



The present and
future of Pharo

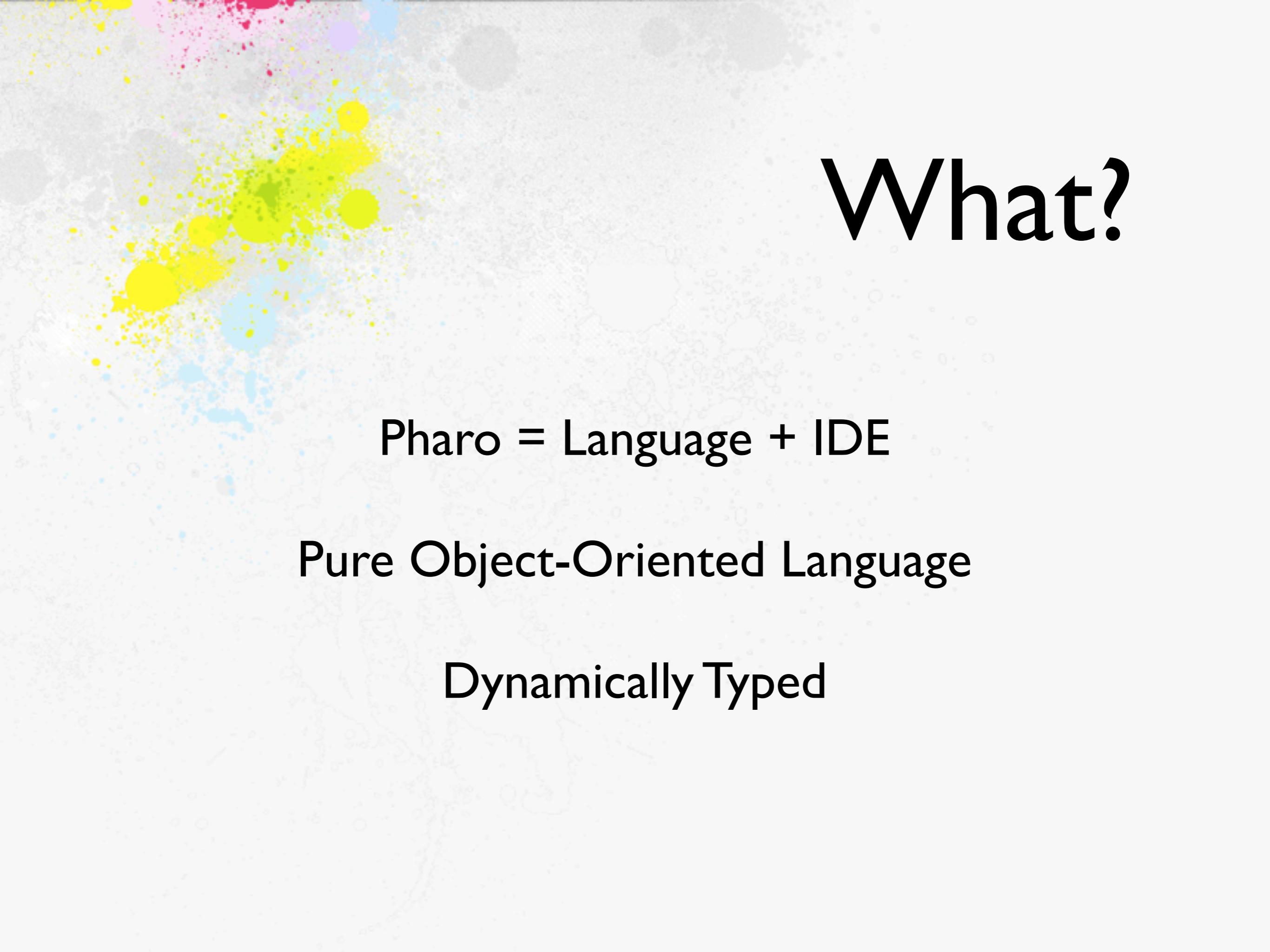


What?

A progressive, open-source Smalltalk platform
for professional use.

What?

A flexible environment to support the research
of new language concepts.

The background of the slide features a subtle, abstract pattern of colored dots in shades of yellow, blue, and pink, scattered across the top left corner.

What?

Pharo = Language + IDE

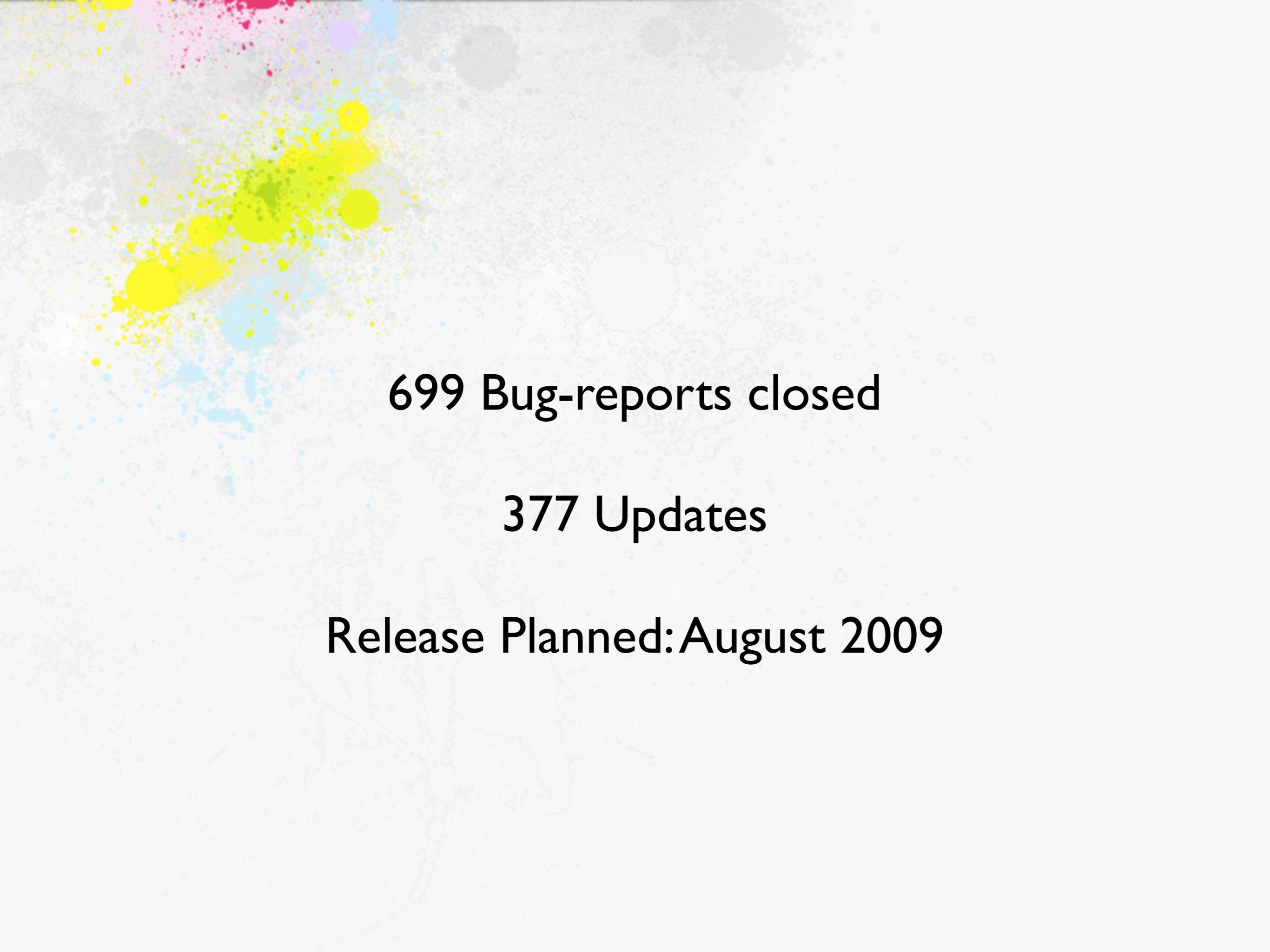
Pure Object-Oriented Language

Dynamically Typed

History

Based on Squeak Smalltalk

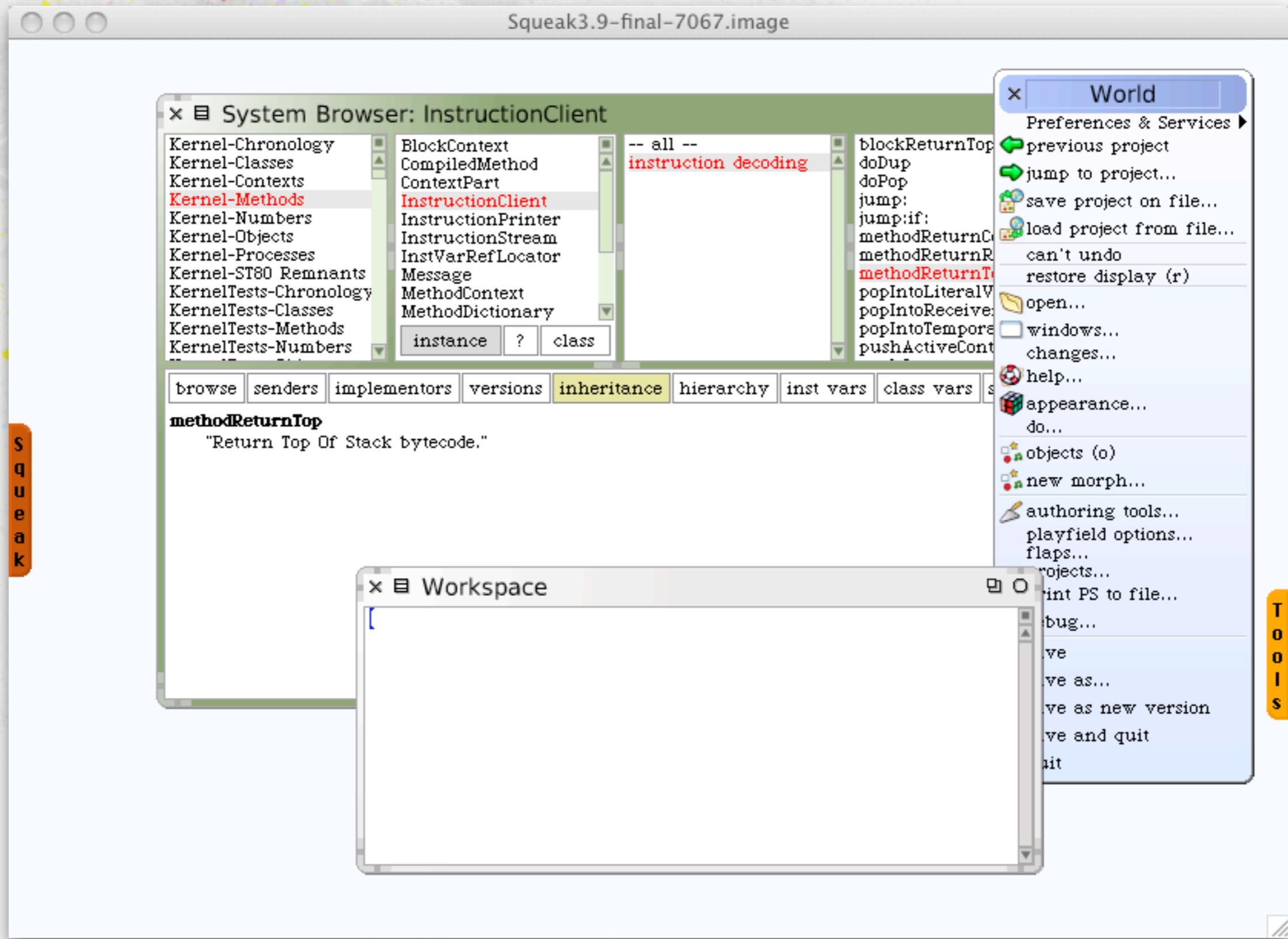
- + Major Cleanups (MVC, eToys)
- + New UI Look / TrueType
- + Tools
- + Block Closures
- + **Lots** of bugfixes and small improvements



699 Bug-reports closed

377 Updates

Release Planned:August 2009





OB Package Browser: AutomaticMethodCategorizerApplier

C... (class search), i... (implementor search), #C... (class ref) << hist. >>

- Recently Modified -

- ImageForDevelopment
- AutomaticMethod
- AutomaticMethod
- AutomaticMethod
- Model
- Tests

AutomaticMethodCategorizerApplier -- all --
AutomaticMethodCategorizerApplier event handling
AutomaticMethodCategorizerApplier event handling - private
AutomaticMethodCategorizerApplier initialization
AutomaticMethodCategorizerApplier testing
AutomaticMethodCategorizerApplier -- local (7) --
AutomaticMethodCategorizerApplier -- required (?) --
AutomaticMethodCategorizerApplier -- long (1) --

initializeUsing:onCharacter
registerForNotification

DF 3/20/2006 21:15 - AutomaticMethodCategorizerApplier - initialization - 4

registerForNotifications

```
changeNotifier
    notify: self
        ofSystemChangesOfItem: AbstractEvent methodKind
        change: AddedEvent changeKind using: #methodAdditionNotifiedWith:;
    notify: self
        ofSystemChangesOfItem: AbstractEvent methodKind
        change: ModifiedEvent changeKind using: #methodModificationNotifiedWith:;
```

World

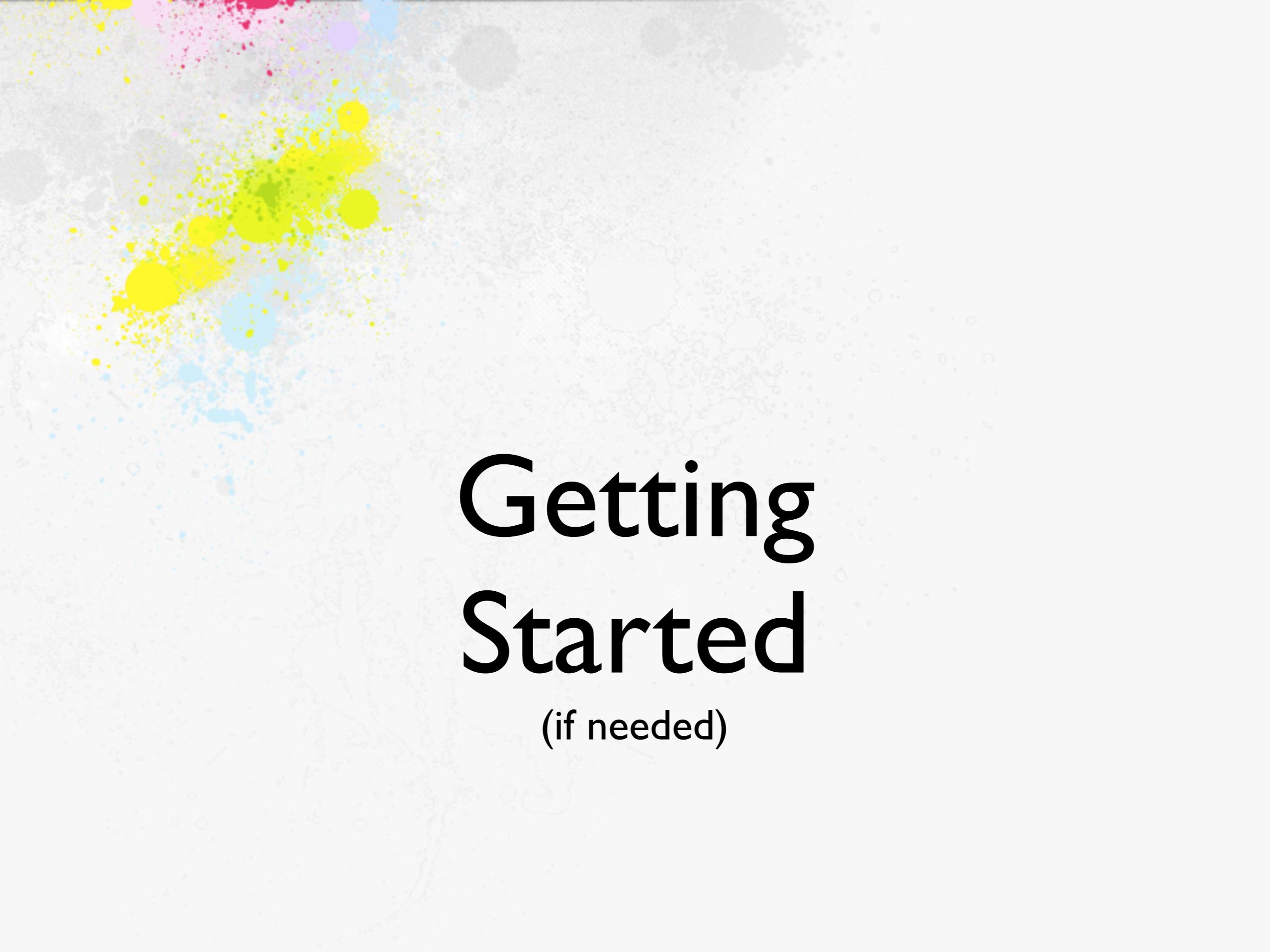
- Class Browser
- Workspace
- Test Runner
- Monticello Browser
- Tools
- Windows
- Debug
- System

Save

Save As...

Save and quit

Quit



Getting Started

(if needed)

seaside

The fast lane to HTTP

About

[Screenshots](#)

[Success Stories](#)

[Examples](#)

[What others think](#)

[Hosting](#)

[Support](#)

[Trivia](#)

[Merchandise](#)

[more](#)

Documentation

[FAQ](#)

[Tutorials](#)

[Migration](#)

[Videos](#)

[more](#)

Community

[Weblogs](#)

[Mailing Lists](#)

[Development](#)

[Contribute](#)

[Merchandise](#)

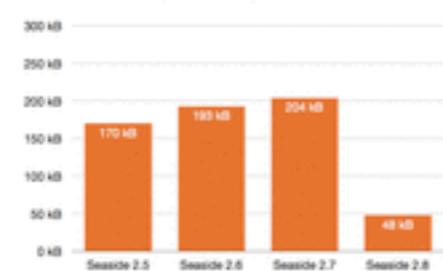
[Extensions](#)

[Projects](#)

[more](#)

Seaside 2.8

Memory Requirements



News

[Smalltalk Tidbits, Industry Rants] [Industry Misinterpretations 144: Children of Smalltalk](#) 12 July 2009

You'll have to listen to the entire podcast to understand where the title comes from :) This week ...

[London Geek Night](#) 10 July 2009

I have some pictures to share for people that missed the London Geek Nights with the title Seaside: ...

[Web Velocity released](#) 29 June 2009

Big news from Cincom on the Seaside mailing list today: they've made an evaluation version of their ...

[London](#) 23 June 2009

I'll be in London (UK) on Wednesday June 24 for 10 days or so. If anyone around there wants to meet ...

[Seaside 2.8.4 and Pharo](#) 5 June 2009

We just announced Seaside 2.8.4, the fourth maintenance release for Seaside 2.8. Again this release ...

[more](#)



Seaside is a free and [Open Source™](#) web application framework distributed under the [MIT License](#).

Seaside is available on the following Smalltalk platforms:

[Pharo Smalltalk \(download\)](#)

[Cincom Smalltalk](#)

[Dolphin Smalltalk](#)

[GemStone Smalltalk](#)

[GNU Smalltalk](#)

[VA Smalltalk](#)

[more](#)

[Alltop](#) [postini](#) [squeaknewcompiler](#) [iPhone Dev Center](#) [4-Seasons.TV](#)

Pier: Content Management System

+

[About](#) [Download](#) [Documentation](#) [Development](#) [Contact](#) [Blog](#)

Pier is a content management system that is light, flexible and free. It is light as in double click and go ([download it](#)). It is flexible as in make it be what you want ([learn more](#)). It is free as in freedom ([MIT license](#)).

Pier 1.2

11 July 2009

Pier 1.2 is now available.

What's new: * Mature value links (see Pier syntax for more details) * ...

July Pier sprint

11 July 2009

Yesterday, Lukas and me spent an intense afternoon working on Pier. We focused on closing the open ...

Pier sprint

5 April 2009

Pier-session.jpg Yesterday, Lukas and me sprinted intensively through Pier. It was quite exciting....

[Read more blog entries ...](#)



Seaside under GLASS
GemStone • Linux • Apache • Seaside • Smalltalk

Smalltalk with OO-Database

Pharo is the IDE



Future



...cleaner



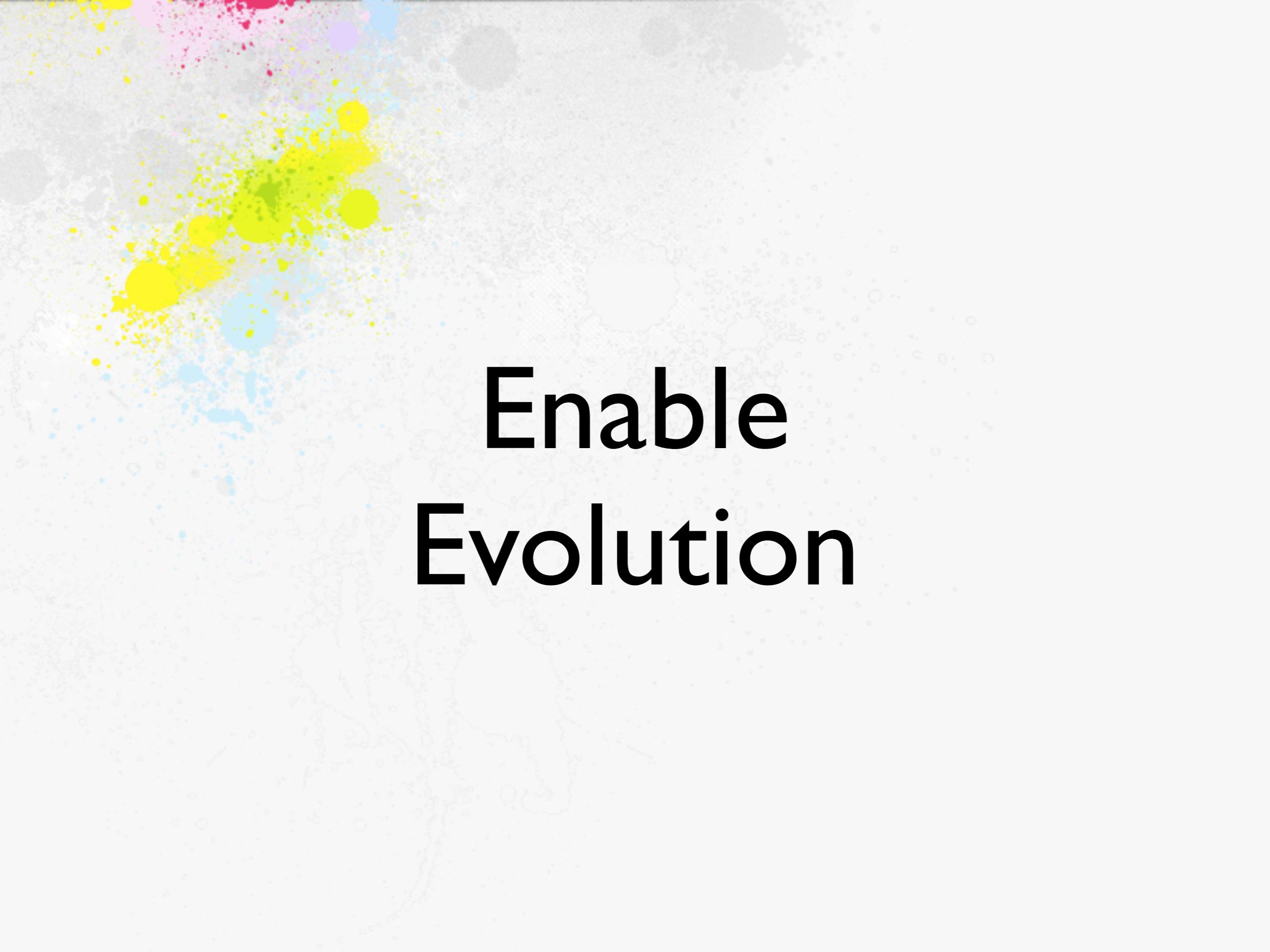
...smaller



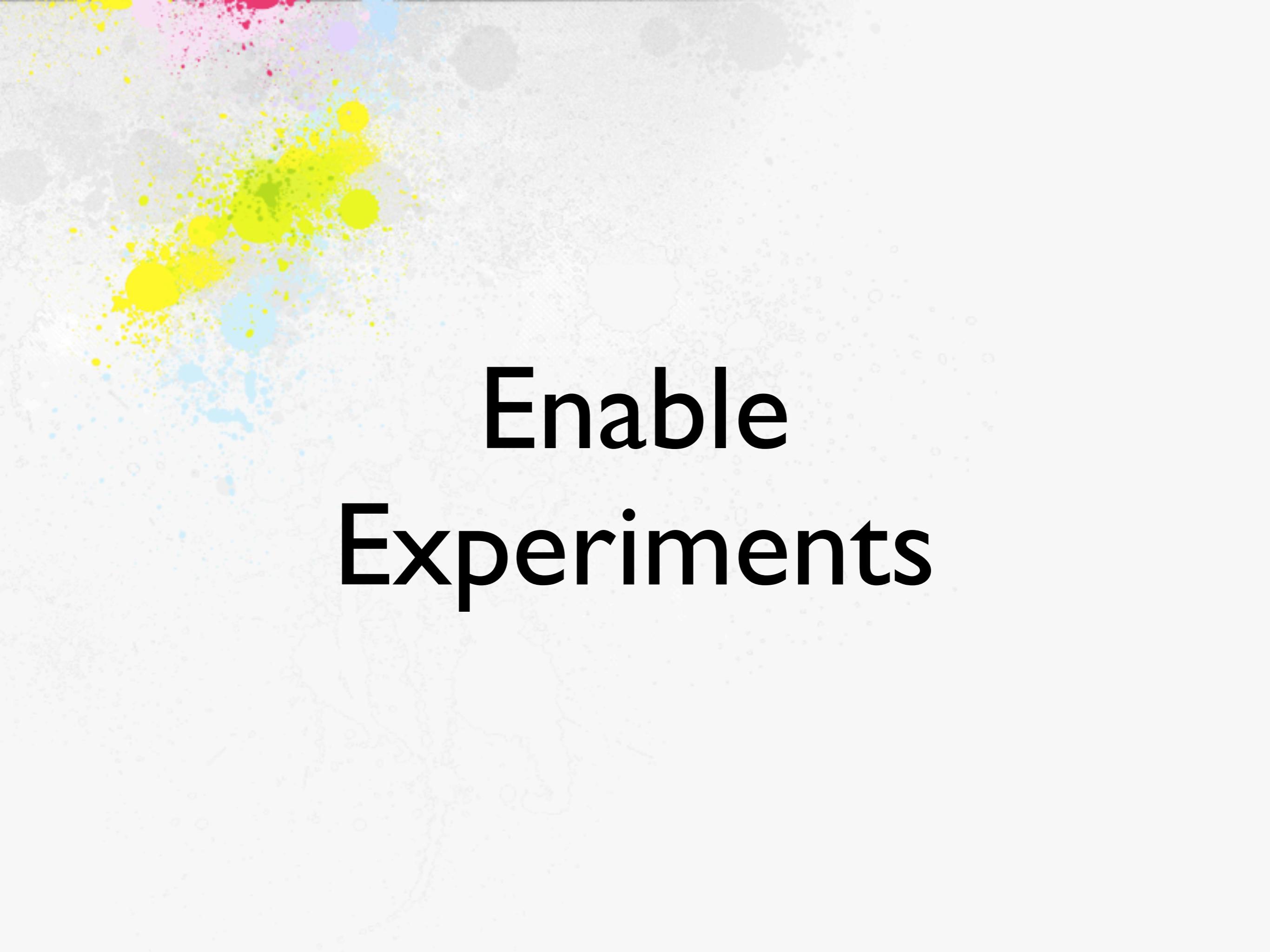
...faster



(of course)



**Enable
Evolution**



**Enable
Experiments**

Examples

- Compiler
- Slots
- JIT and AOStA

Old Compiler

- From 1976
- Not OO
- Hard to understand. Hard to change

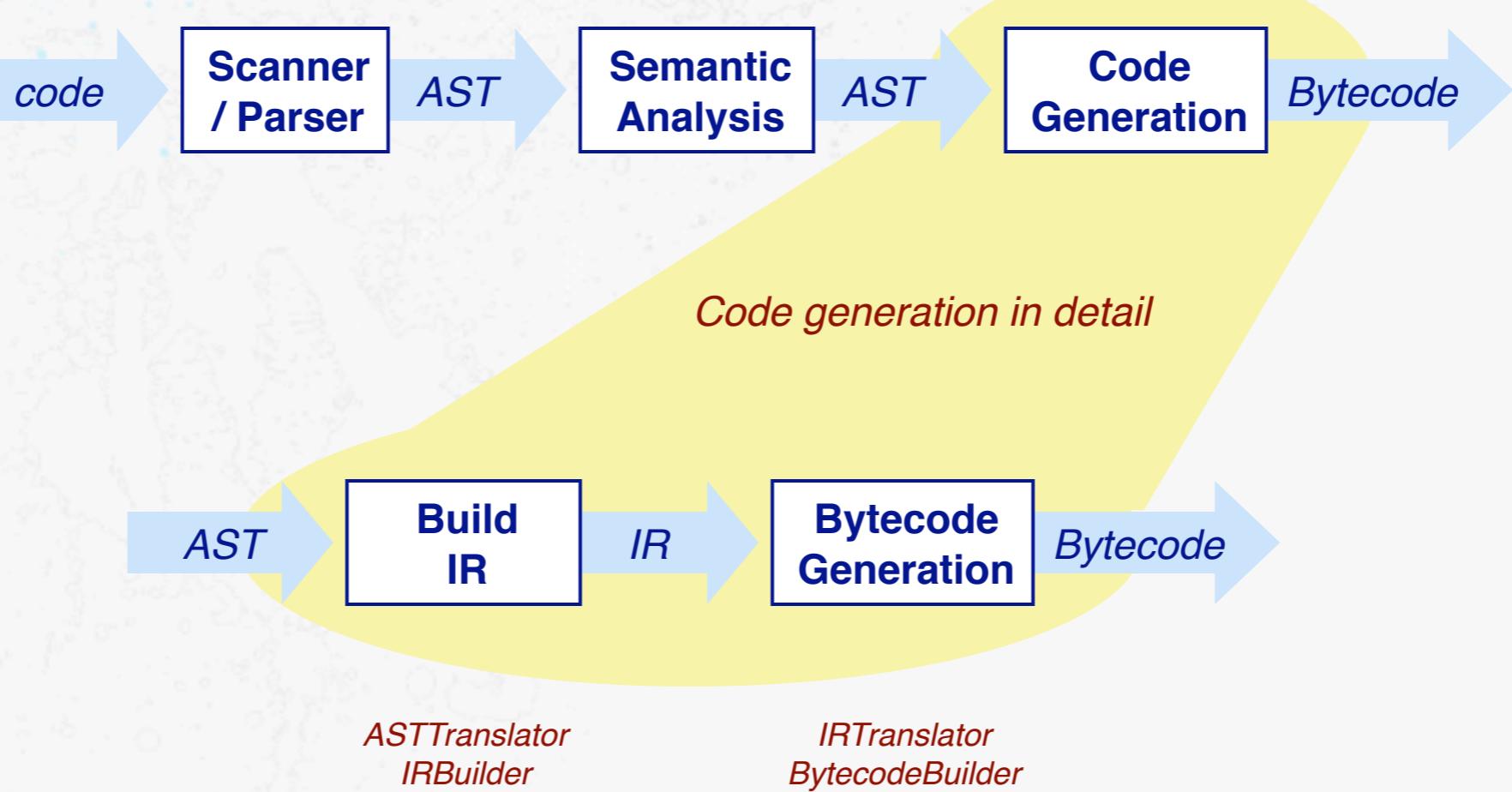
We need a better Compiler!

New Compiler

- Originally by Anthony Hannan
- Based on Visitors
- Reusable and Pluggable Backend (IRBuilder)
- Uses RB AST
- SmaCC Parser

Easier to understand.
Easier to change

New Compiler



Plans...

- New BlockClosure Format
- Plugin architecture: Parser, Checker, CodeGen
- Use RB Parser (Faster, Error messages)

Examples

- Compiler
- Slots
- JIT and AOStA

Instance Vars

- Not objects
- Change behavior?
- Change memory layout?

Instance state should be more powerful!

Slots

- Are objects
- Allow custom subclasses
- Change behavior and memory layout

Without Performance penalty!

Auto-Accessor

```
MopObject subclass: #SlotExample
  slots: '{AutoAccessorSlot named: #hello}'
  classVariableNames: ''
  poolDictionaries: ''
  category: 'SlotExamples'
```

Create accessor-methods at compile-time

Active Slot

```
MopObject subclass: #SlotExample
slots: '{ActiveSlot named: #hello
         action: [Beeper beep]}'
classVariableNames: ''
poolDictionaries: ''
category: 'SlotExamples'
```

Evaluate block on read

Sparse Slot

```
MopObject subclass: #SlotExample
slots: '{SparseSlot named: #a.
         SparseSlot named: #b.
         SparseSlot named: #c}'
classVariableNames: ''
poolDictionaries: ''
category: 'SlotExamples'
```

Store values in one Dictionary

iVar Slot

```
MopObject subclass: #SlotExample
  slots: '{IVarSlot named: #a}'
  classVariableNames: ''
  poolDictionaries: ''
  category: 'SlotExamples'
```

Normal instance Variable

Auto-Accessor

```
MopObject subclass: #SlotExample
  slots: '{AutoAccessorSlot named: #hello}'
  classVariableNames: ''
  poolDictionaries: ''
  category: 'SlotExamples'
```

Create accessor-methods at compiled-time

```
Slot subclass: #AutoAccessorSlot
instanceVariableNames: ''
....
```

```
postCreationAction
    self generateGetter.
    self generateSetter.
```

```
generateGetter
class compile:
    (String streamContents: [:stream | stream
        nextPutAll: self name; crtab;
        nextPutAll: '^', self name asString])
```

Active Slot

```
MopObject subclass: #SlotExample
slots: '{ActiveSlot named: #hello
         action: [Beeper beep]}'
classVariableNames: ''
poolDictionaries: ''
category: 'SlotExamples'
```

Evaluate block on read

Active Slot

```
AutoAccessorSlot subclass: #ActiveSlot  
instanceVariableNames: 'action'  
....
```

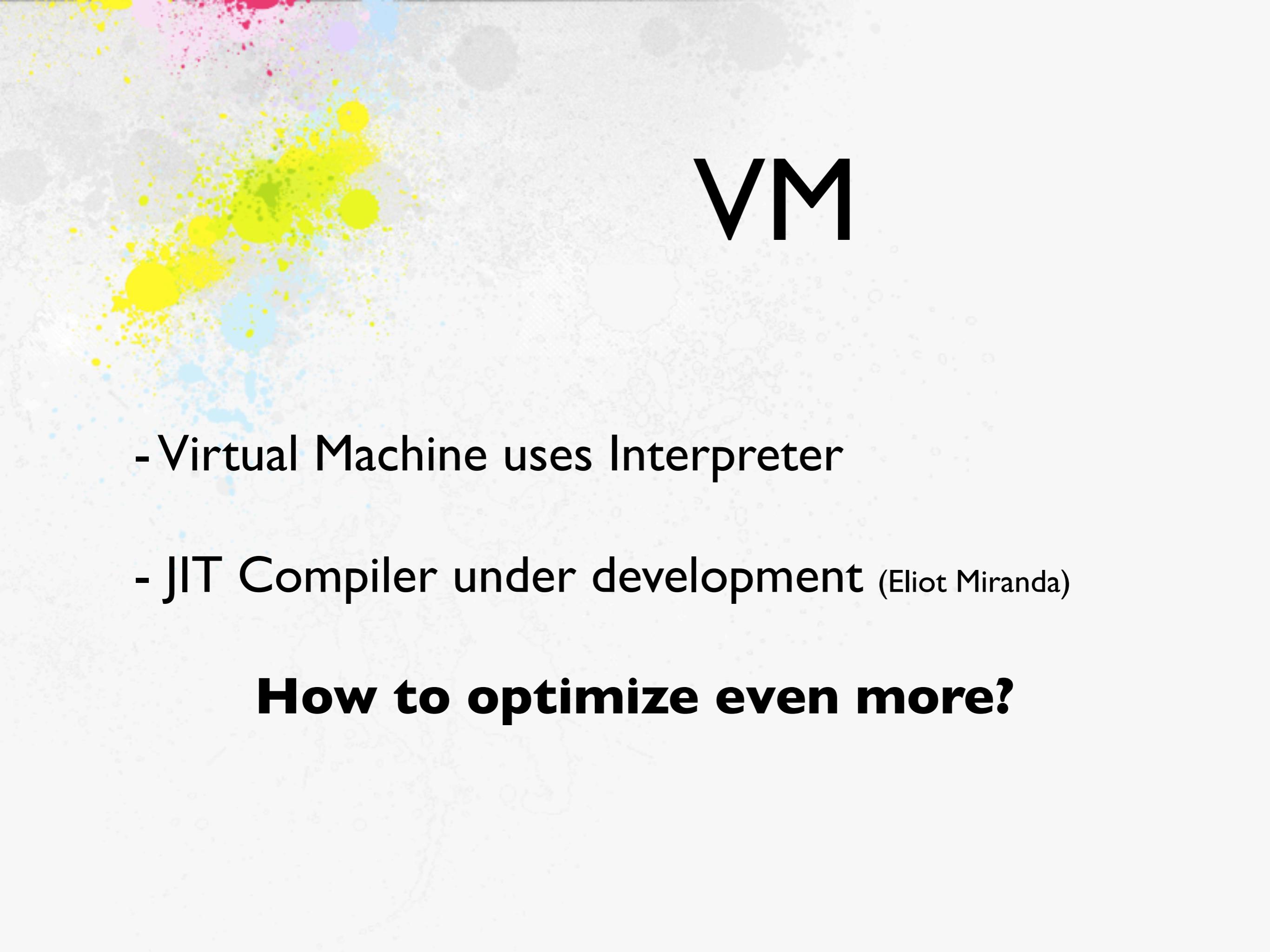
```
emitReadUsing: methodBuilder  
methodBuilder  
    pushLiteral: action;  
    send: #value.  
super emitReadUsing: methodBuilder
```

Status

- First Prototype
- Integration with ClassBuilder
- Compatibility with iVars
- Explore different designs

Examples

- Compiler
- Slots
- **JIT and AOStA**



VM

- Virtual Machine uses Interpreter
- JIT Compiler under development (Eliot Miranda)

How to optimize even more?

AOStA

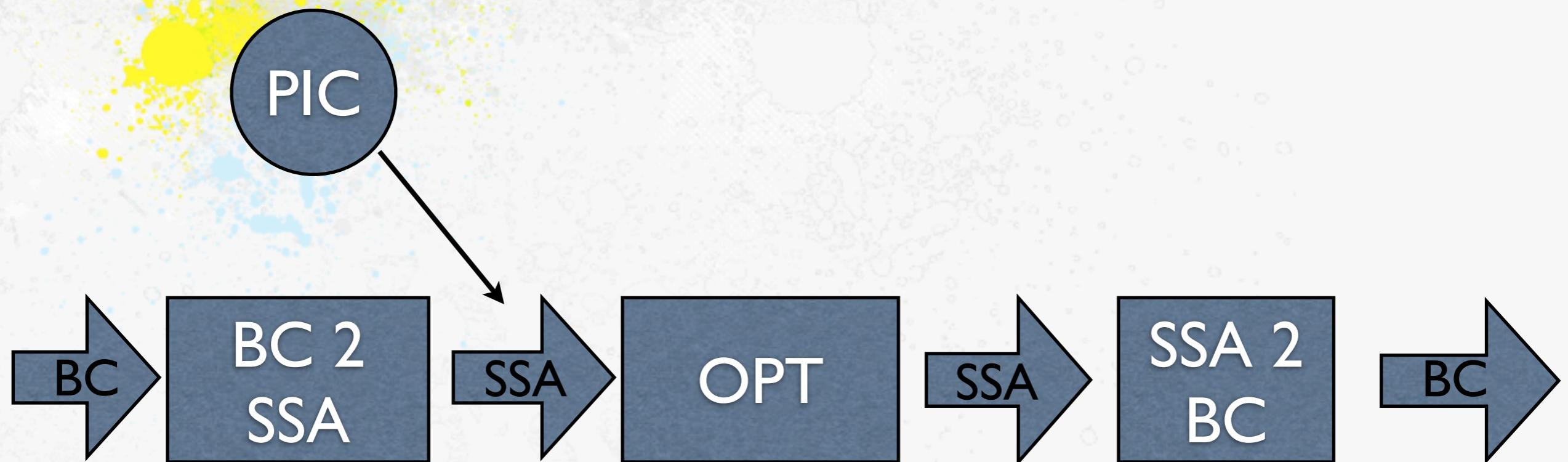
- Provide API for accessing runtime data
- PIC: Polymorphic Inline Cache

Record class per send instruction

- Hotness Counter

Where do we spend time?

AOSTA



Optimize SSA-Form
(Static Single Assignment)

AOStA

- Use PIC-Data for specialization
- Inline often-called methods
- Special Bytecode for primitive types (Floats)

AOStA

- Bytecode-to-Bytecode optimization
- Purely realized in Smalltalk
- Accessible in the image. Reflection?

More...

- Improve Tools
- Real Modules
- Reflection
-



Join Us!

Goal: learning and having fun

<http://pharo-project.org>