Pharo Status

Marcus Denker

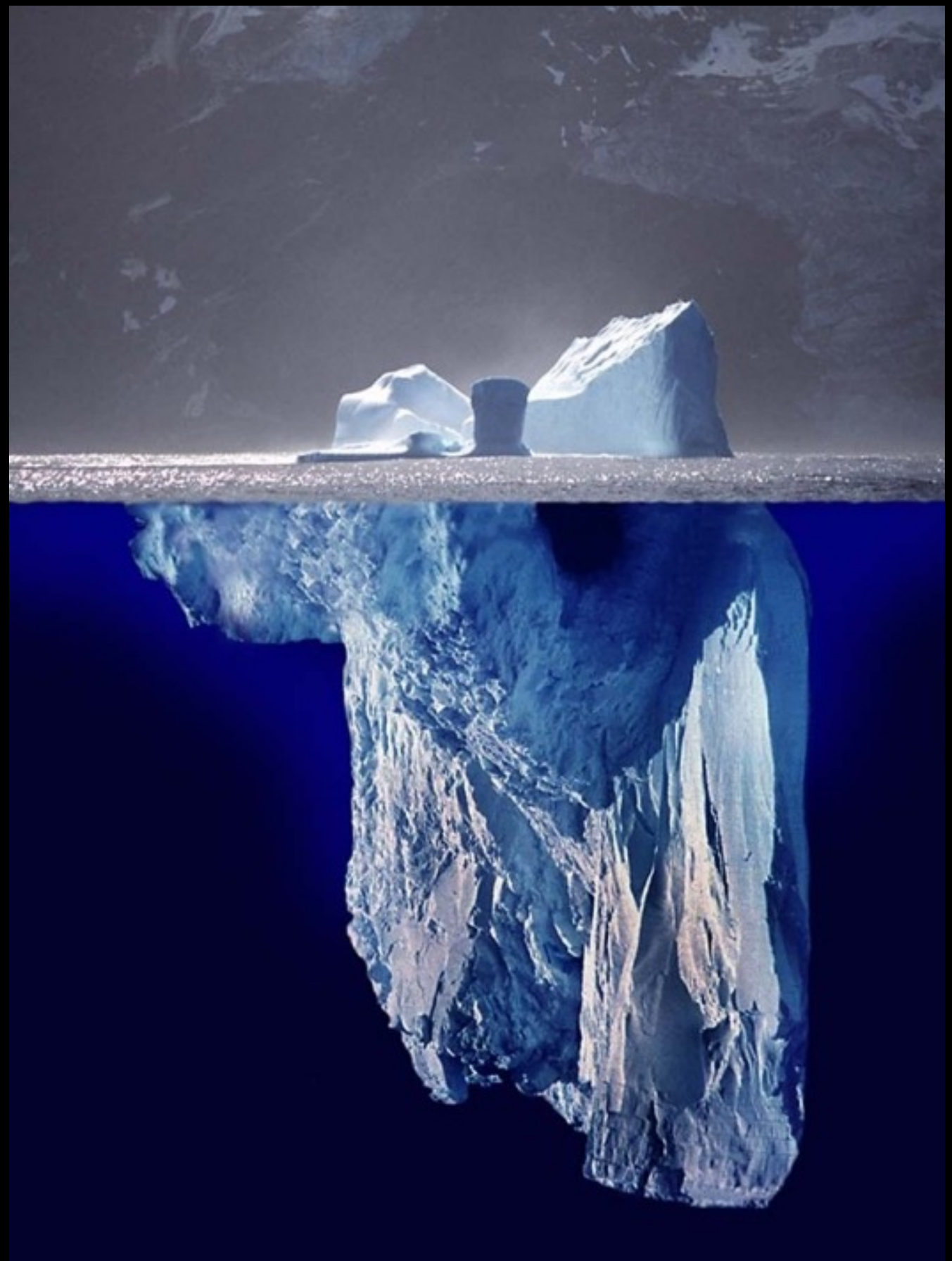
http://www.pharo-project.org

Pharo3: Release April `14

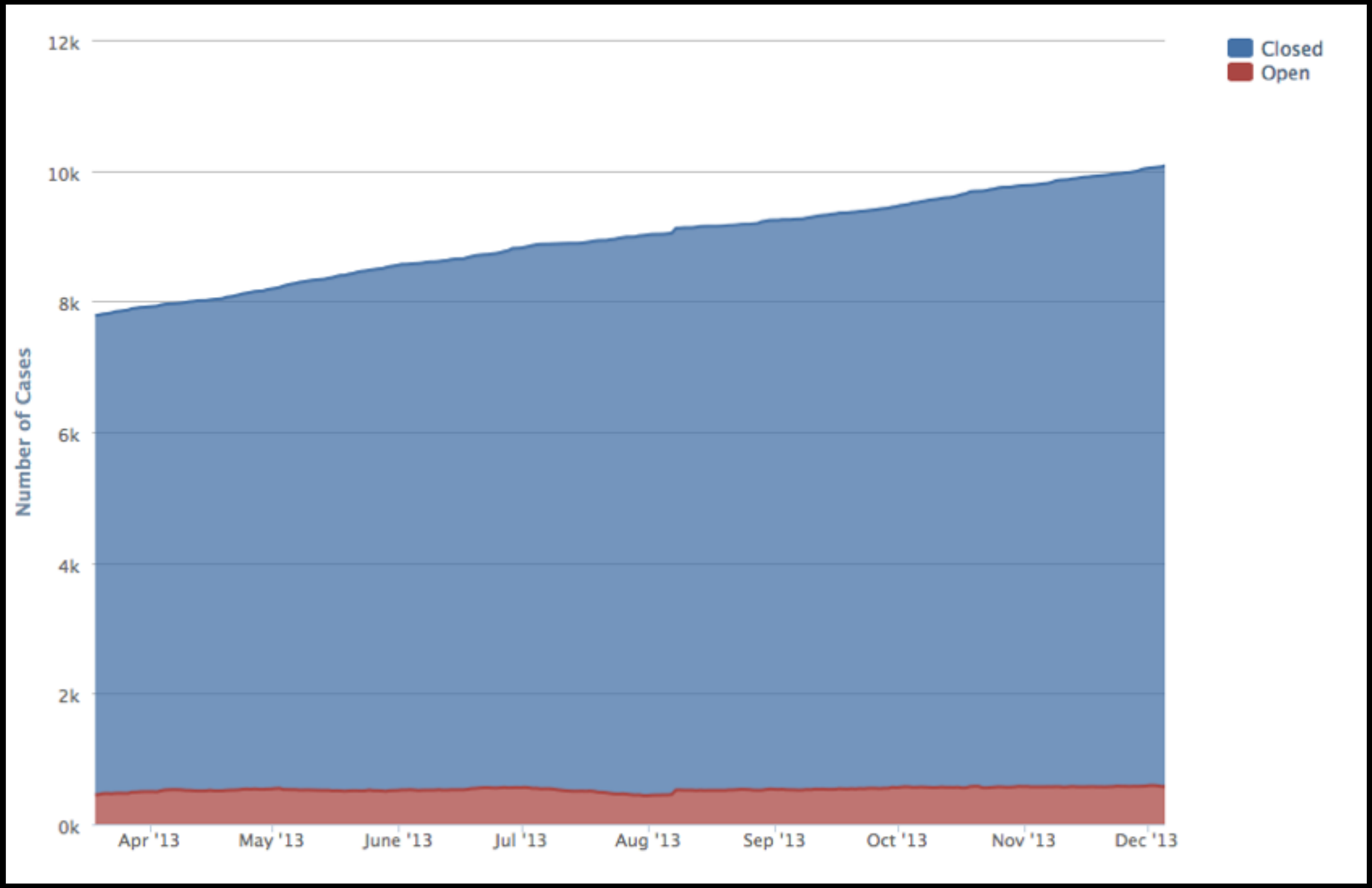
- Started March 2013
- 2390 Issue tracker entries with Pharo3 tag closed
- 854 Updates

Iceberg

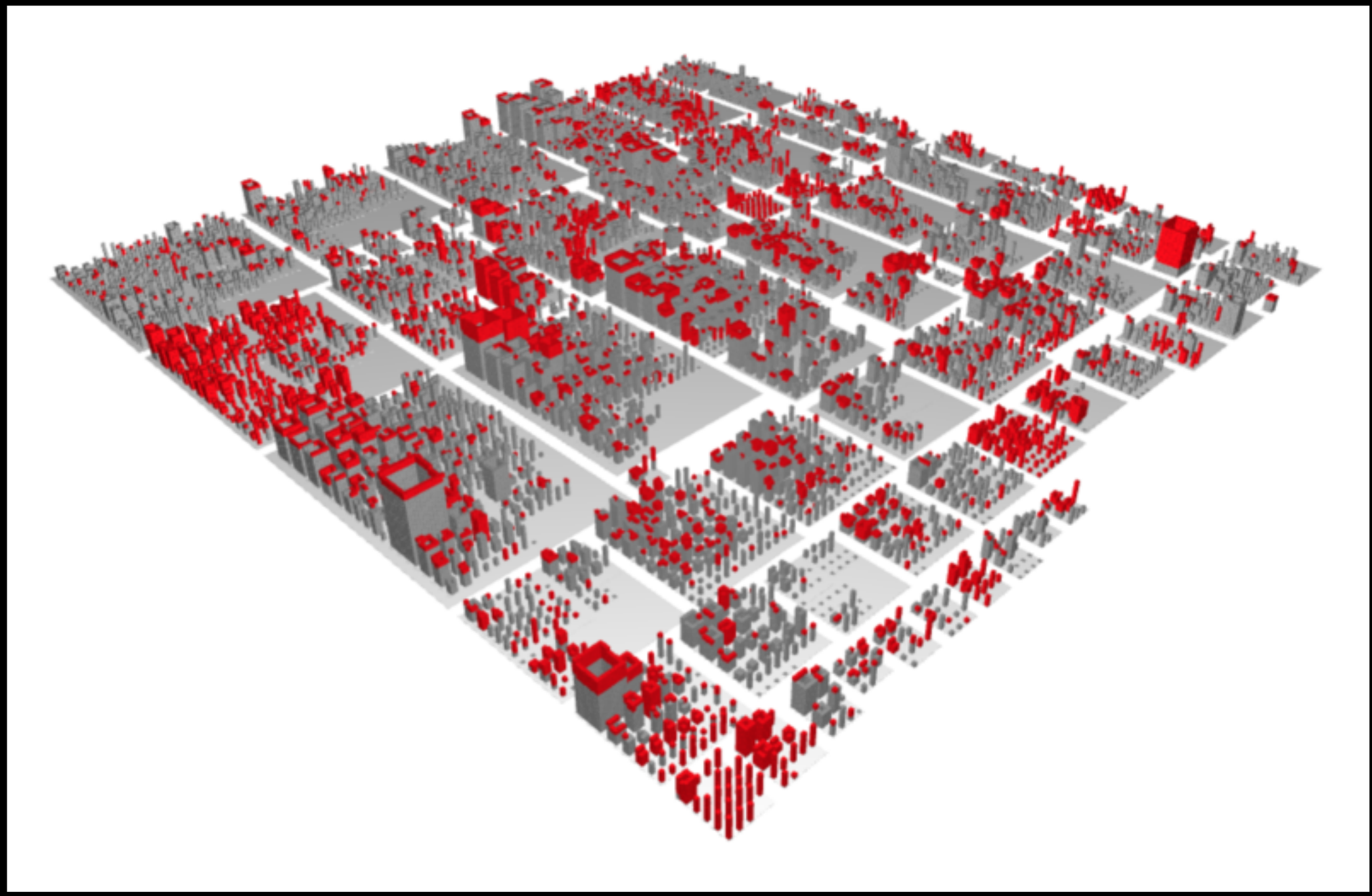
- A lot of Changes!
- Not everything visible



Lots of Activity



A lot of Change!



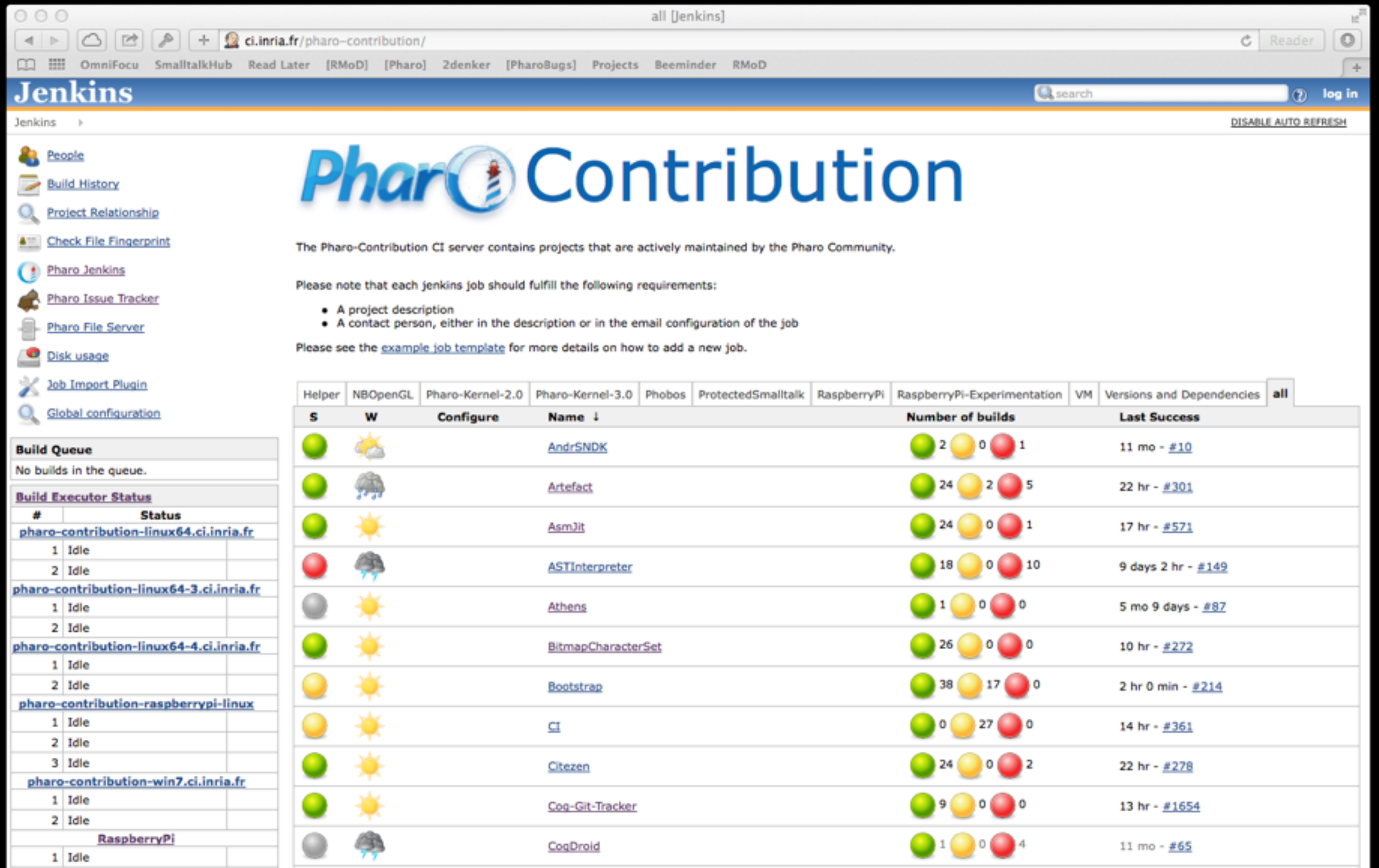
Yet easy to adopt

- Moose switched in two afternoons (two people)
- Others: “I just loaded my packages"

Infrastructure: CI

- https://ci.inria.fr is stable and used **a lot**
- Every fix is validated automatically before human review
- Every update triggers test run on 3 Architectures
- over 80 projects in pharo-contribution

ci.inria.fr/pharo-contribution/



Infrastructure: Misc

- http://files.pharo.org

- http://get.pharo.org

- SmalltalkHub: http://smalltalkhub.com
  - 1411 users, >1200 repos

Small Stuff

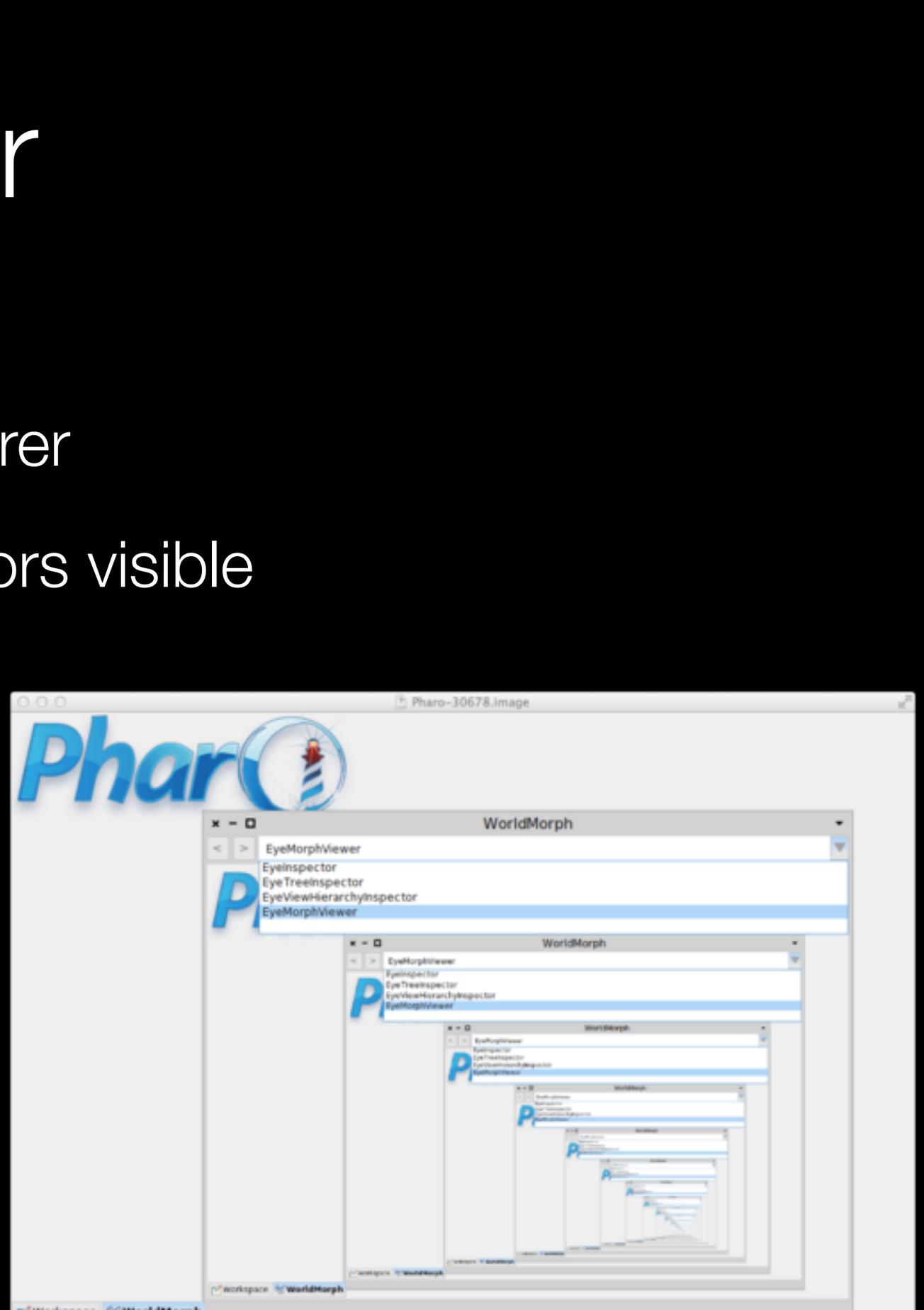
- Lots of Cleanups
- Lots of tuning (perfomance, memory)
- Lots of small improvements

Lots of larger things

- Closure class now standard in Pharo3
- Terminal output for stderr
- Cleanup Source file related code
- AST Interpreter
- AST based Navigation in Browser
- Komitter
- Launcher
- Font Speedup
- ….

New Inspector

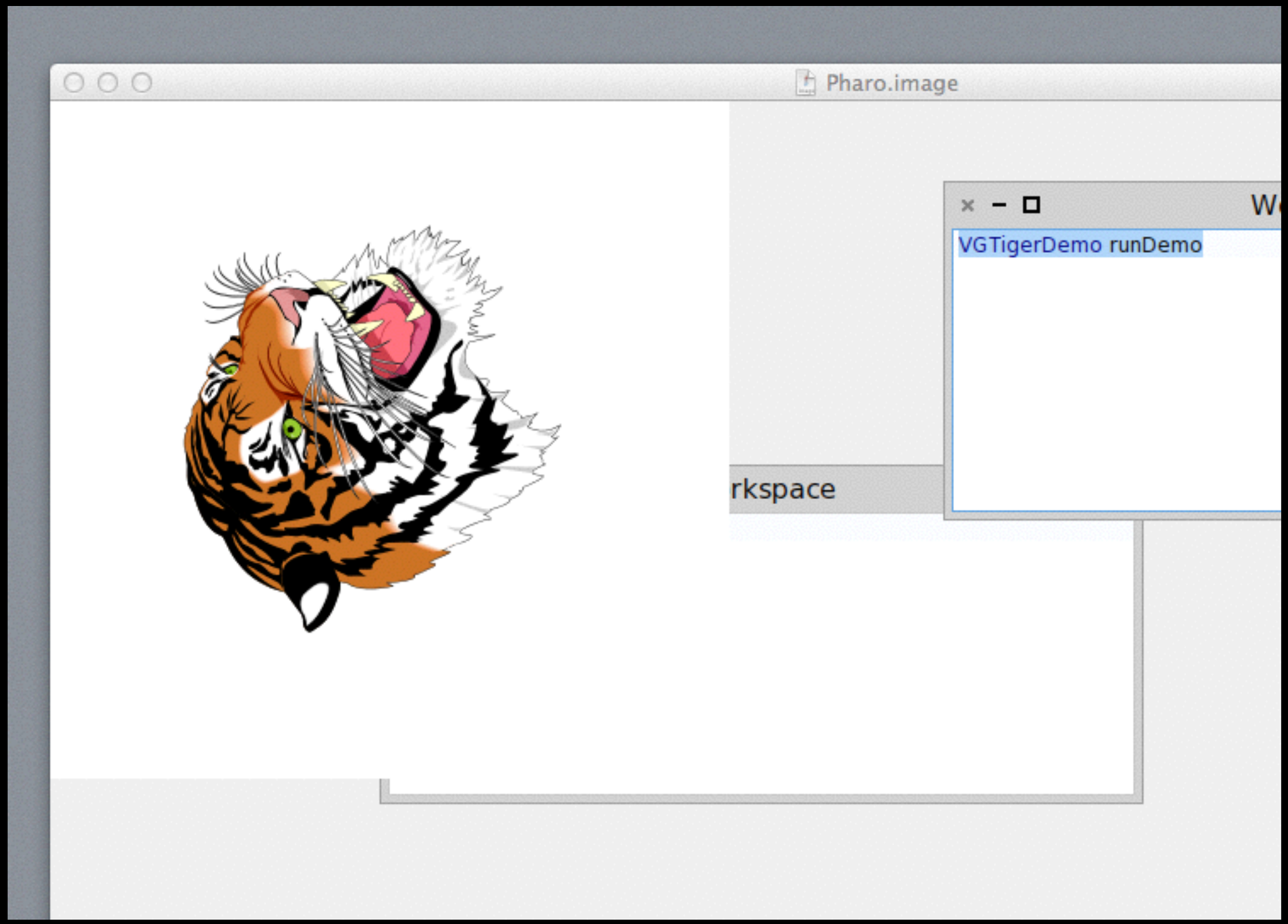
- Unify Inspector and Explorer
- Make specialised inspectors visible



Athens: Vector Graphics

- New API for Vector Graphics
- Independent of Backend
  - For now: Cairo
  - Balloon3D for Debugging
  - Future: OpenGL

Athens: Demo



Opal Compiler

- Uses RB AST
- IRBuilder: Bytecode backend with high-level builder
- Much easier to change
- Basis for advanced Reflection

New ClassBuilder

- Replaces the old ClassBuilder
- Easier to understand and more flexible
- Basis for First Class Variables (Slots)

New Debugger

- Model now separate from View
- Model is scriptable
- Debugger is extensible with Commands

Command Line

```
denker$ ./pharo Pharo.image --list
Currently installed Command Line Handlers:
    Fuel                Loads fuel files
    config              Install Configurations
    save                Rename the image and changes file
    update              Load updates
    printVersion        Print image version
    st                  Loads and executes .st source files
    test                A command line test runner
    clean               Run image cleanup
    eval                Directly evaluates one line scripts
```

A lot of change…

But just one iteration

Pharo4

Pharo4

- Again: To be released Spring 2015
- Already 175 updates
- 480 Issues closed
- Very stable

For example…

- Improved Refactorings
- 6MB Deployment Image
- ifTrue: on non-Booleans
- Browser and Tool cleanups
- Context Cleanup (MethodContext/ContextPart merge)

Pharo-20140602.image

Workspace

```
HP35CalculatorUIModel open.

HP35CalculatorUIModel new inspect; openWithSpec.
```

HP-35

3.141592653589793

x^y	LOG	LN	e^x	CLEAR
SQRT	ARC	SIN	COS	TAN
1/x	SWAP	ROLL	STO	RCL
ENTER	+/-	EE	CLx	
-	7	8	9	
+	4	5	6	
*	1	2	3	
/	0	.	π	

HP35CalculatorModel>>#pi

Type: Pkg1 | ^Pkg2 | Pk.*Cor

- HP35-Calculator
- HP35-Seaside-Calculat
- HelpSystem-Core
- HelpSystem-Tests
- History
- HudsonBuildTools20
- jQuery-Core
- jQuery-UI
- Javascript-Core
- Javascript-Pharo20-Cor

- HP35CalculatorModel
- HP35CalculatorTests
- HP35CalculatorModelTes
- HP35CalculatorUIModelT
- HP35CalculatorUIModel
- RPNCalculatorCore
- RPNCalculatorCoreTests

-- all --
accessing
copying
initialize
keys
printing
private
testing

one
pi
power
rcl
reciprocal
rollDown
seven
sin
six
sqrt

Groups | Hierarchy

```
pi
    self autoEnter.
    core x: Float pi.
    self resetInputAutoEnter
```

RPNCalculatorCoreTests>>#testDivision

Type: Pkg1 | ^Pkg2 | Pk.*Cor

- HP35-Calculator
- HP35-Seaside-Calculat
- HelpSystem-Core
- HelpSystem-Tests
- History
- HudsonBuildTools20
- jQuery-Core
- jQuery-UI
- Javascript-Core
- Javascript-Pharo20-Cor

- HP35CalculatorModel
- HP35CalculatorTests
- HP35CalculatorModelTes
- HP35CalculatorUIModelT
- HP35CalculatorUIModel
- RPNCalculatorCore
- RPNCalculatorCoreTests

-- all --
running
testing

- setUp
- testAddition
- testArcSinCosOutOfRan
- testClear
- Run the test rcCos
- testDivision
- testDivisionByZero
- testEmpty
- testExpLn
- testLog

Groups | Hierarchy | Class side | Comments

RPNCalculatorCoreTests >>#testDivision

```
testDivision
    core x: 100; enter; x: 4; divide.
    self assert: core x equals: 25
```

56

Test Finished
Method:
RPNCalculatorCoreTests>>#testDivision

HP35CalculatorModel

< > EyeTreeInspector

- root: a HP35CalculatorModel('3.141592653589793' a RPNCalculatorCore(3
  - core: a RPNCalculatorCore(3.141592653589793 0 0 0)
  - memory: 0
  - input: ''
  - inputState: #new
  - arcMode: false
  - autoEnter: true
  - error: nil

```
self reciprocal
```

Test Runner

Hit return to accept | Hit return to accept

- Seaside-Tests-RenderLoop
- Seaside-Tests-Environmen
- Seaside-Tests-Pharo-Funct
- Seaside-Tests-Pharo-Conti
- Seaside-Tests-Welcome
- Bootstrap-Tests-Core-Canv
- Bootstrap-Tests-Core-Libra
- Bootstrap-Tests-Widgets-C
- MemcachedTests
- Beacon-Tests
- SystemLogger-Tests
- SystemLogger-TestsExtens
- Zinc-WebSocket-Tests
- Zinc-SSO-OAuth1-Tests
- Zinc-SSO-OpenID-Tests
- FuelBeacon-Tests
- HP35-Calculator
- HP35-Seaside-Calculator

- *HP35CalculatorTests*
- HP35CalculatorModelTests
- HP35CalculatorUIModelTes
- RPNCalculatorCoreTests

118 run, 118 passes, 0 skipped, 0 expected failures, 0 failures, 0 errors, 0 unexpected passes

Run Selected | Run Profiled | Run Coverage | Run Failures | Run Errors | File out results

Workspace | Workspace | HP-35 | HP35CalculatorM... | HP35CalculatorM... | RPNCalculatorCo... | Test Runner

In Progress…

First Class Variables

- First class Instance Variables (Slots)
- First class globals + class variables

For what?

- Allows programmers to define behavior
- Easy reflection on variable access
    - Break on variable read, for example

Property Slots

```
Object
    subclass: #PropertyObject
    layout: PointerLayout
    slots: {
     #ivar.
    #property1 => PropertySlot.
    #property2 => PropertySlot.
      ...
    #propertyN => PropertySlot.
}
```

Property Slots

```
Object
    subclass: #PropertyObject
    layout: PointerLayout
    instanceVariables: {
     #ivar.
    #property1 => PropertySlot.
    #property2 => PropertySlot.
      ...
    #propertyN => PropertySlot.
}
```

Examples

- BitSlot
- BooleanSlot
- Alias
- Relationships (e.g. one-one, one-many)
- …. Your Domain level Slot! ==> Magritte

More in Paper from OOPSLA

Flexible Object Layouts

Enabling Lightweight Language Extensions by Intercepting Slot Access

Toon Verwaest Mircea Lungu
Oscar Nierstrasz

Software Composition Group, University of Bern, Switzerland

http://scg.unibe.ch

Camillo Bruni

RMoD, INRIA Lille - Nord Europe, France

http://rmod.lille.inria.fr

Abstract

Programming idioms, design patterns and application libraries often introduce cumbersome and repetitive boilerplate code to a software system. Language extensions and external DSLs (domain specific languages) are sometimes introduced to reduce the need for boilerplate code, but they

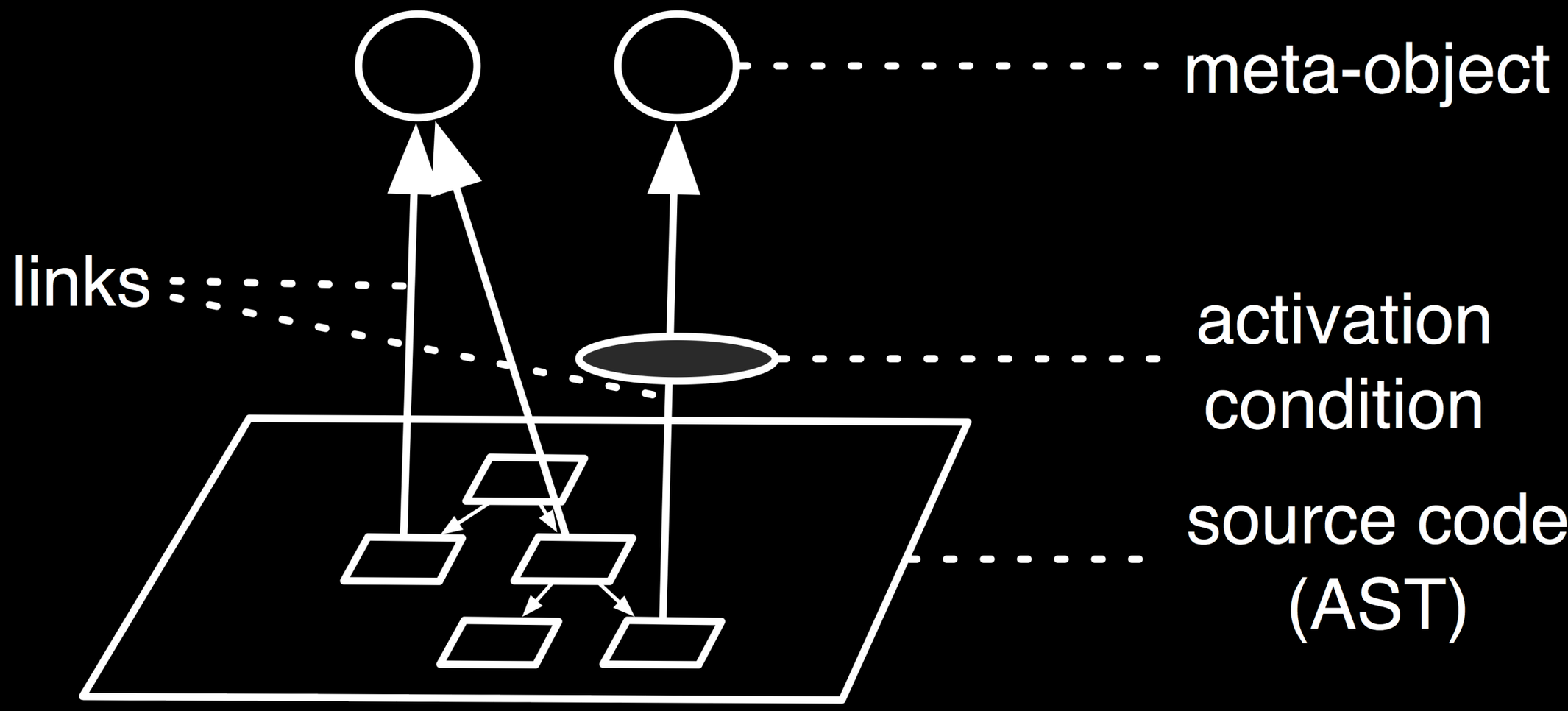
1. Introduction

Object-oriented programming languages (OOPL) are hi
effective as modeling languages. Features including cla
and inheritance can be used to model concepts at a
level of abstraction, normally leading to compact and
cise code. Unfortunately there are many situations in w

Advanced Reflection

- Partial Behavioral Reflection
- Associate MetaObject with structural object
  - Slots, Globals
  - AST nodes

The Meta Link



Why?

- Change behaviour for selected AST Nodes or Variables
- “All variable reads”
- “this message send”

But without changing the program code!

Uses...

- Debugger
  - BreakPoints, WatchPoints
- Profilers
- Coverage Analysis
- AOP

One File Pharo

- .sources, .changes. .image
- It is time to simplify that!

Epicea

- Replace .changes
- High level model:
  - aggregate changes (refactoring)
  - serialized to disk independent of source model

Epicea Log Browser

Prior+Trigger view

Expand all

Event

more

Undo 24/1/2014 10:34

ExampleClass >> #fortyTwo 24/1/2014 10:34

Undo 24/1/2014 11:06

ExampleClass >> #fortyTwo 24/1/2014 11:06

#fortyTwo ---> #forty2 24/1/2014 11:07

ExampleClass >> #forty2 24/1/2014 11:07 This is a comment

ExampleClass >> #fortyThree 24/1/2014 11:07

ExampleClass >> #fortyTwo 24/1/2014 11:07

'I didn''t like the name' 24/1/2014 11:08

'' 24/1/2014 11:09

'''comment '' trim' 24/1/2014 12:52

EpEntryItem >> #displayWidget 24/1/2014 12:53

MC save: Epicea-MartinDias.470 on: http://smalltalkhub.com/mc/MartinDias/Epicea/main/ 24/1/2014 12:55

Snapshot: /Users/tinchodias/Downloads/Epicea/Epicea-1.image 24/1/2014 12:55

'312312321' 24/1/2014 13:41

Content Filters

```
"protocol: #example"

fortyThree  ^ self fortyTwo + 1
```

```
"protocol: #example"

fortyThree  ^ self forty2 + 1
```

Sources

- It is 2014: Memory is cheap.
- Complexity is expensive
- Why not just put the sources in the image?
  - Externalize when needed (small devices)
  - Code history is in Monticello (or Git)

Bootstrap

- Create an image from a git repository
- Control what the image contains
- Easier to make changes
- Enforces Modularity

Boostrap

- Working for Pharo3 as a prototype
- Can we even use this for Pharo4 on the build server?

And more…

- GT Tools
- VM related news
  - Spur, Sista, 64bit…
  - ……. there are lots of talks here

Questions ?