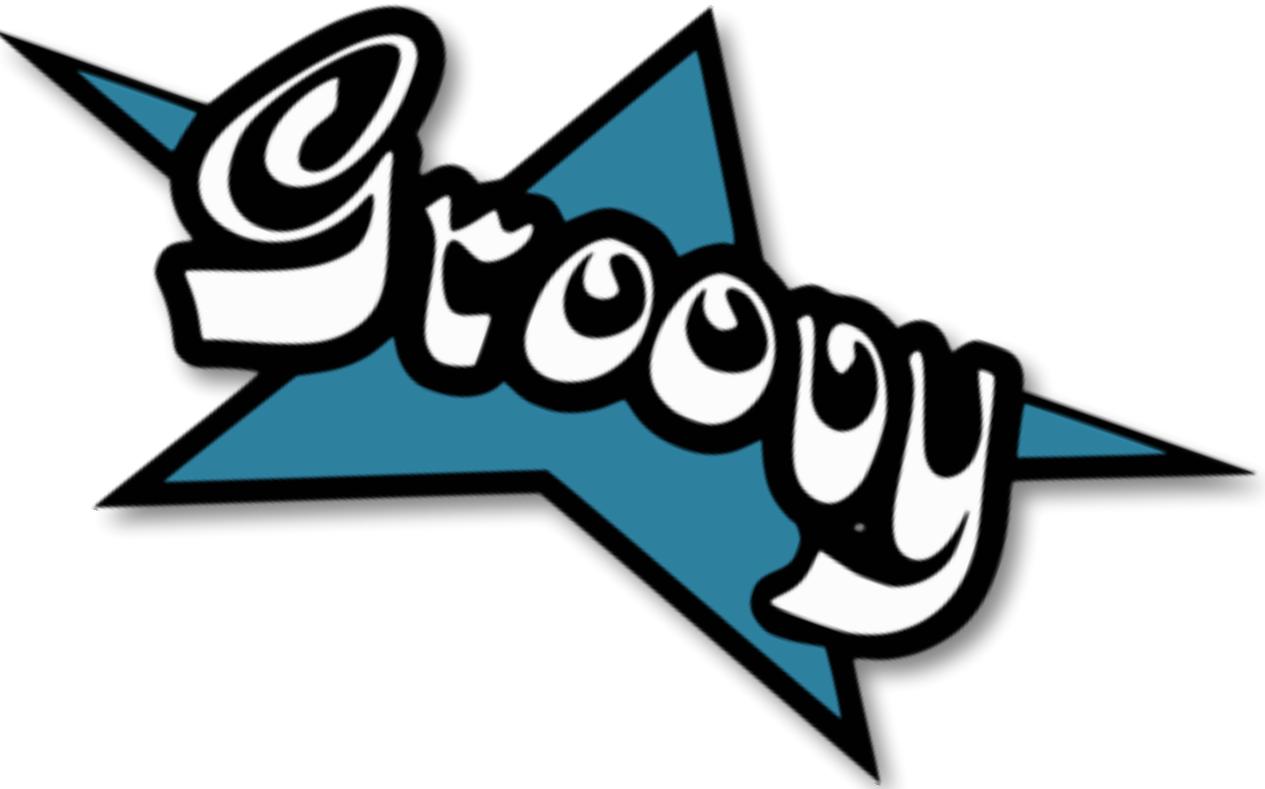




Breizh C@mp  
Mix de technologies



Guillaume Laforge  
@glaforge  
Restlet – the Web API platform  
Groovy project team

GROOVY  
son histoire,  
son oeuvre

# Restlet



We know  
about  
APIs!

RESTLET  
**studio**

RESTLET  
**framework**

RESTLET  
**APISpark**

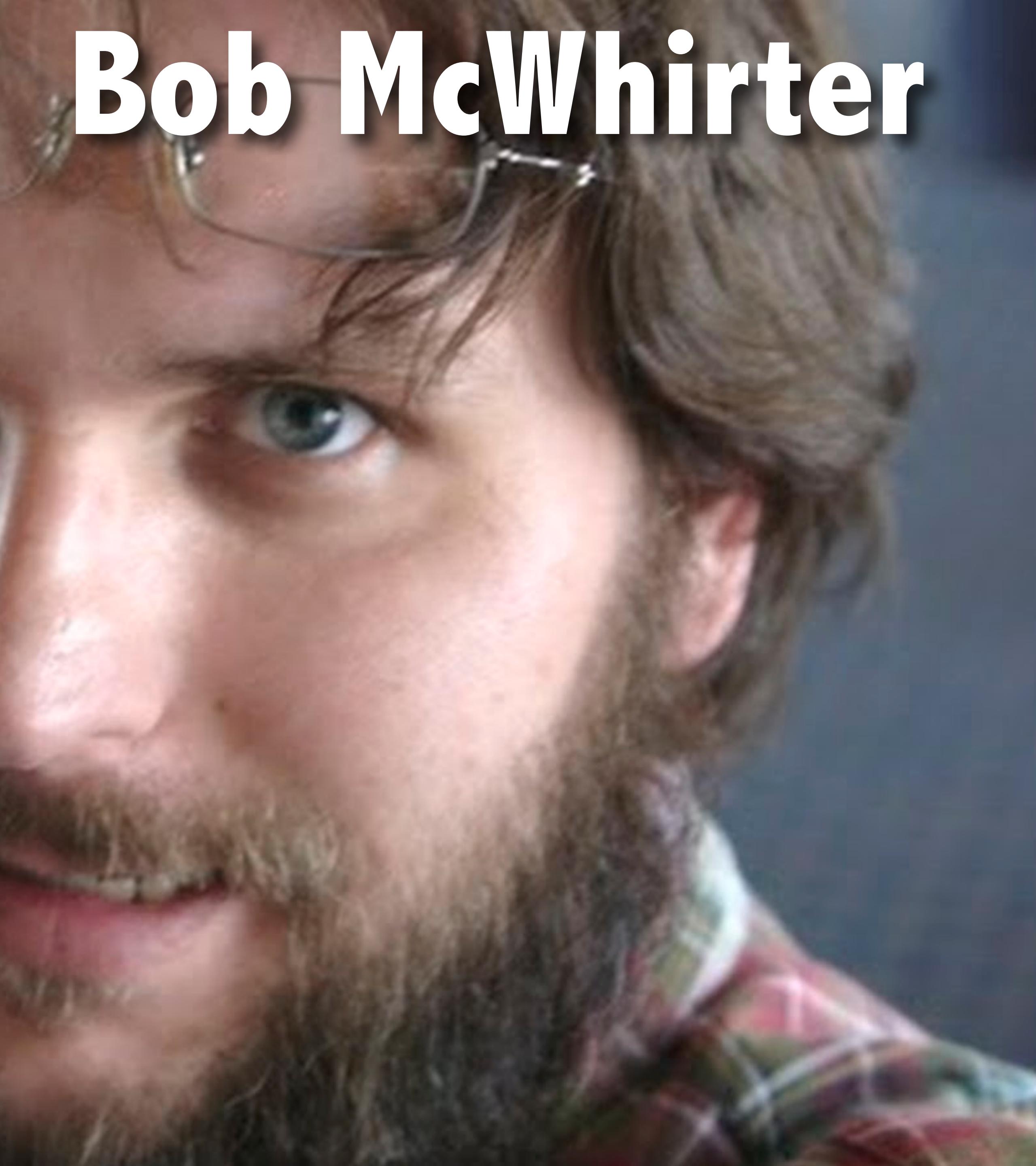


# From 2003 onward...

Restlet

# 2003



A close-up portrait of a man with light brown hair and blue eyes. He is wearing a patterned shirt with shades of red, green, and white. The lighting is dramatic, with strong highlights on his forehead and nose.

**Bob McWhirter**



**James Strachan and Bob McWhirter gave birth to Groovy: a dynamic and agile scripting language for the JVM.**

[...] James always used to say that it was all Bob's fault, but indeed, James had so many groovy ideas that it's hard to believe it's not his own fault. He kept saying: "Wouldn't it be groovy if we could do this and that...". Hence the name "Groovy". That's roughly how it all started.



**James Strachan**

# James Strachan's Weblog

Ramblings on Open Source, Java, Groovy, XML and other geeky malarkey

## James Strachan's Weblog



Friday, August 29, 2003

### [Groovy - the birth of a new dynamic language for the Java platform](#)

Dynamically typed languages like Ruby and Python are getting quite popular it seems. I'm still not convinced we should all move to dynamically typed languages any time soon - however I see no reason why we can't use both dynamically and statically typed languages and choose the best tool for the job.

I've wanted to use a cool dynamically typed scripting language specifically for the Java platform for a little while. There's plenty to choose from but none of them quite feel right - especially from the perspective of a die hard Java programmer. Python and Ruby are both pretty cool - though they are platforms in their own right. I'd rather a dynamic language that builds right on top of all the groovy Java code out there & the JVM.

Jython, beanshell and Rhino are quite close - but none of them quite feel right for my needs. (JRuby looks dead, Jython's quite behind Python though it is the closest to what I want right now). Apart from beanshell, most of them are languages designed with other use cases or criteria in mind and the bridge to the JVM was an afterthought and so don't quite sit nicely in a Java (platform & language) developers toolkit - even if that's just some minor syntax things (like `__init__` and `self` in Jython).

So I've been musing a little while if its time the Java platform had its own dynamic language designed from the ground up to work real nice with existing code; creating/extending objects normal Java can use and vice versa. Python/Jython's a pretty good base - add the nice stuff from Ruby and maybe sprinkle on some AOP features and we could have a really Groovy new language for scripting Java objects, writing test cases and who knows, even doing real development in it.

I'm finding that most of the time spent in a TDD style development model is actually coding the unit tests. There's typically lots of unit test code for little application code. Also there does seem to be some baggage when writing unit tests in java. This seems a great opportunity for using a concise & powerful dynamically typed language. So the initial idea was to make a little dynamic language which compiles directly to Java classes and provides all the nice (alleged) productivity benefits of python / ruby but allows you to reuse, extend, implement and test your existing Java code - and use that to write your unit tests. Add a little groovy maven plugin and hey presto, we've boosted the TDD development cycle and created a nice dynamically-typed alternative to Java along the way.

After some IRC chats with [bob](#) a new groovy language for the JVM has started to take shape, called Groovy. Its still very early days and much of this just exists on the [GroovyWiki](#) though bob's started work on the lexer & I've started hacking some GroovyTests (unit tests in groovy) together to test the Groovy language implementation whenever we get the compiler/runtime working.

We'd love to hear any thoughts or ideas folks have - either via [groovy-dev email list](#) or [GroovyWiki](#) or just via the comments on this blog entry. Please do add any thought you've got to the [WishList](#) page.

We're starting simple with the nice tuples, sequences, maps from python & closures from ruby and being concise & dynamically typed with a java-look-and-feel though where we end up is anyones guess right now.



# Groovy pre-1.0

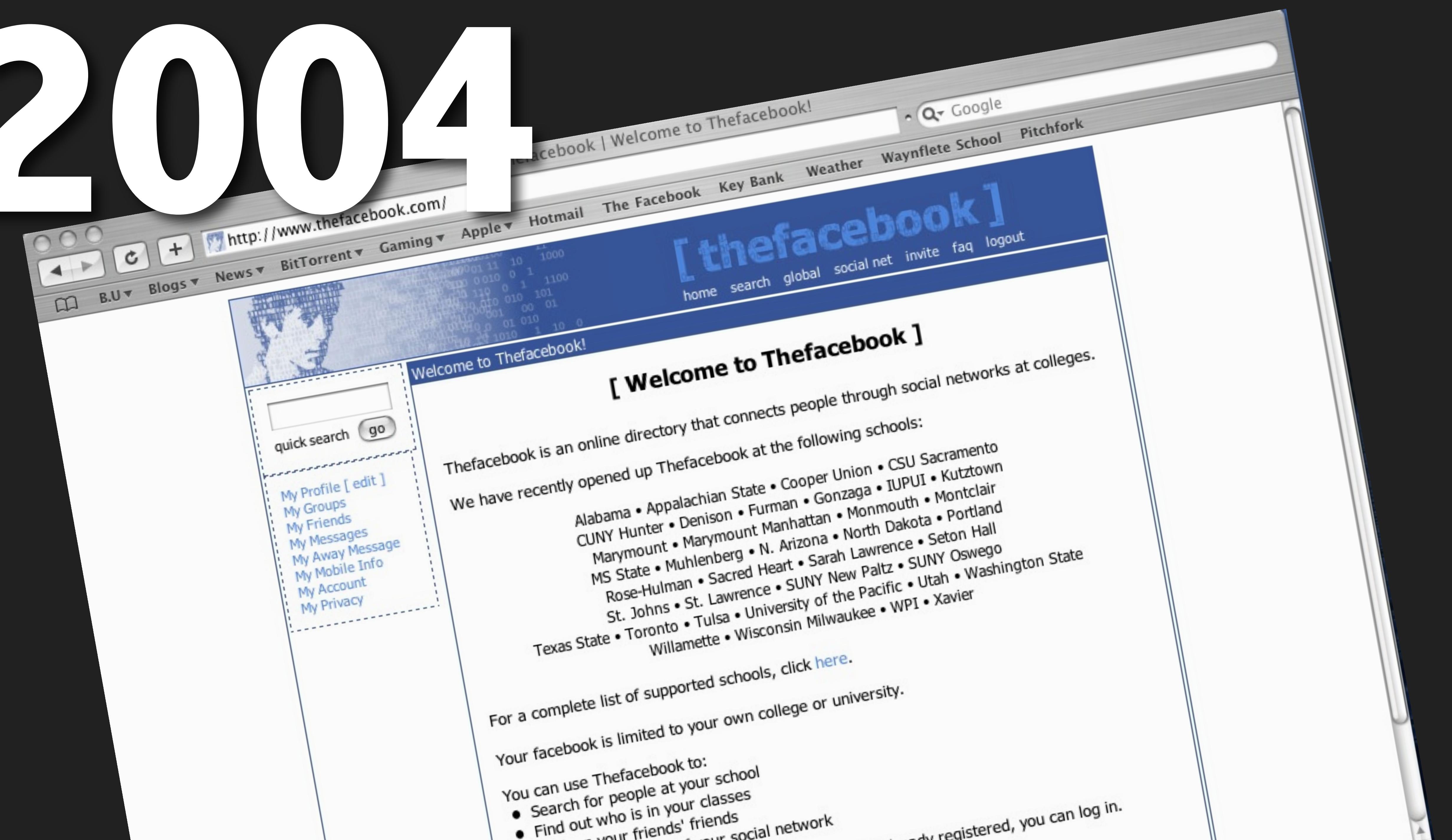
- **Closures** available since 2003 on Java 1.4+
- All the nice shortcuts already, inspired by Python & Ruby
  - native **syntax for lists, maps, etc...**
- Tons of **useful methods to simplify common Java boiler-plate code**
- Initially dynamically-typed only



A photograph of a man with short brown hair and glasses, wearing a dark t-shirt with a white graphic, speaking into a black and yellow microphone. He is standing in front of a blue curtain. A green speech bubble points from the text to him.

**I join the project in  
2003, patches after  
patches, become  
committer, then lead  
when founders left**

# 2004







# Groovy in 2004

- A brand new parser based on Antlr
  - after two hand-rolled versions
  - by John Rose and Jeremy Rainer
- Jochen « blackdrag » Theodorou joins the project

[All](#) | [General](#) | [Java](#)[« RIP JBoss Mail Servi...](#) | [Main](#) | [Kill build.property...](#) [»](#)**FRIDAY DECEMBER 10, 2004**

**The groovy sinking ship** Anyone subscribed to the groovy mailing list must either feel exceptionally gleeful, or horribly depressing. I won't rub too much salt in the wounds of the latter group by cackling about being in the former group. Oh wait, I will, because they're a bunch of spastic hobbyist fuckwits would should all be fired.

It is with particular glee that I note that no serious work is now being done on groovy. Even the morons who decided to commit to it are starting to complain. Bugs are piling up, the developers have mostly moved onto the next shiny bauble, and there are calls for fresh blood and a 'new' set of developers to get things going; a fairly compelling symptom of a project in its death throes.

Particularly hilarious is how all the turdburglars involved feel that getting together for some drinks constitutes a move forward. It's really quite amazing that any of these people have actual jobs. Is that how you do things at work? Go out for a drink and your projects just magically finish themselves? Perhaps you blog about it a bit and tell everyone how great it is, and then documents magically write themselves? Or perhaps you're one of these new breed of fuckstains, who think that the code is the documentation and brag of that, instead of the traditional shame usually associated with making such a comment. On the other hand, groovy could be breaking (more) new ground by being the first JSR where the final spec format is...mp3 (for those of you not reading the lists for their comical value, the 'outcome' of the groovy JSR gettogether is nothing but a bunch of mp3's).

Still, at least it's good to see that after all these months, the end of line semicolon saga is still dragging on, as are all of the syntactic sugar issues that people raised all those months ago. Democracy at its finest.

It almost seems spiteful in fact, how the groovy fearless leader seems to go out of his way to never post anything to the mailing lists.

It's also delightfully satisfying to see that the early adopters are being punished, as they should be, by guarantees that the syntax will change in non-backward compatible ways. You fuckers need to be clubbed on the head like that so one day, one day, you might actually learn.

As if that wasn't enough, there's also a regular posting of the build breaking or someone unable to get it going, thanks to the wonderful magic of maven. It's truly astounding what sort of shit people will put up with in order to uphold their religious convictions.

The one person who has to be pitied though is poor Guillaume Laforge, who is the only remaining groovy developer. He tirelessly responds to emails on the mailing lists even though he probably knows deep down inside that it's a lost cause. It's sad and depressing, but I suppose some people just don't know when to give up. If I were his employer though I suspect I'd gradually start getting more and more annoyed at my employee pouring so much time and effort into such a dirty toilet.

[All](#) | [General](#) | [Java](#)[« RIP JBoss Mail Servi...](#) | [Main](#) | [Kill build.property...](#) [»](#)**FRIDAY DECEMBER 10, 2004**

**The groovy sinking ship** Anyone subscribed to the groovy mailing list must either feel exceptionally gleeful, or horribly depressing. I won't rub too much salt in the wounds of the latter group by cackling about being in the former group. Oh wait, I will, because they're a bunch of spastic hobbyist fuckwits would should all be fired.

It is with particular glee that I note that no serious work is now being done on groovy. Even the morons who decided to commit to it are starting to complain. Bugs are piling up, the developers have mostly moved onto the next shiny bauble, and there are calls for fresh blood and a 'new' set of developers to get things going; a fairly compelling symptom of a project in its death throes.

Particularly hilarious is how all the turdburglars involved feel that getting together for some drinks constitutes a move forward. It's really quite amazing that any of these people have actual jobs. Is that how you do things at work? Go out for a drink and your projects just magically finish themselves? Perhaps you blog about it a bit and tell everyone how great it is, and then documents magically write themselves? Or perhaps you're one of these new breed of fuckstains, who think that the code is the documentation and brag of that, instead of the traditional shame usually associated with making such a comment. On the other hand, groovy could be breaking (more) new ground by being the first JSR where the final spec format is...mp3 (for those of you not reading the lists for their comical value, the 'outcome' of the groovy JSR gettogether is nothing but a bunch of mp3's).

Still, at least it's good to see that after all these months, the end of line semicolon saga is still dragging on, as are all of the syntactic sugar issues that people raised all those months ago. Democracy at its finest.

It almost seems spiteful in fact, how the groovy fearless leader seems to go out of his way to never post anything to the mailing lists.

It's also delightfully satisfying to see that the early adopters are being punished, as they should be, by guarantees that the syntax will change in non-backward compatible ways. You fuckers need to be clubbed on the head like that so one day, one day, you might actually learn.

As if that wasn't enough, there's also a regular posting of the build breaking or someone unable to get it going, thanks to the wonderful magic of maven. It's truly astounding what sort of shit people will put up with in order to uphold their religious convictions.

The one person who has to be pitied though is poor Guillaume Laforge, who is the only remaining groovy developer. He tirelessly responds to emails on the mailing lists even though he probably knows deep down inside that it's a lost cause. It's sad and depressing, but I suppose some people just don't know when to give up. If I were his employer though I suspect I'd gradually start getting more and more annoyed at my employee pouring so much time and effort into such a dirty toilet.



2005



# GRAILS

# 2006







# First paid committer

- Jochen becomes the **first full-time paid committer**
  - sponsored by BigSky Technologies / No Fluff Just Stuff



# 2007











**G2**  
**one**



**Skills Matter**

Search

3 DAY CONFERENCE

# Grails eXchange 2007

Topics covered:

groovy grails

Wednesday, 17th - Friday, 19th October at [Sekforde Street, London](#)

18 experts. will be speaking. Starts at 10:00 AM.

Overview    **Programme**    Venue    Other Years    Skillscasts    Sponsors

Skills Matter are pleased to have organised the First Grails eXchange 2007! This 3-day event held in Oct 2007 was a great opportunity to get a heads-up on the latest developments in the world of Groovy & Grails and to meet interesting speakers and fellow Groovy and Grails developers.





# Spock

the Enterprise  
testing framework





# Groovy in 2007 — Groovy 1.5

- The long awaited Java 5 update
  - **annotations, enums, generics...**
- **Joint compiler!**
- Varargs support
- Static imports (including import aliasing)
- Named parameters without parentheses



**Paul King, #1 committer  
from 2007 to 2010, and  
overall #1 committer too!**



# 2008



# GR Conf

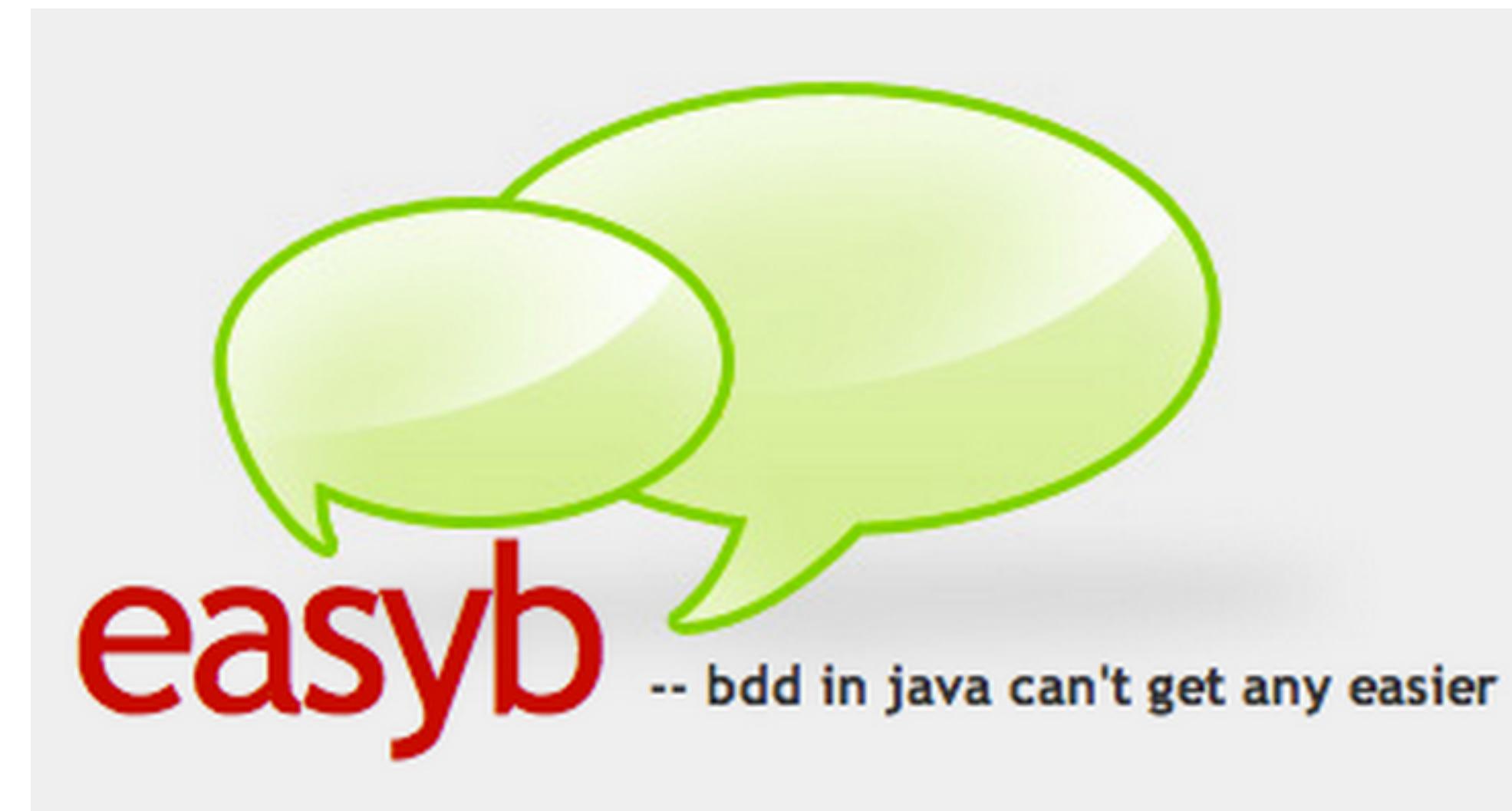




**spring**  
s o u r c e



# Griffon and Easyb





# Groovy++ fork!

- Not the first fork, but 3rd
  - by Alex Tkachman (former G2One co-founder)
- Big focus on static compilation
  - interesting playground for our forthcoming static type checking and static compilation support

2009



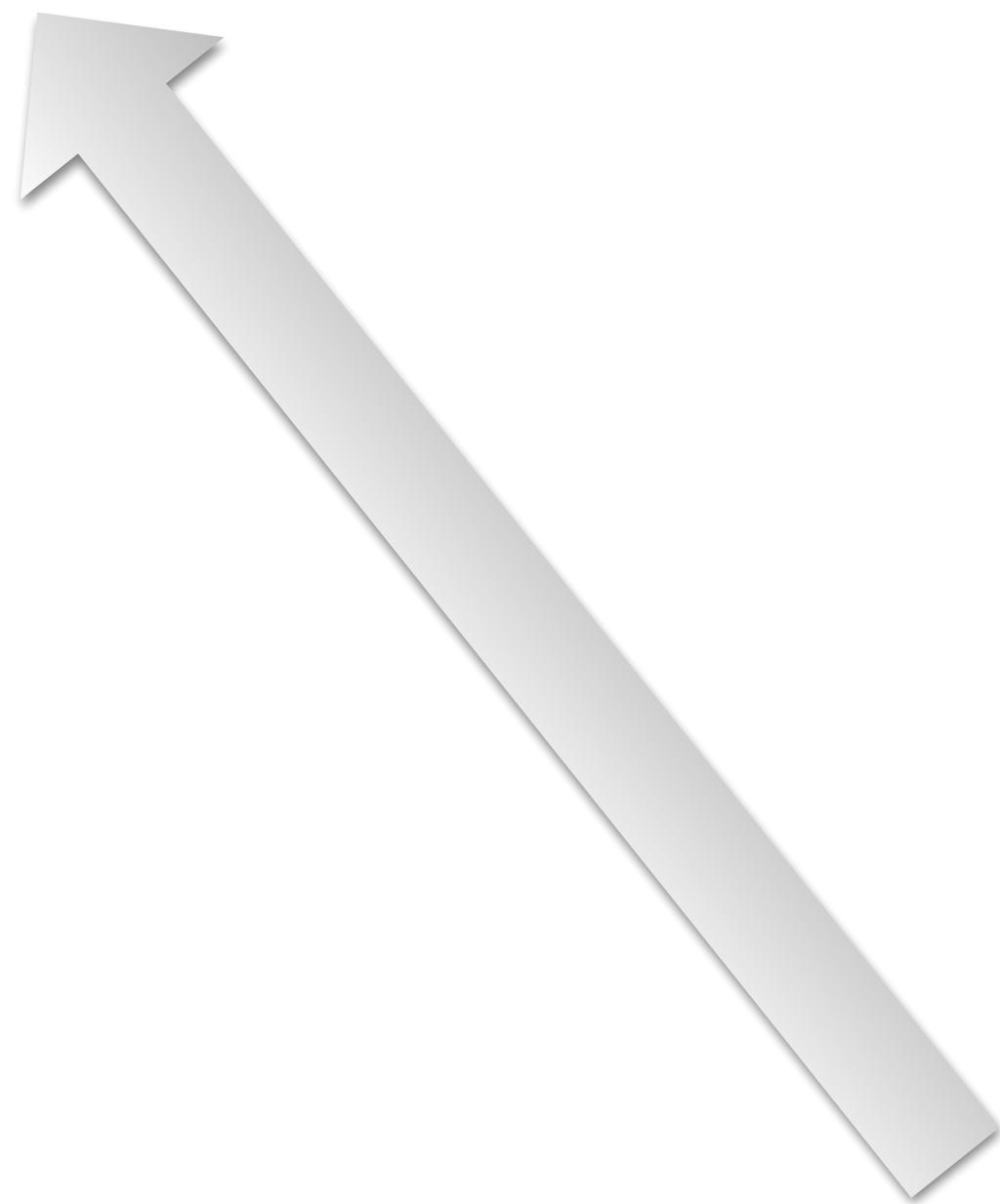
I can honestly say if someone had shown me the Programming Scala book back in 2003, I'd probably have never created Groovy

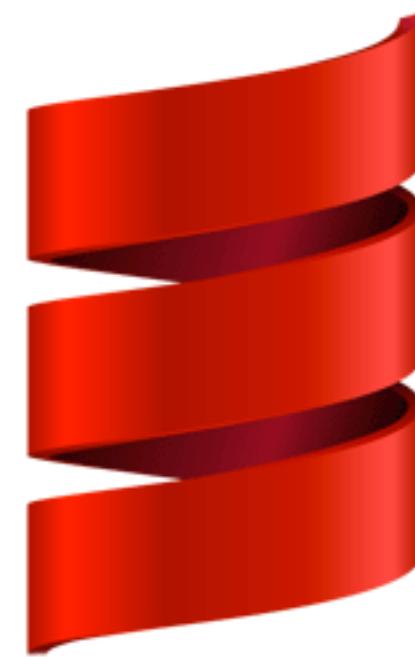




Have I had JPA  
back then,  
I would  
certainly not  
have created  
**Hibernate!**



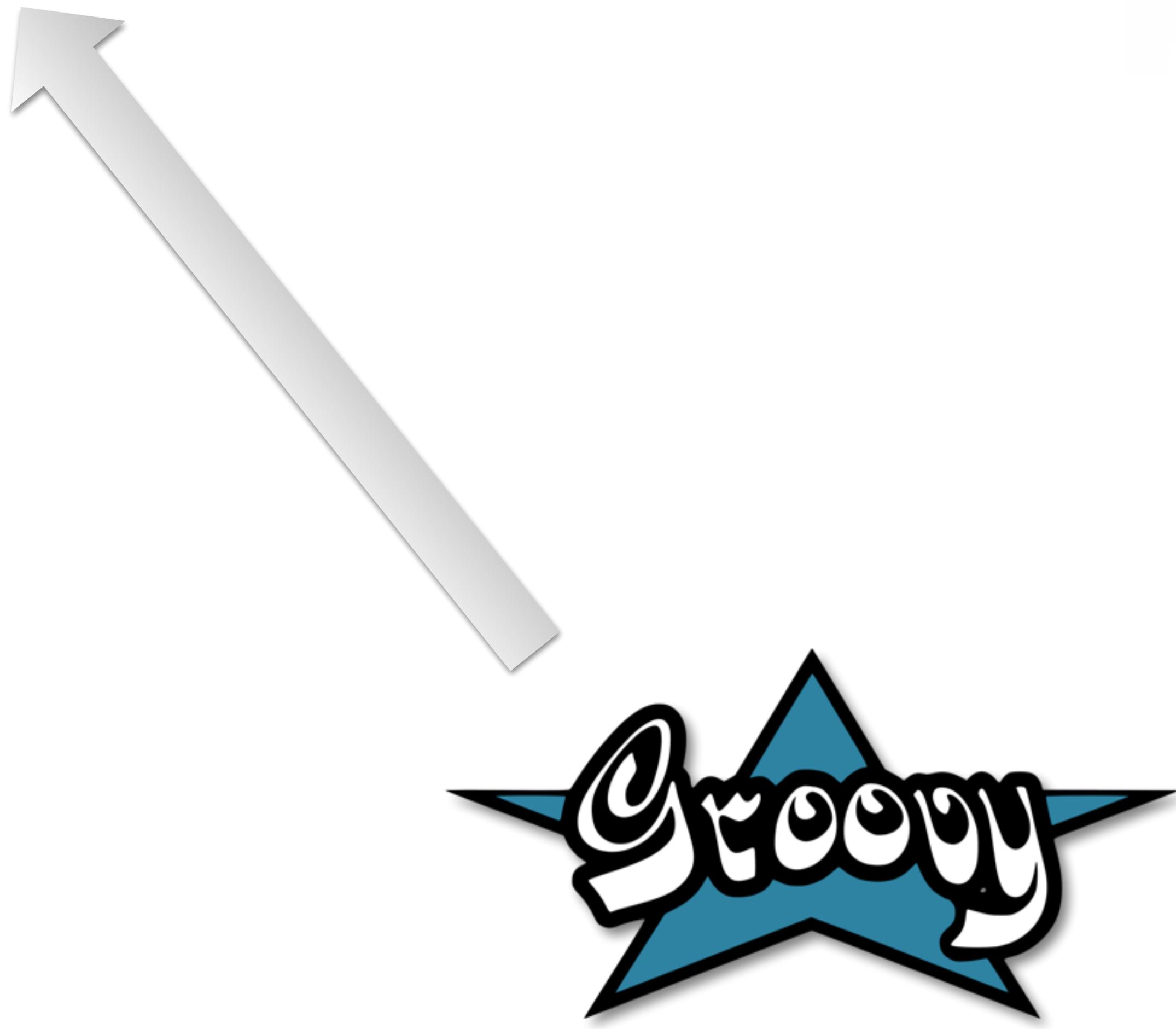


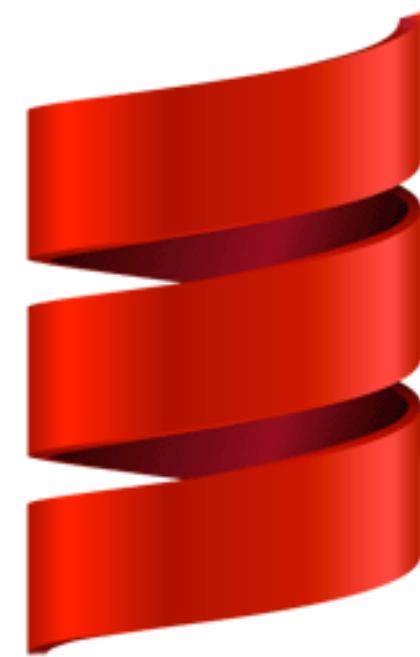


Scala



PROJECT  
**Kotlin**  
by *jet*BRAINS

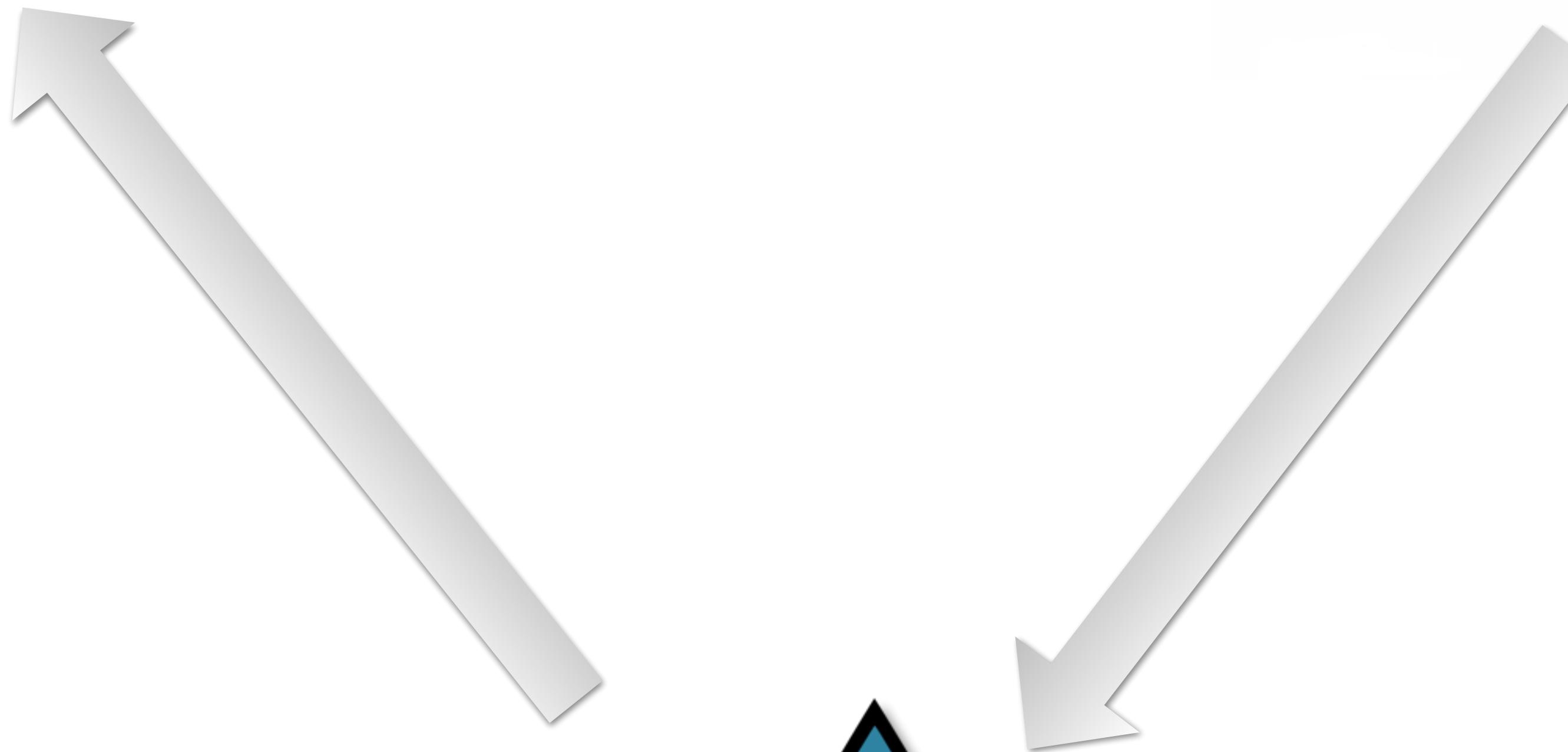




Scala



PROJECT  
**Kotlin**  
by *jet*BRAINS







**James Strachan**  
@jstrachan



Following

hacking groovy in IDEA/REPL/Jenkins via the  
Job DSL for CI/CD

[fabric8.io/guide/cdeliver...](http://fabric8.io/guide/cdeliver...) forgot how much  
fun it is! Its so groovy! #groovyFTW



...

RETWEETS

28

FAVORITES

26



4:34 PM - 5 Jun 2015





**James Strachan**  
@jstrachan



Following

if you use jenkins then have a look at the  
Jenkins job DSL plugin

[github.com/jenkinsci/job-dsl-plugin](https://github.com/jenkinsci/job-dsl-plugin) groovy, jenkins &  
git are a great combination!



**jenkinsci/job-dsl-plugin**

A Groovy DSL for Jenkins Jobs - Sweeeeet! Contribute to job-dsl-plugin development by creating an account on GitHub.







**James Strachan**  
@jstrachan



Following

it puts a smile on your face seeing elegant code in the language you created on slides in interesting presentations ;) #mucon  
#groovyFTW



...

RETWEETS

8

FAVORITES

13



12:07 PM - 28 Nov 2014

vmware®



# The Groovy Ecosystem grows...





# Groovy 1.6

- Performance improvements with call-site caching
- Multiple assignments
- Optional return in if/else, try/catch, switch/case
- More AST transformations
  - `@Singleton`, `@Lazy`, `@Immutable`, `@Delegate`
- The Grape module system with `@Grab`
- OSGi support



# Groovy 1.7

- Nested and inner classes, and nested static classes
- Missing annotation definition support
- **Power Asserts** from Spock
- Customize the Groovy Truth!
- AST viewer and AST builder
- SQL support improvements

An aerial photograph of a volcano in eruption. The volcano's slopes are dark and covered in ash. A massive, billowing plume of dark grey smoke and ash rises from the crater, extending across the frame. The background shows a bright sky and some distant land or clouds.

2010



# Groovy ecosystem marching on...



# GContracts



2011





# More Groovy ecosystem...

# VERT.X



# GroovyFX

*The easiest way to write JavaFX 2.0 applications in Groovy.*



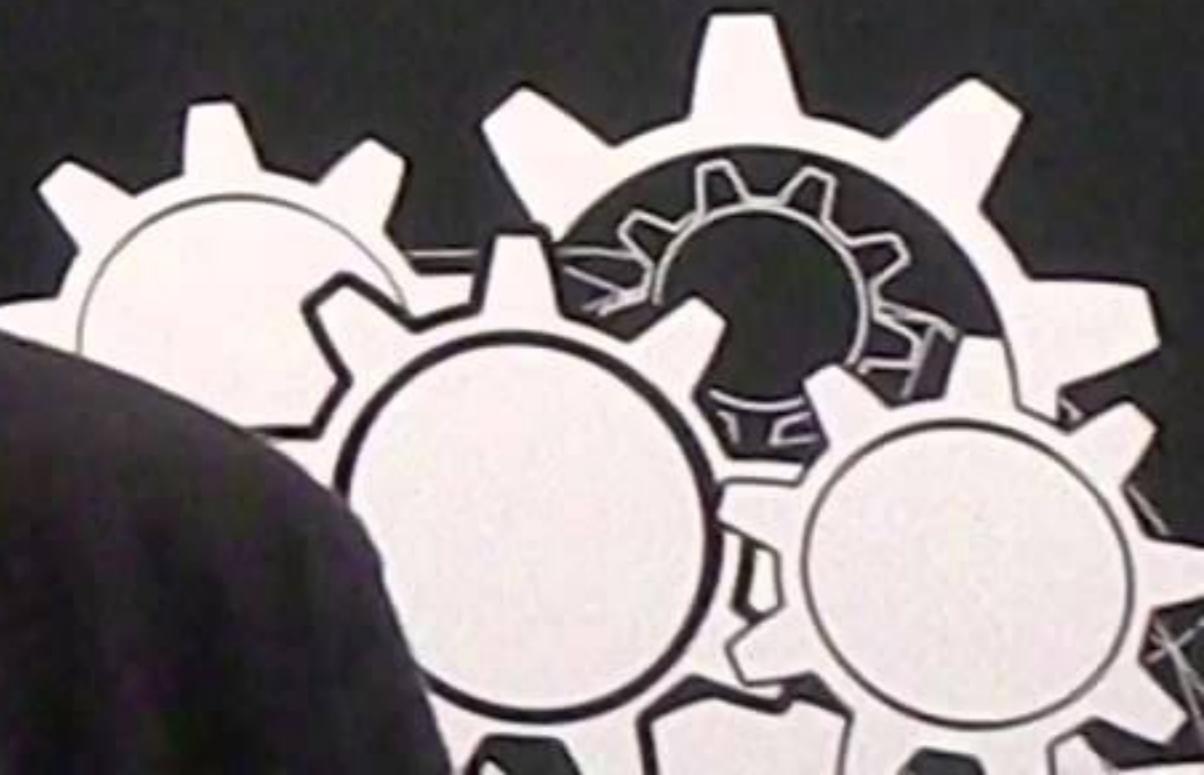
# Groovy 1.8

- **Command chain expressions**
- GPars bundled with the Groovy distribution
- Primitive calculation optimizations
- **Closures**
  - as annotation parameters
  - currying, memoization and tail recursion
- Built-in JSON support



**VOX**  
SHARE THE K...

I'm joining the  
Groovy team!





**github**  
SOCIAL CODING



2012



# Groovy ecosystem continued...

- Netflix
  - Asgard,
  - Zuul
- LinkedIn Glu
- GroovyStream





# Groovy 2.0

- **Static type checking and static compilation**
  - type inference, flow typing...
- « Project Coin » enhancements
  - binary literals, underscore in numbers, multicatch block
- Initial « invoke dynamic » support
- **Modularity of the Groovy code base with modules**

2013



Pivotal

TM



# Richer Groovy ecosystem...



**Gretty**

Gradle plugin for running web-apps on servlet containers.

 **Gaiden**

**nextflow**

**gIPSY**



# Groovy 2.1

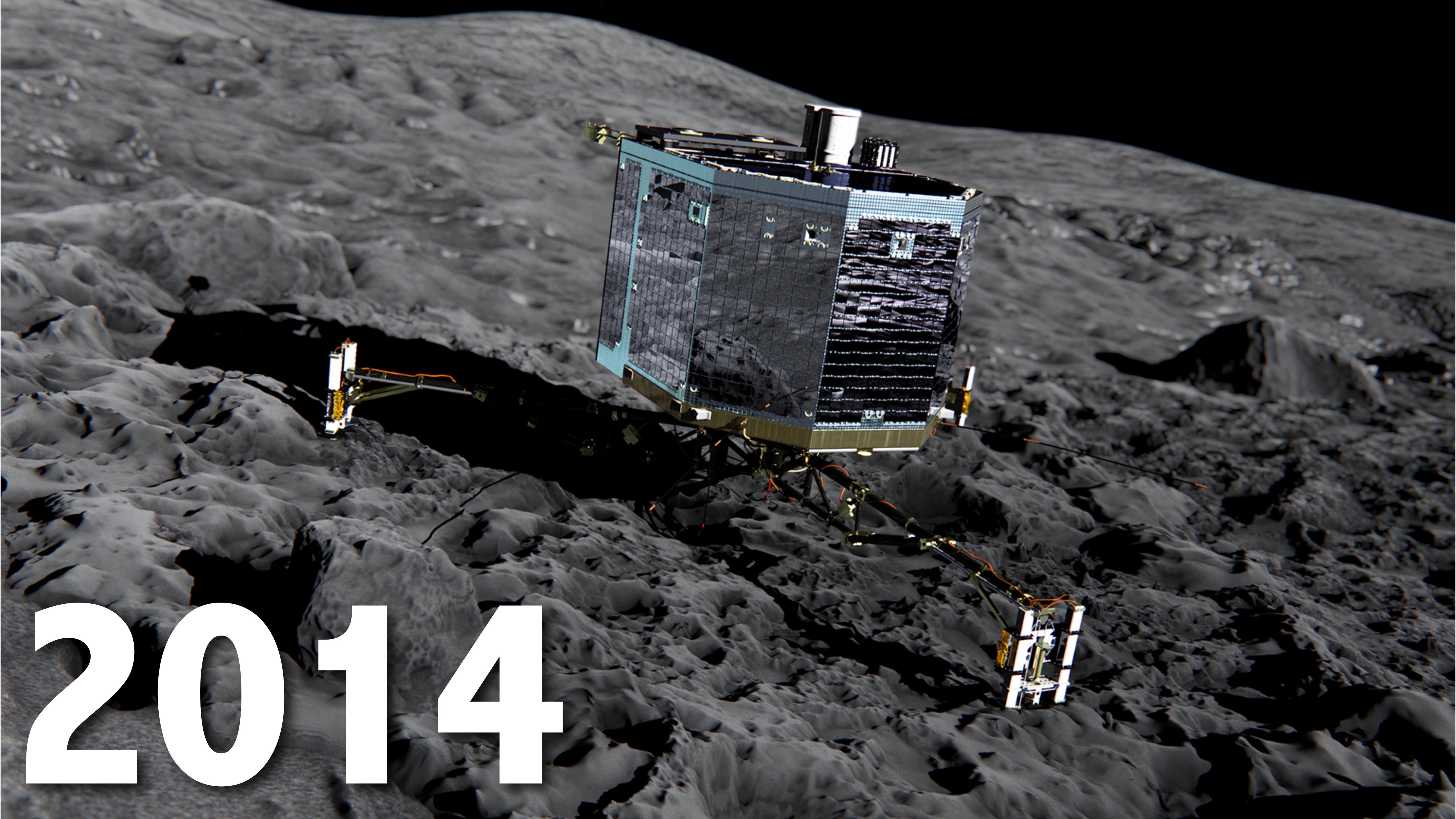
- Full « invoke dynamic » support
- **Type-checking extensions**
- **Meta-annotations**
- **@DelegatesTo annotation**
- Compiler customization scripts
- Distribution bundling GPars 1.0



# Groovy 2.2

- Implicit **closure coercion to SAM types**
  - Java lambda friendly!
- Pre-compiled type checking extensions
- **@Memoized** transformations for methods

# 2014







# Groovy ecosystem

{ elasticsearch.





# Groovy 2.3

- Traits!
- Improved JDK 8 support
- Drastic performance improvement for JSON
- New AST transformations & improvements
  - @Builder, @Sortable, @SourceURI, @BaseScript
- New markup template engine
- More NIO support

# A look back at the past 6 months



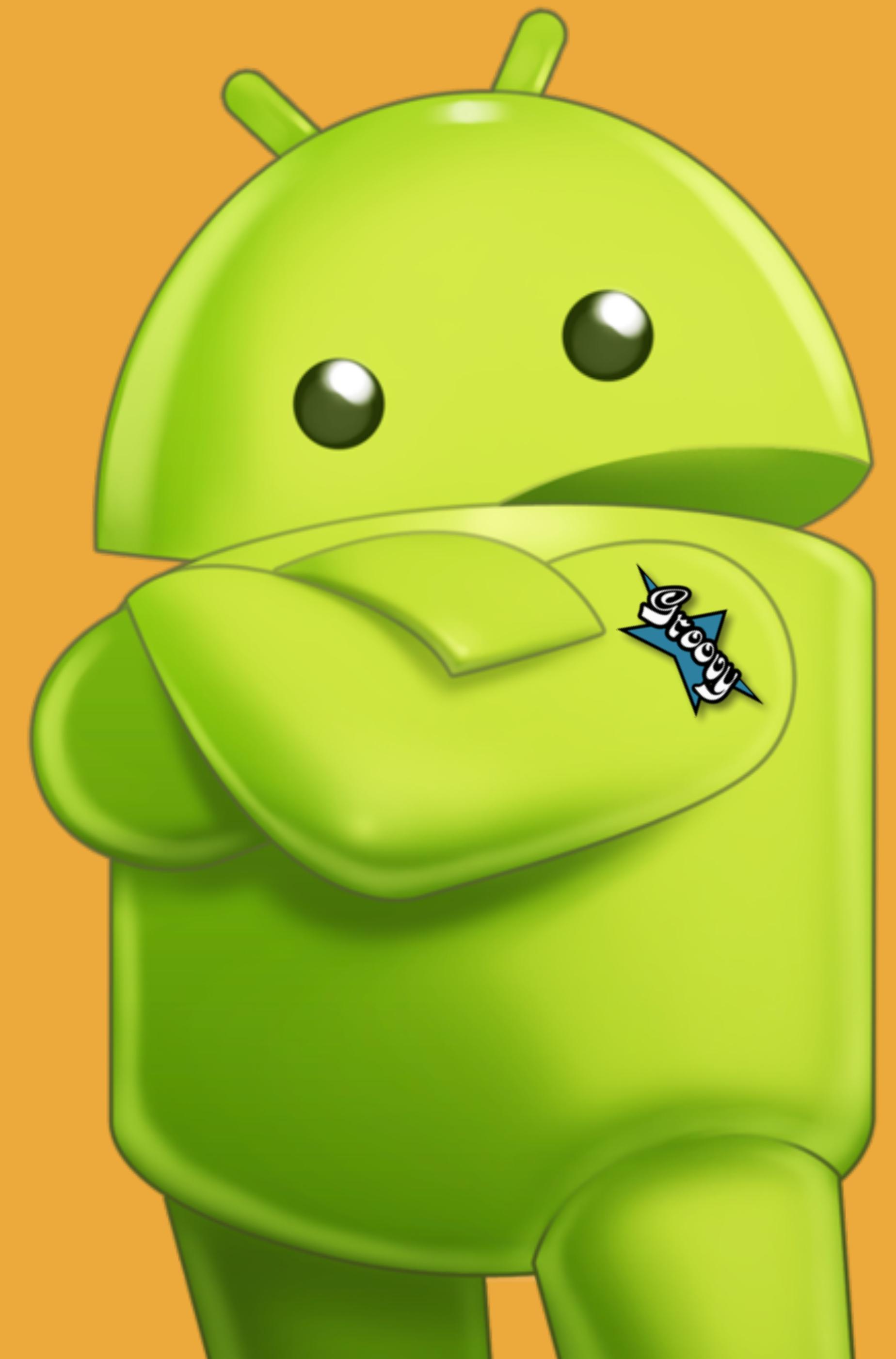
# FIRE!



MARCH

31st

MARCH 31ST 2013  
WAAZ GROOVY GRAINS



The New York Times

MARY E. WINSTEAD

Nude Photos of Jennifer Lawrence Are Latest Front in Online Privacy Debate

With Note 4 Release, Samsung Still Betting on Big Phones

BITS BLOG

Making the Leap Web to Mobile A

**OPEN**

All the Code That's Fit to `printf()`

## Getting Groovy With Reactive Android

By MOHIT PANDEY AUGUST 18, 2014 2:16 PM 2 Comments

E-MAIL

FACEBOOK

TWITTER

SAVE

MORE

“Slow,” a word no one ever wants to see associated with their app. Delay and lag of as little as 100–200 milliseconds will be perceived by users as poor, slow performance. Keeping long running tasks on the main thread will result in that perception, the dreaded “Application Not Responding” (ANR) and a bad user experience. This leaves us developers one choice: concurrent execution of code. The recent growth in popularity of functional programming is well

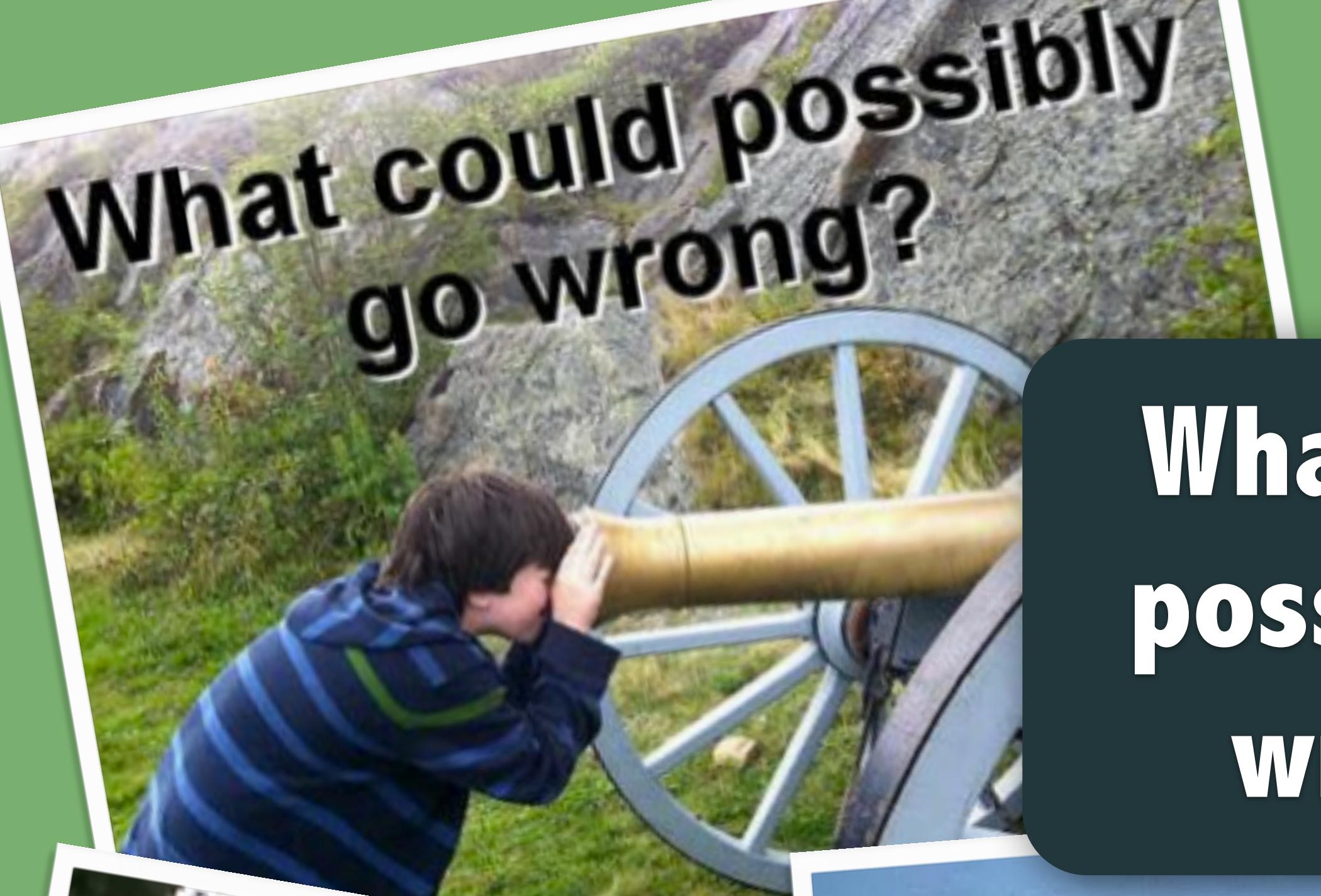




# In Groovy 2.4 also...

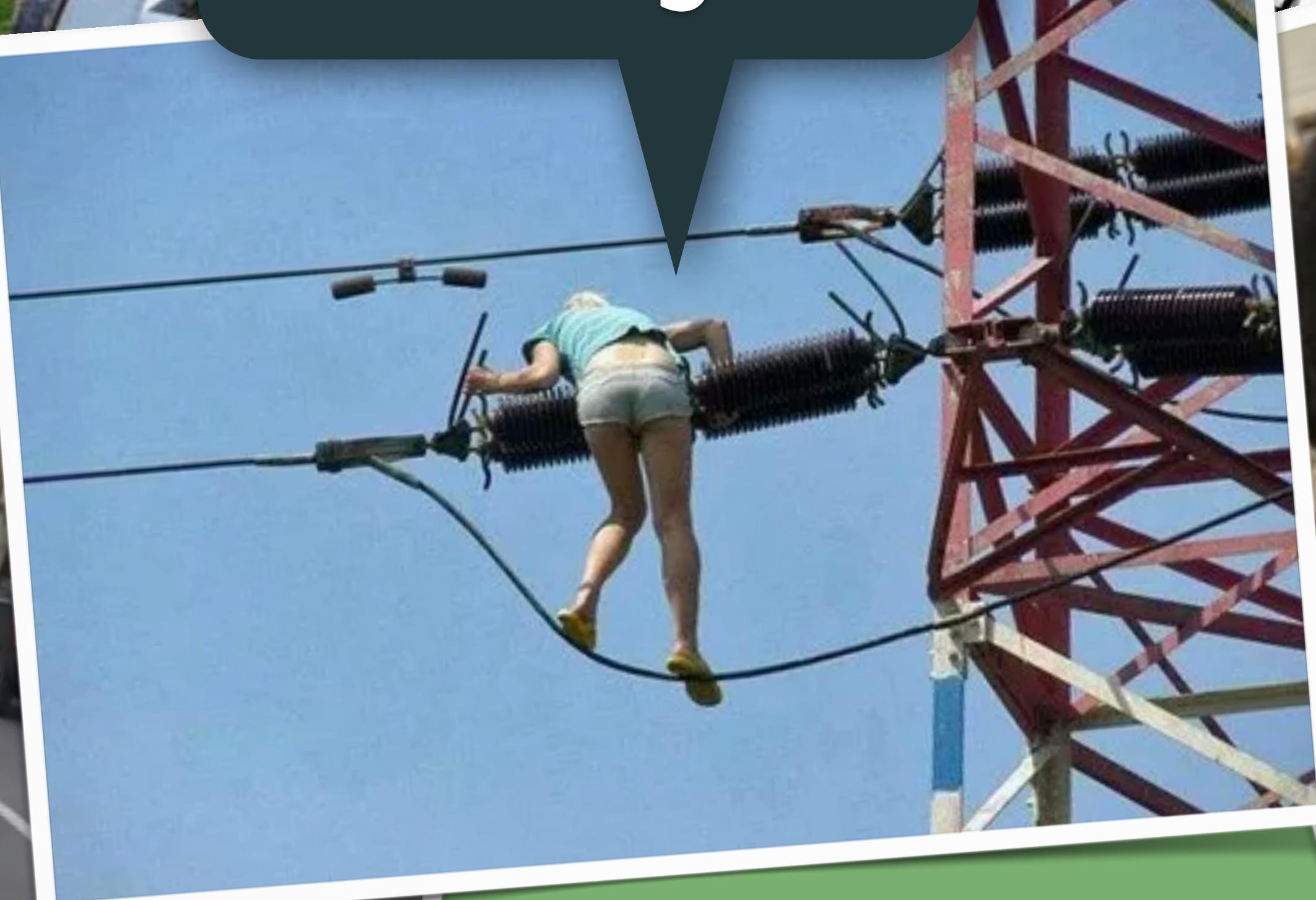
- Optimized bytecode
- Bytecode size reduction
- Reduced memory consumption
- Traits `@SelfType` annotation
- More handy GDK methods
- More AST transformation refinements
- Groovysh improvements





What could possibly  
go wrong?

What could  
possibly go  
wrong?





THE CODEHAUS  
DEMISE



# IMPACT?

# Downloads

2012

2013

2014

2015

# Downloads

1.7M

2012

3M

2013

4.5M

2014

2015

?

**2.1 MILLIONS**

**FOR FIRST 4 MONTHS**

**OF 2015**

**MAVEN  
CENTRAL**

**ONLY**

**CODEHAUS**  
+  
**COGAGA**

**MAVEN**  
**CENTRAL**  
**ONLY**

+CODEHAUS  
+  
CODEGAU  
+  
CODEGAU  
+  
CODEGAU

MAVEN  
CENTRAL  
ONLY

+  
CENTRAL  
ONLY



# And now, what's next?

Restlet

# MUSICAL CHAIRS



# Restlet



**GROOVY**

**CEDRIC  
GUILLAUME  
JOCHEN**

**GRAILS**

**GRAEME  
JEFF  
LARI**



**?**

# Restlet

**GUILLAUME**

**CEDRIC**

**LARI**



**GROOVY**

**GRAILS**

**JOCHEN** ?



**GRAEME**  
**JEFF**



# The Apache Software Foundation



# MESSAGE TO THE COMMUNITY



HERE TO  
STAY

HERE TO  
STAY

INDEPENDENCE

HERE TO  
STAY  
INDEPENDENCE

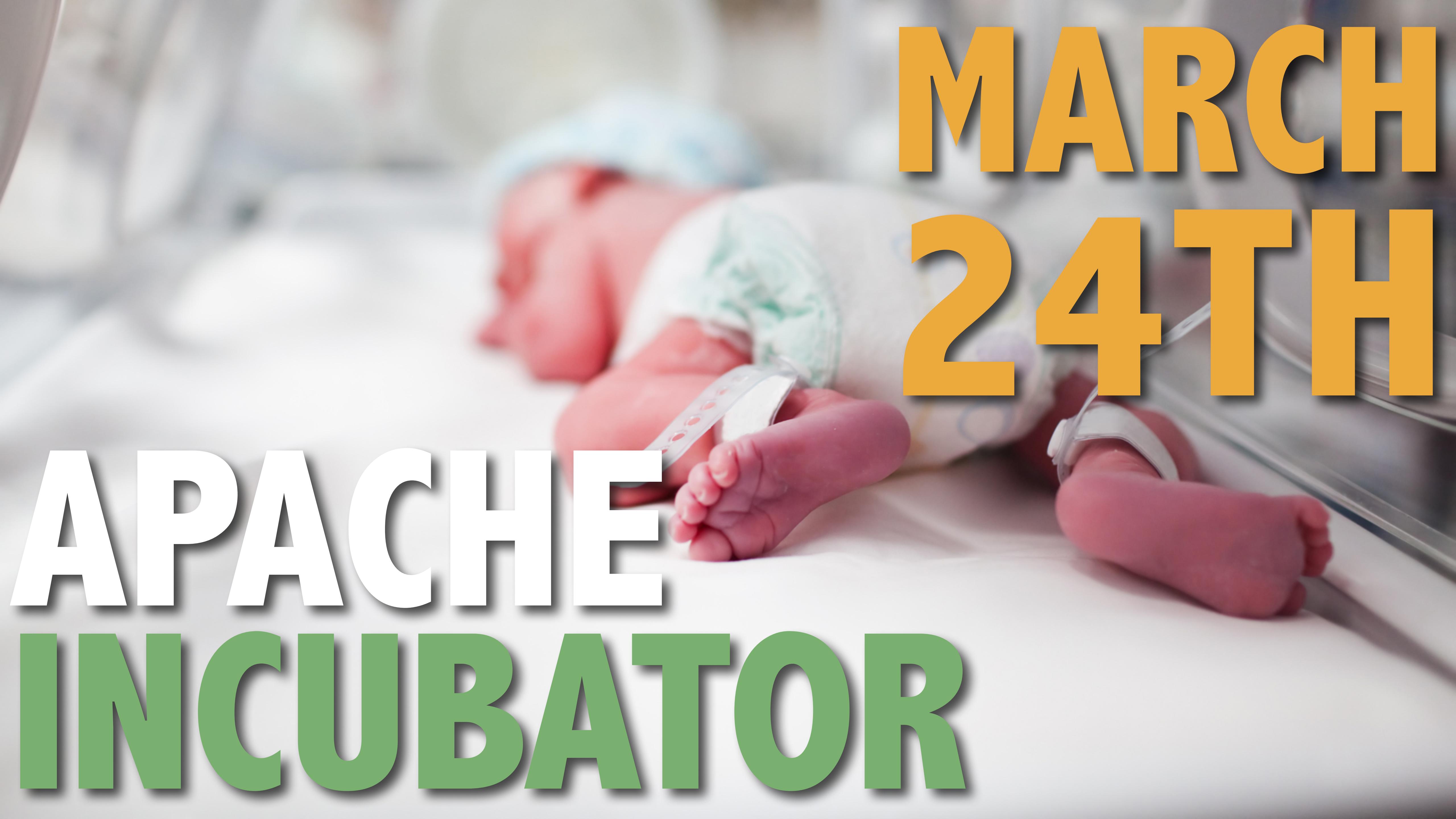
COMMITMENT  
ABOVE ALL

# APACHE INCUBATOR



MARCH  
24TH

APACHE  
INCUBATOR





# Incubation — mentors

- Emmanuel Lécharny
- Bertrand Delacrétaz
- Roman Shaposhnik
- Jim Jagielski
- Andrew Bayer



# Incubation — initial committers

- Paul King
- Cédric Champeau
- Pascal Schumacher
- Jochen Theodorou
- Guillaume Laforge



# Incubation — infrastructure

- **Mailing-lists** created
  - archived by Nabble
- **JIRA** issues imported
- Sources moved to Apache's **Git**
  - with a mirror on Github





# Incubation — new committers

- Officially joined the project as committer
  - Andrés Almiray
- To be announced soon too!
  - Dierk König
  - Russel Winder



# Incubation — next steps

- The next key step is **make a first release!**
  - following the Apache guidelines
- Made a first attempt June 7th,  
but still some things to tweak to be fully ready!



# Now what?

Restlet



# Groovy 2.4.4 / 2.5

- Improved compiler performance
  - with an ASM class reader
- `@Canonical` becomes a meta-annotation
- More control on the meta-annotation collector
- New `@MapConstructor` transformation
- Property validation in transformation parameters
- Further AST transformation improvements

G<sub>2</sub> R<sub>1</sub> A<sub>1</sub> T<sub>1</sub> E<sub>1</sub> F<sub>4</sub> U<sub>1</sub> L<sub>1</sub>

O<sub>1</sub>

O<sub>1</sub>

Bootstrapping







**Mikeal Rogers**

@mikeal



 Follow

The hardest part of open source is always getting people to participate and yet we continue to talk about almost everything but that.



...

RETWEETS

8

FAVORITES

3



8:24 PM - 8 Jun 2015



Reply to @mikeal



# Thanks for your attention



# Questions & Answers

Restlet