# PharoGs: Hosting Pharo in GemStone/S

James Foster

ESUG 2019 – Köln, Germany

Walla Walla University
SEVENTH-DAY ADVENTIST HIGHER EDUCATION

# Agenda

- **Motivation**
- GemStone Namespaces
- Hosting Process
- Demo
- IDE
- Summary and Questions

# Motivation

- "Now since people deploy on GemStone … I would love to have … Pharo fully execute with compiler and other on top of GS VM."
  - Stéphane Ducasse (1 June 2019)
  - http://forum.world.st/Re-Versioning-with-Iceberg-td5099792i20.html#a5099905

# "Develop in Pharo, deploy in GemStone"*

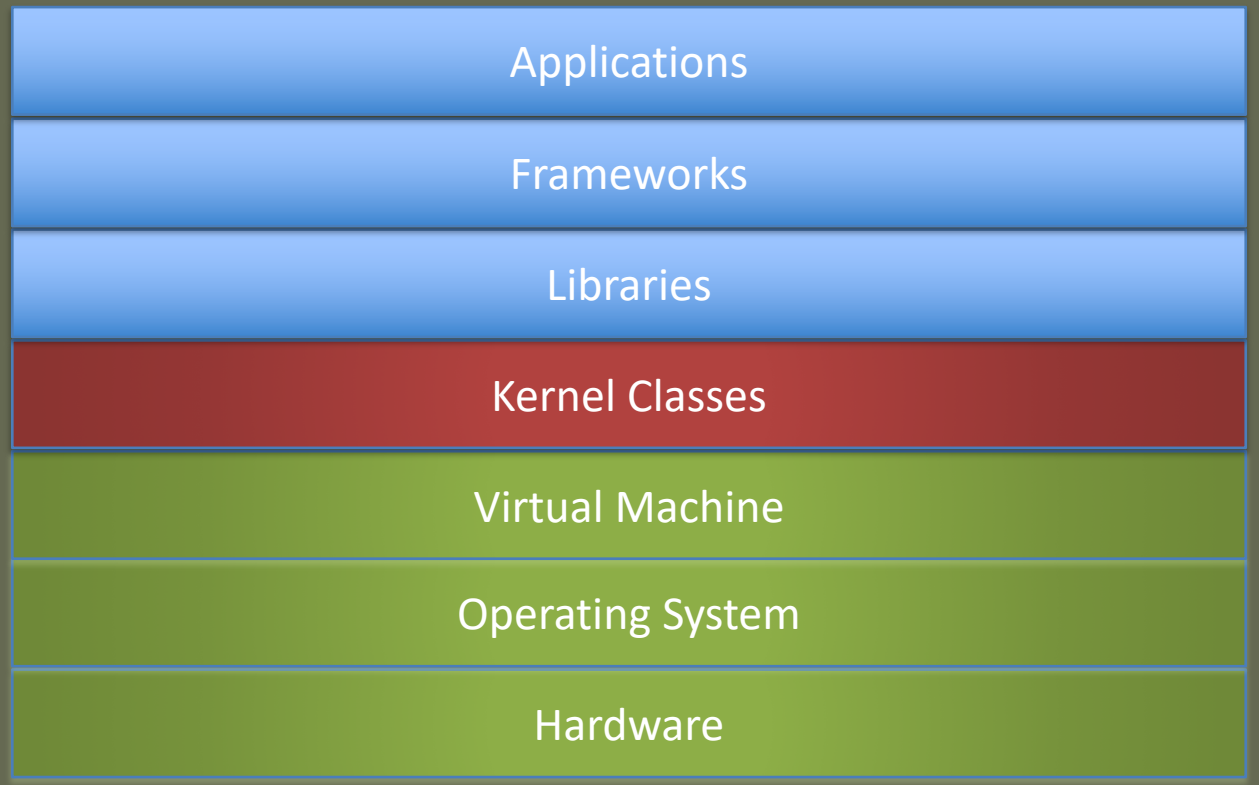
- Develop in Pharo
  - Use familiar and powerful IDE
- Deploy in GemStone
  - Scale-up image size (terabytes)
  - Scale-up concurrent sessions (thousands)
  - Shared object space with transactional persistence
  - Industrial-strength reliability

*Dale Henrichs, 2004

# Code Portability

- Grease (https://github.com/SeasideSt/Grease)
  - Limited to common features
  - Often an afterthought
  - Constant porting effort
- Why not the full* Pharo class library?

  * For some definition of "full"!



# Version/Dialect Cohabitation

- Current project: one "guest"
  - Host Pharo8.0 in GemStone 3.5
- Why not two or more "guests"?
  - Use a Pharo7.0 library from Pharo8.0 code!
- Analogous to the module/namespace problem

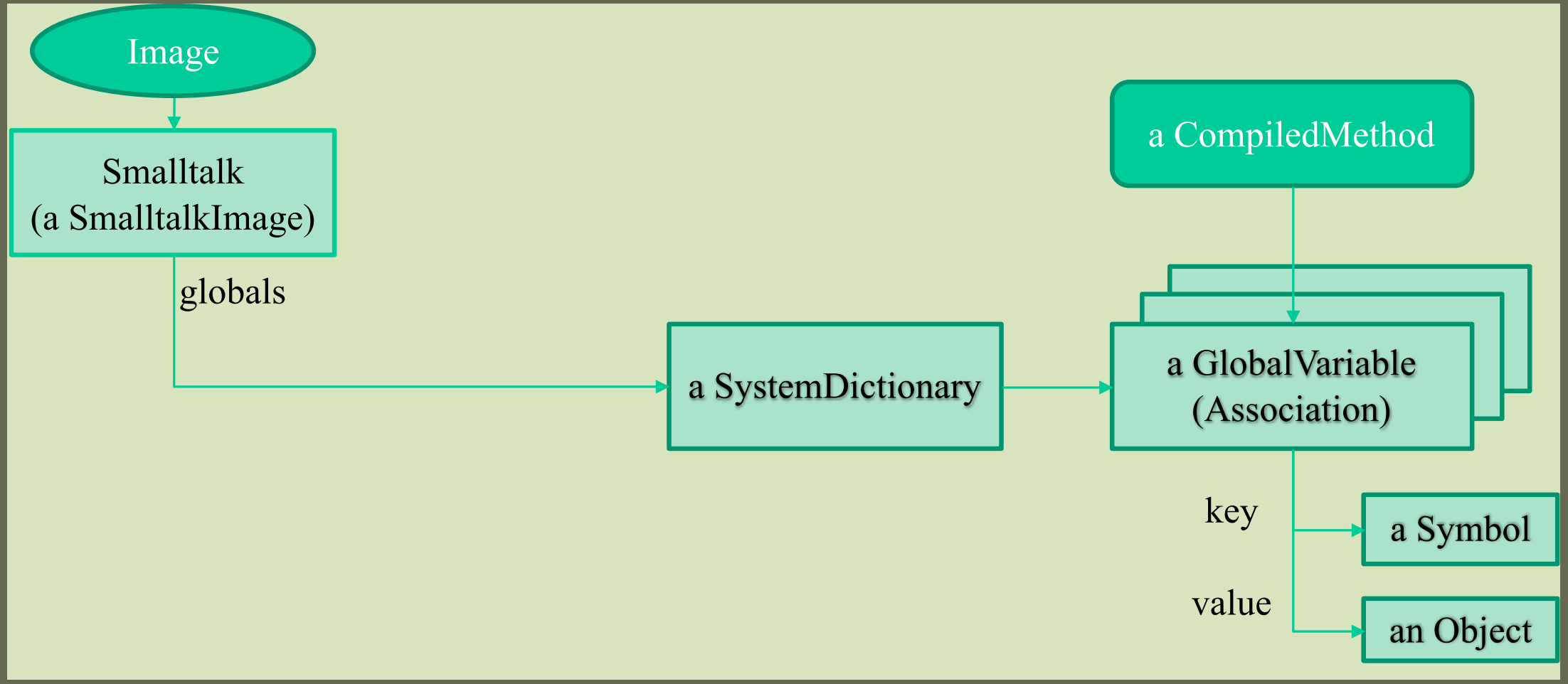
# IDE for a "Headless" Image

- Development for an image that does not have tools
  - Use GemStone code browser, debugger, inspector
- Research into how "minimal" we can make an image
  - No concerns about breaking image
- Research about what is truly "kernel"
  - What would be minimal, stable API for base class library

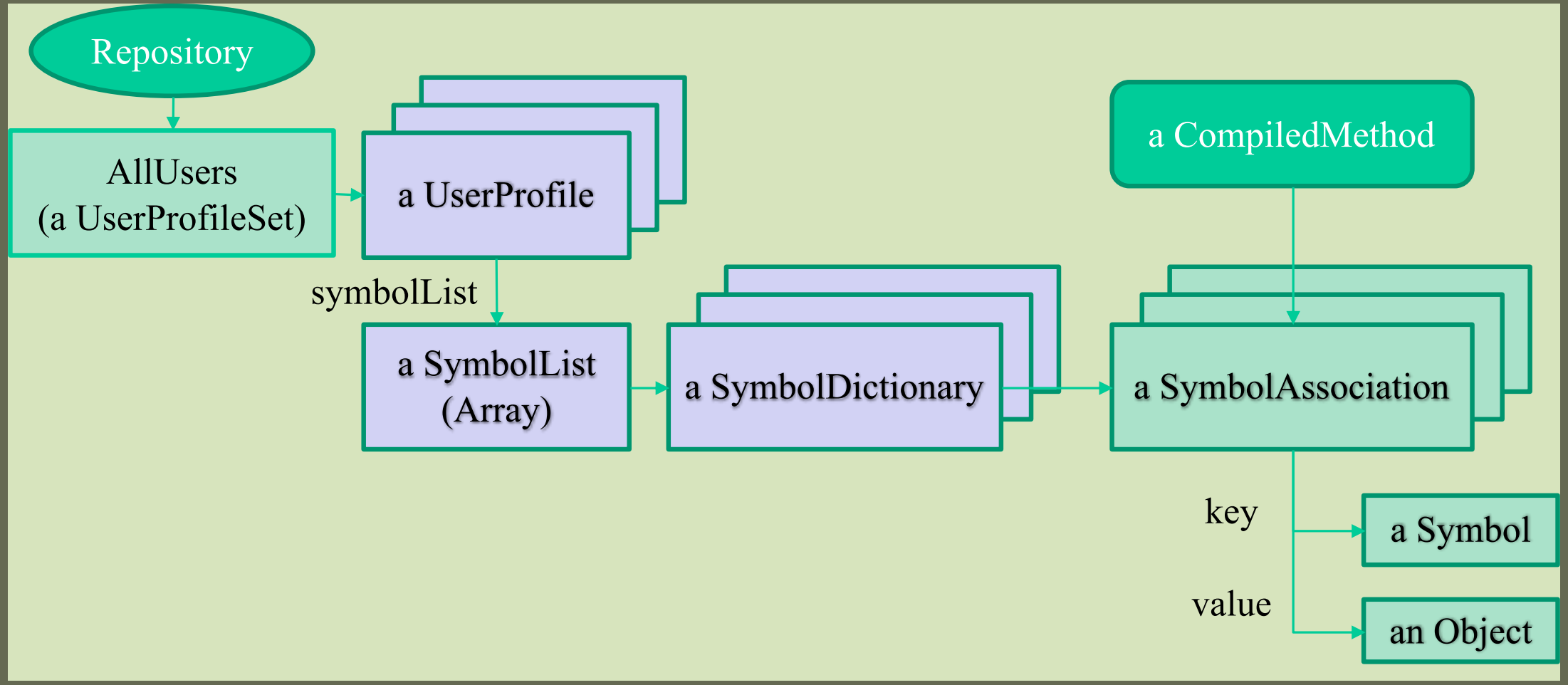
# Agenda

- Motivation
- GemStone Namespaces
  - **Globals**
  - Methods
- Hosting Process
- Demo
- IDE
- Summary and Questions

# Pharo Global Lookup



# GemStone Global Lookup



# Pharo Globals

- Create a new SymbolDictionary (named #'Pharo') to hold all Pharo globals
- Most classes are unique to Pharo
  - Even if they have the same name
- Some classes are shared between GemStone and Pharo
  - Literals
  - Other classes known to the VM

# Literals (and their Superclasses)

- Array: #()
- BlockClosure: []
- Boolean: true, false
- ByteArray: #[1 2 3]
- Character: $a
- Float: 1.23
- SmallInteger: 42
- String: ‘Smalltalk’
- Symbol: #Array
- UndefinedObject: nil

# Other Classes Known to the VM

- Behavior, Class, Metaclass
- Exception, MessageNotUnderstood, ZeroDivide, …
- Pragma
- Process
- ProcessorScheduler
- Semaphore

# Agenda

- Motivation
- GemStone Namespaces
  - Globals
  - **Methods**
- Hosting Process
- Demo
- IDE
- Summary and Questions

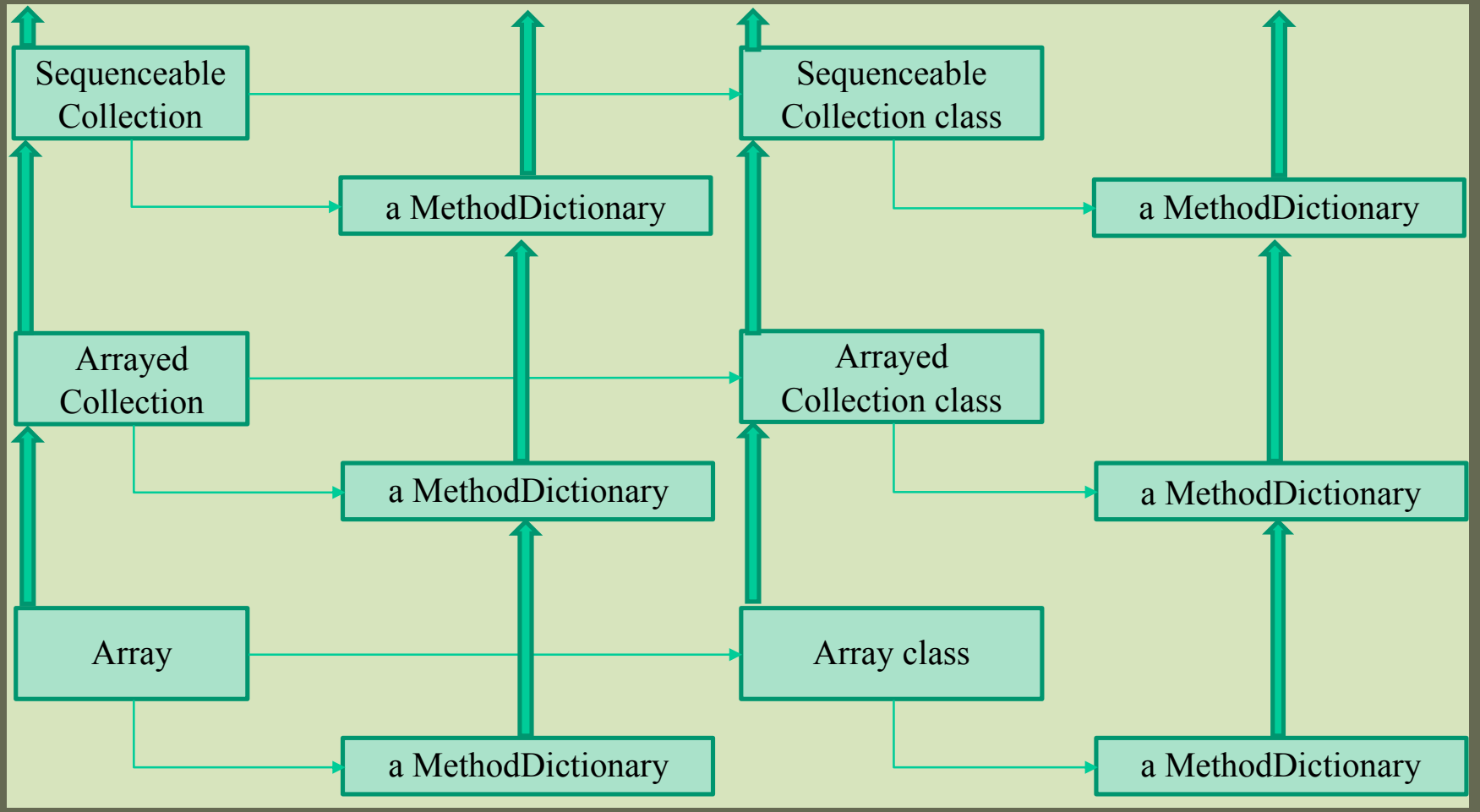
# Problem: Conflicting Implementation

- Array>>printOn:
  - Pharo
    - #(1 2 3) printString  '#(1  2  3)'
  - GemStone
    - #(1 2 3) printString  'anArray( 1, 2, 3)'

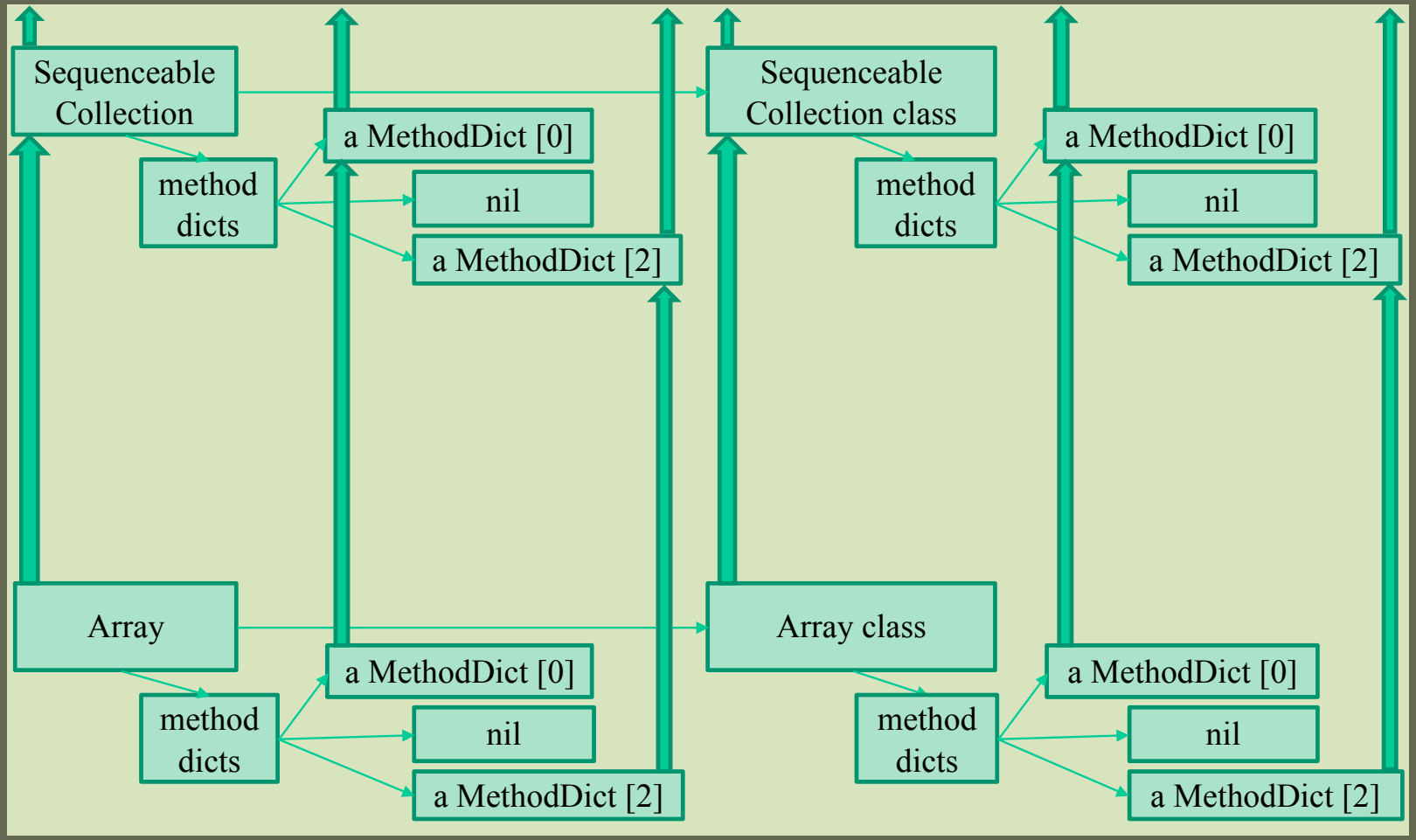
# Method Namespaces

- Each GemStone class has a *collection* of MethodDictionary instances
  - Methods are compiled into an “environment”
  - Message sends to same environment (by default)
- Method environments:
  0 = GemStone/S (default)
  1 = Maglev (reserved for Ruby)
  2+ are for others
  - We use 2 for Pharo

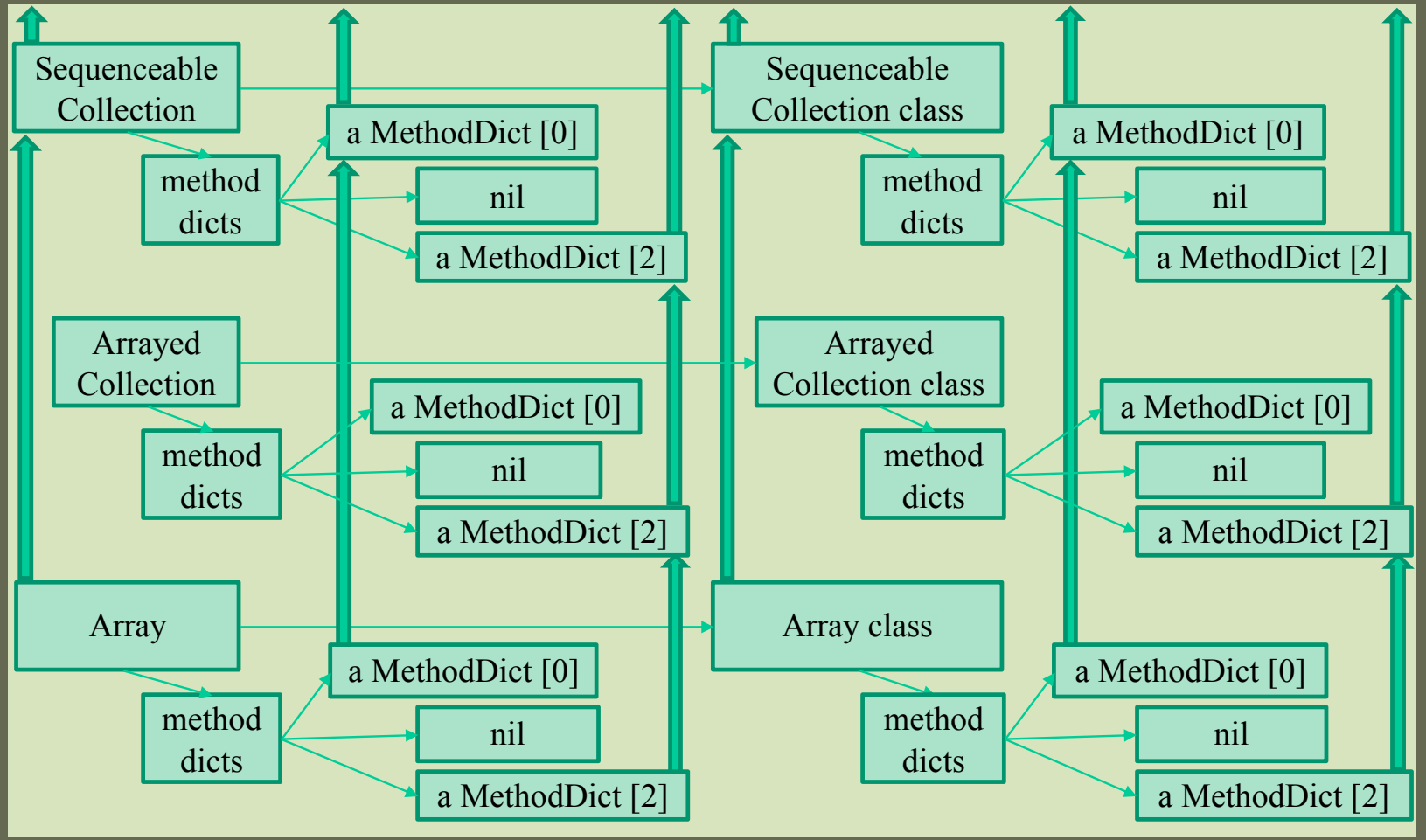
# Pharo Method Dictionaries



# GemStone Method Dictionaries - 1

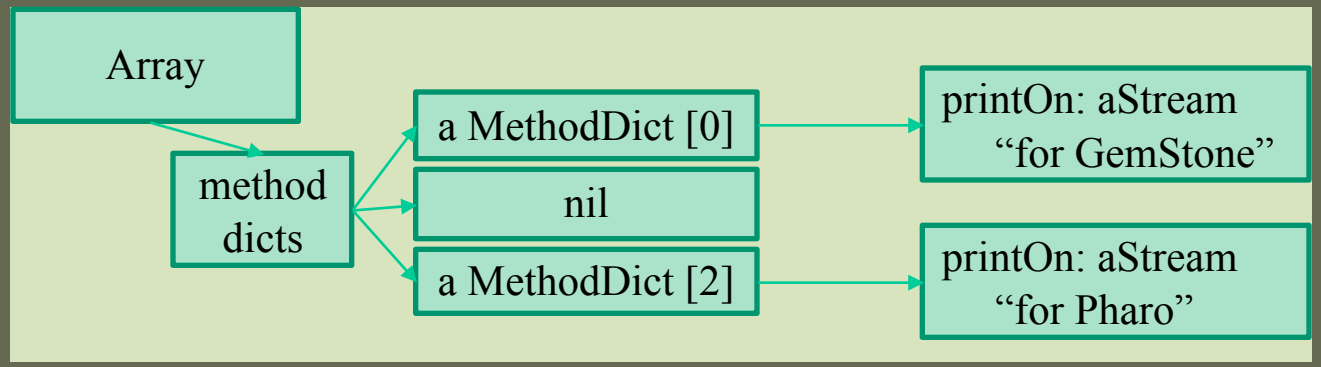


# GemStone Method Dictionaries - 2



# Message Send Examples

- Implicit
  - “in env 0”#(1 2 3) printString  'anArray( 1, 2, 3)'
  - “in env 2”#(1 2 3) printString  '#(1  2  3)'
- Explicit
  - #(1 2 3) @env0:printString  'anArray( 1, 2, 3)'
  - #(1 2 3) @env2:printString  '#(1  2  3)’



# Agenda

- Motivation
- GemStone Namespaces
- **Hosting Process**
- Demo
- IDE
- Summary and Questions

# Overview of Hosting Process

- Prepare Pharo image
- Create GemStone files from Pharo
- Load to GemStone
- Install edits to methods
- Run tests
- "rinse and repeat"

# Prepare Pharo Image

- Pharo "cleanup"
  - 35 GitHub pull requests to Pharo (thanks to the community for help)
  - Use "Minimal Image" (headless) to reduce porting effort
  - Still has over 1 500 classes, 25 000 methods, and 500 primitives
- Load additional packages
  - Minimal image is incomplete (references to undefined classes)
  - Monticello
  - SUnit and several test packages

# Create GemStone Files from Pharo

- PharoGs.st generates PharoGs.tpz and other class-specific files
- 7 globals, 13 pools, 1862 classes, 30384 methods
  - 46 classes are shared
- Automatic translation
  - Comment out code with known issues, replace with error
  - Primitives get a <PharoPrimitive> pragma
  - Known compile errors get a <PharoCompileError> pragma
    - Reference to missing instance variable

# Use 30 GemStone Class with Same Name

- Array
- Behavior
- Boolean
- ByteArray
- CannotReturn
- Character
- Class
- Collection
- Delay
- Error
- Exception
- ExceptionSet
- Fraction
- Integer
- LargeInteger
- Magnitude
- Message
- MessageNotUnderstood
- Notification
- Number
- Object
- Pragma
- ProcessorScheduler
- Semaphore
- SequenceableCollection
- SmallInteger
- Symbol
- UndefinedObject
- Warning
- ZeroDivide

# Use 16 GemStone Classes with Different Name

- ArithmeticError -> NumericError
- BlockClosure -> ExecBlock
- BoxedFloat64 -> Float
- ByteString -> String
- ByteSymbol -> Symbol
- ClassDescription -> Module
- CompiledMethod -> GsNMethod
- Float -> BinaryFloat
- Metaclass -> Metaclass3
- MethodDictionary -> GsMethodDictionary
- Process -> GsProcess
- ScaledDecimal -> FixedPoint
- SmallFloat64 -> SmallDouble
- String -> CharacterCollection
- WideString -> QuadByteString
- WideSymbol -> QuadByteSymbol

# Load to GemStone

- Create #'Pharo' SymbolDictionary
  - Also create a #'Pools' SymbolDictionary
- Add global for #'thisContext'
  - Also some missing globals for tests, Iceberg, and Seaside
- Load classes and methods
- Set Pharo-specific method-lookup superclass
  - Object "subclasses" from ProtoObject
  - Array, ByteArray, and String "subclass" from ArrayedCollection

# Image Initialization

- Initialize selected classes
  - Not ready to initialize all of them!
- Smalltalk := SmalltalkImage new
- SharedPool allSubclasses do: [:each | each initialize]
- SystemOrganization := SystemOrganizer new

# Install Edits to Methods

- 122 (of 1862) classes have at least one method modified
- Method edits
  - PharoGs 809
  - PharoGsError 320
  - PharoCompileError 225
  - PharoPrimitive 70

# Run Tests

- Identify test failures
- Update replacement methods
- Reload
- "rinse and repeat"

```
FloatTest                  41 ran, 41 passed, 1 skipped
FractionTest               16 ran, 16 passed
IntegerDigitLogicTest       7 ran,  7 passed
IntegerTest                49 ran, 49 passed, 3 skipped
NumberTest                 14 ran, 14 passed
ScaledDecimalTest          27 ran, 27 passed
SmallIntegerTest           12 ran, 12 passed
BooleanTest                 3 ran,  3 passed
FalseTest                  15 ran, 15 passed
TrueTest                   15 ran, 15 passed
UndefinedObjectTest        17 ran, 17 passed
FileTest                    9 ran,  9 passed
Base64MimeConverterTest     3 ran,  3 passed
Base64Test                  4 ran,  4 passed
NetNameResolverTest         1 ran,  1 passed, 1 skipped
SocketAddressTest           5 ran,  5 passed
TCPSocketTest               8 ran,  8 passed
SocketStreamTest           19 ran, 19 passed
TCPSocketEchoTest           1 ran,  1 passed
FileStreamTest             12 ran, 12 passed
MCAncestryTest              4 ran,  4 passed
ZipArchiveTest              9 ran,  9 passed
ZipCrcTest                  9 ran,  9 passed
ZipExtensionTest            1 ran,  1 passed
ZipWriteStreamTest          2 ran,  2 passed
```

# Load Seaside

```
Metacello new
    baseline:'Seaside3';
    repository: 'github://SeasideSt/Seaside:master/repository';
    onWarningLog;
    load.
```

- Replace one method: GRPharoPlatform>>seasideSuspendFlowDo:
  - Create a partial continuation
  - Note that everything else in GRPharoPlatform works!

# Agenda

- Motivation
- GemStone Namespaces
- Hosting Process
- **Demo**
- IDE
- Summary and Questions

# Demo

- (video)

# Agenda

- Motivation
- GemStone Namespaces
- Hosting Process
- Demo
- **IDE**
- Summary and Questions

# Integrated Development Environment (IDE)

- JADE
  - Code Browser
  - Workspace
  - Inspector
  - Debugger

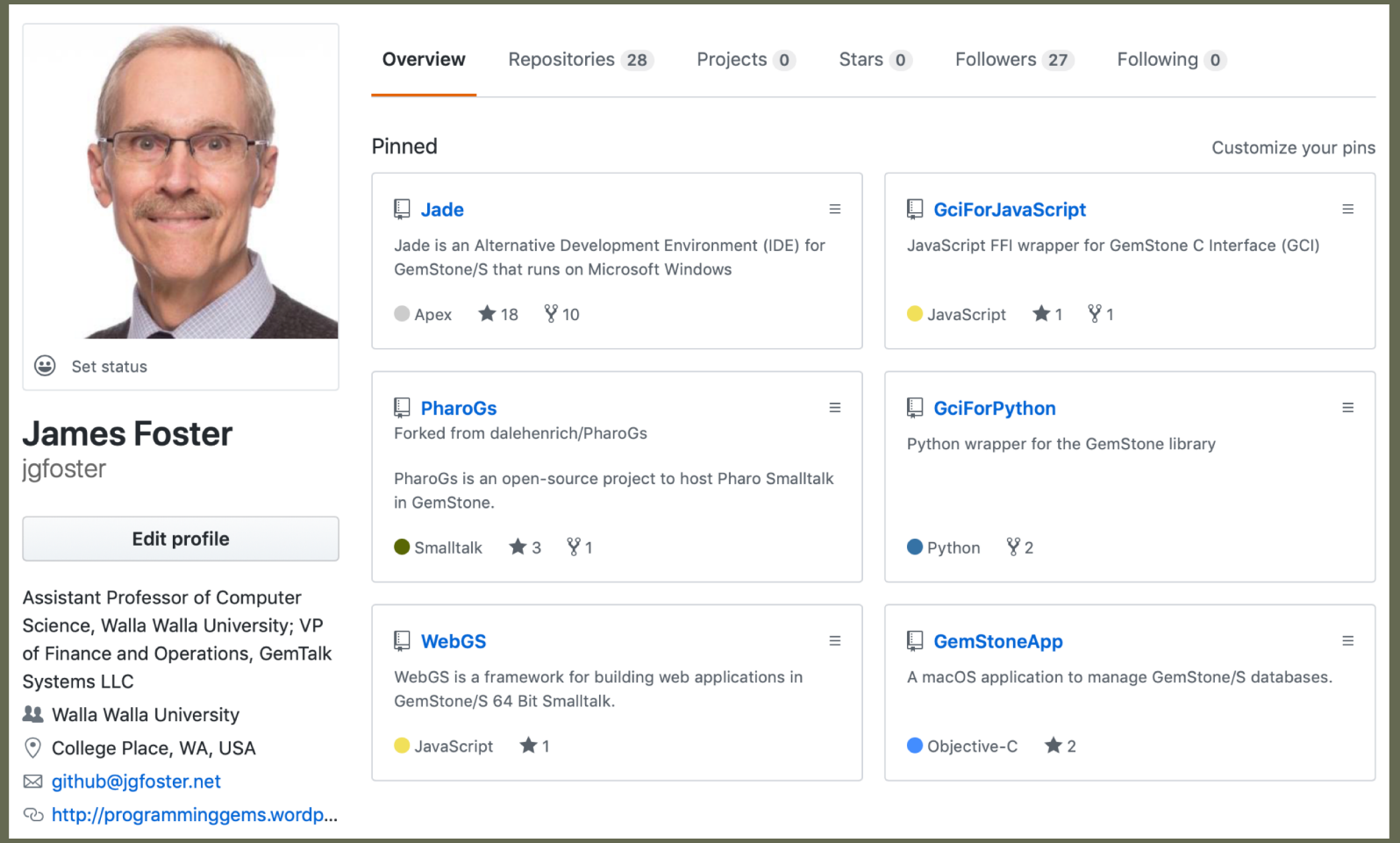
# Agenda

- Motivation
- GemStone Namespaces
- Hosting Process
- Demo
- IDE
- **Summary and Questions**

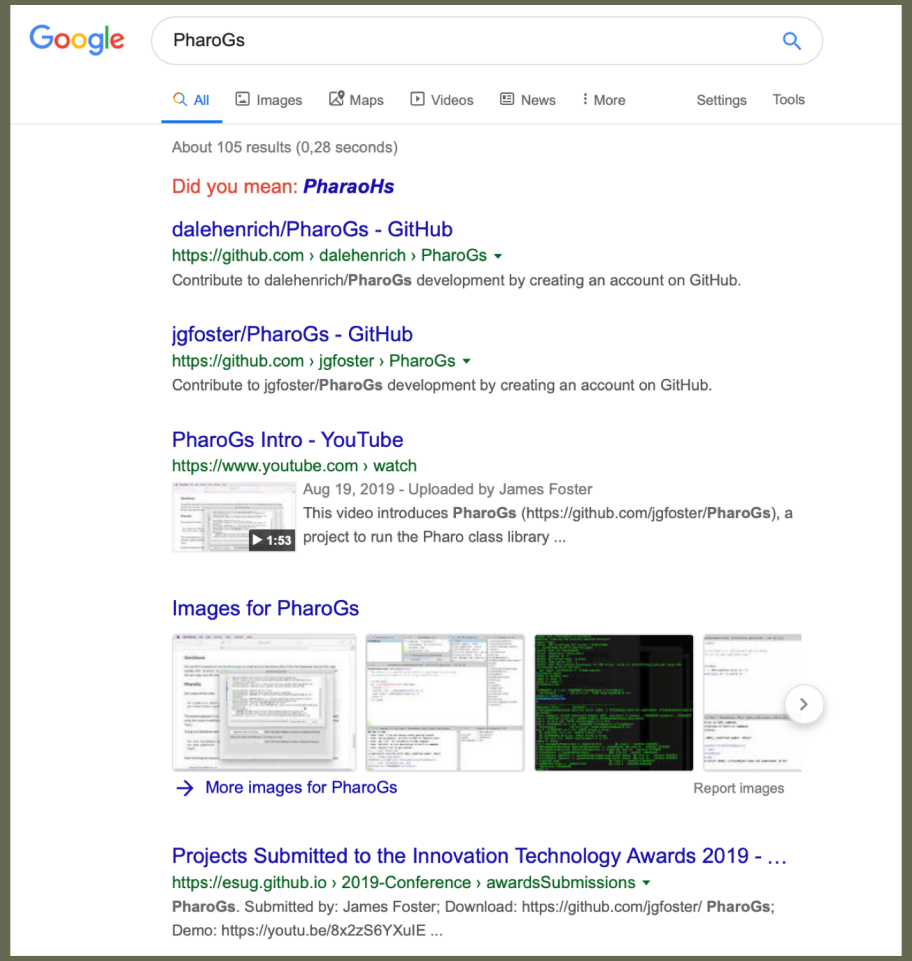
# Summary

- Proof of concept
  - Can it be done?
- Challenges?
  - Compiler, UFFI, thisContext, slots, traits, …
- Personal motivation
  - Academic research (publish, degree)
  - Not a GemTalk sponsored project
    - But GemTalk has shown a lot of support for Pharo; talk to us about interest!

# Code: https://github.com/jgfoster/PharoGs



# Google "PharoGs"



# Questions?

- James.Foster@GemTalkSystems.com
  - VP for Finance and Operations
- James.Foster@WallaWalla.edu
  - Assistant Professor of Computer Science