT AVATAR

- ▶ Focus on Thin Server Architecture
- ▶ Build services using JavaScript
- ▶ Focus on REST, WebSocket, Server Sent Event (SSE) endpoints
- ▶ Use familiar Node.js model and modules
- ▶ Built on Avatar.js



PROJECT AVATAR ARCHITECTURE



**JUST
ONE
MORE
THING...**

- JAVA IS STILL IMPROVING
- IT ISN'T DEAD YET

JAVA 8



TL;DR

- **INCREASED PRODUCTIVITY**
- **LESS BOILER PLATE CODE**
- **LESS CEREMONY**
- **SIMPLIFIES COMPLEXITY**
- **FLEXIBILITY**

- While Java continues to age, The JVM continues to be a fertile for newer, modern languages
- I encourage you to go out and explore the languages from this session deeper (Maybe use them for your next project!)
- These aren't the only languages on the JVM! Be adventurous, we live in a polyglot world; look for the best tool to do the job!

CHEERS!

