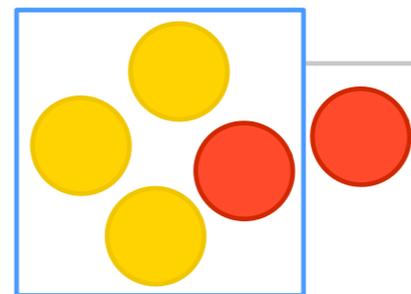


Building the VM

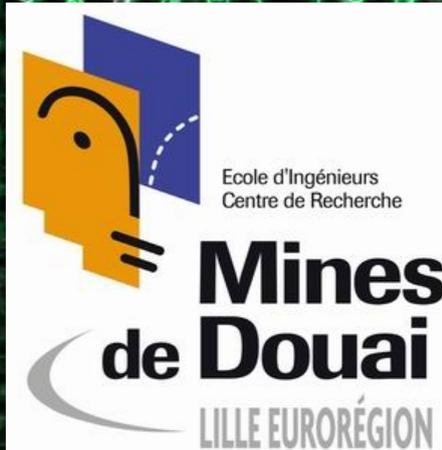
Mariano Martinez Peck
marianopeck@gmail.com

<http://marianopeck.wordpress.com/>

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



RMod

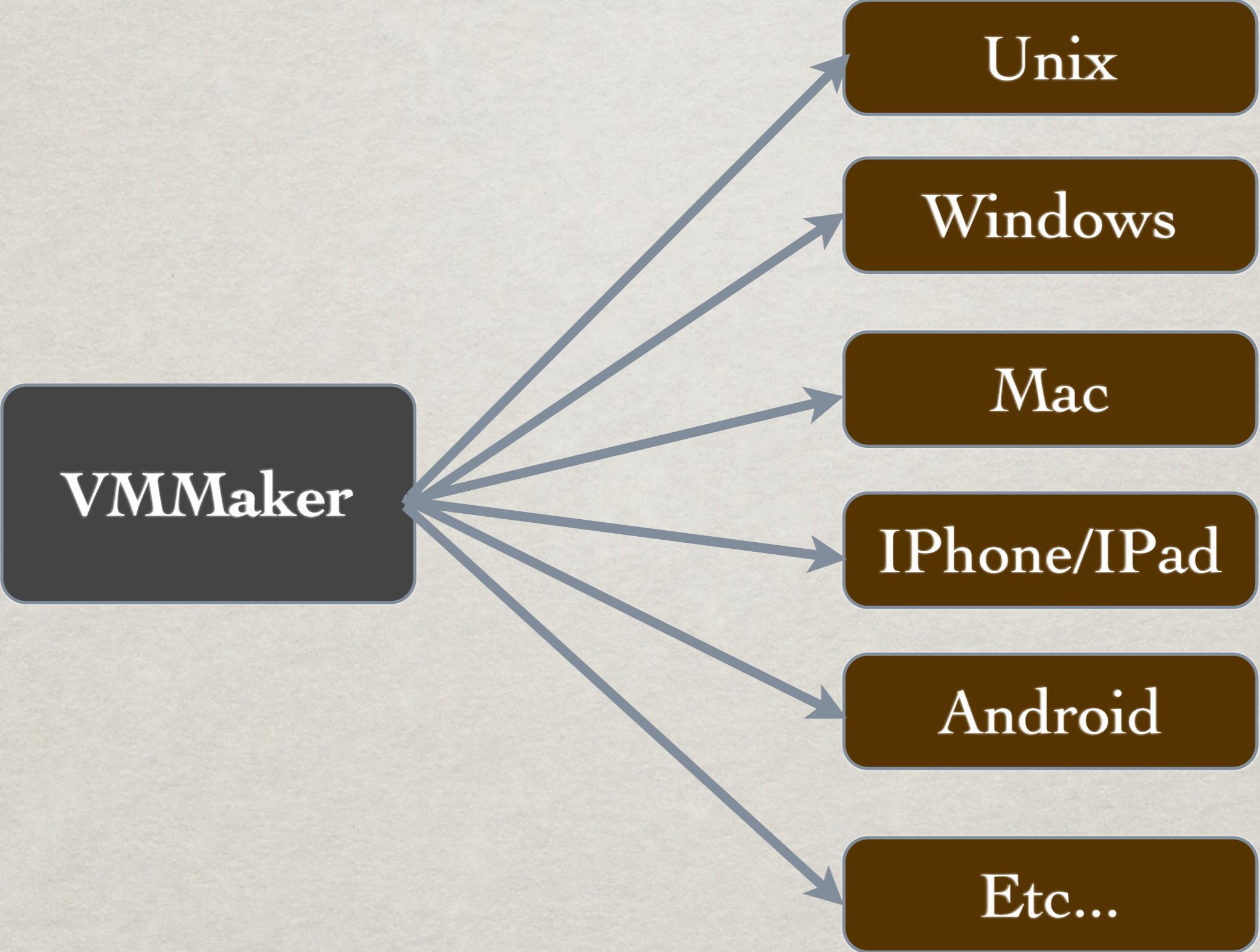


VMMaker

Platform code

All VM kernel: GC, method lookup, bytecode interpretation, memory management, etc.

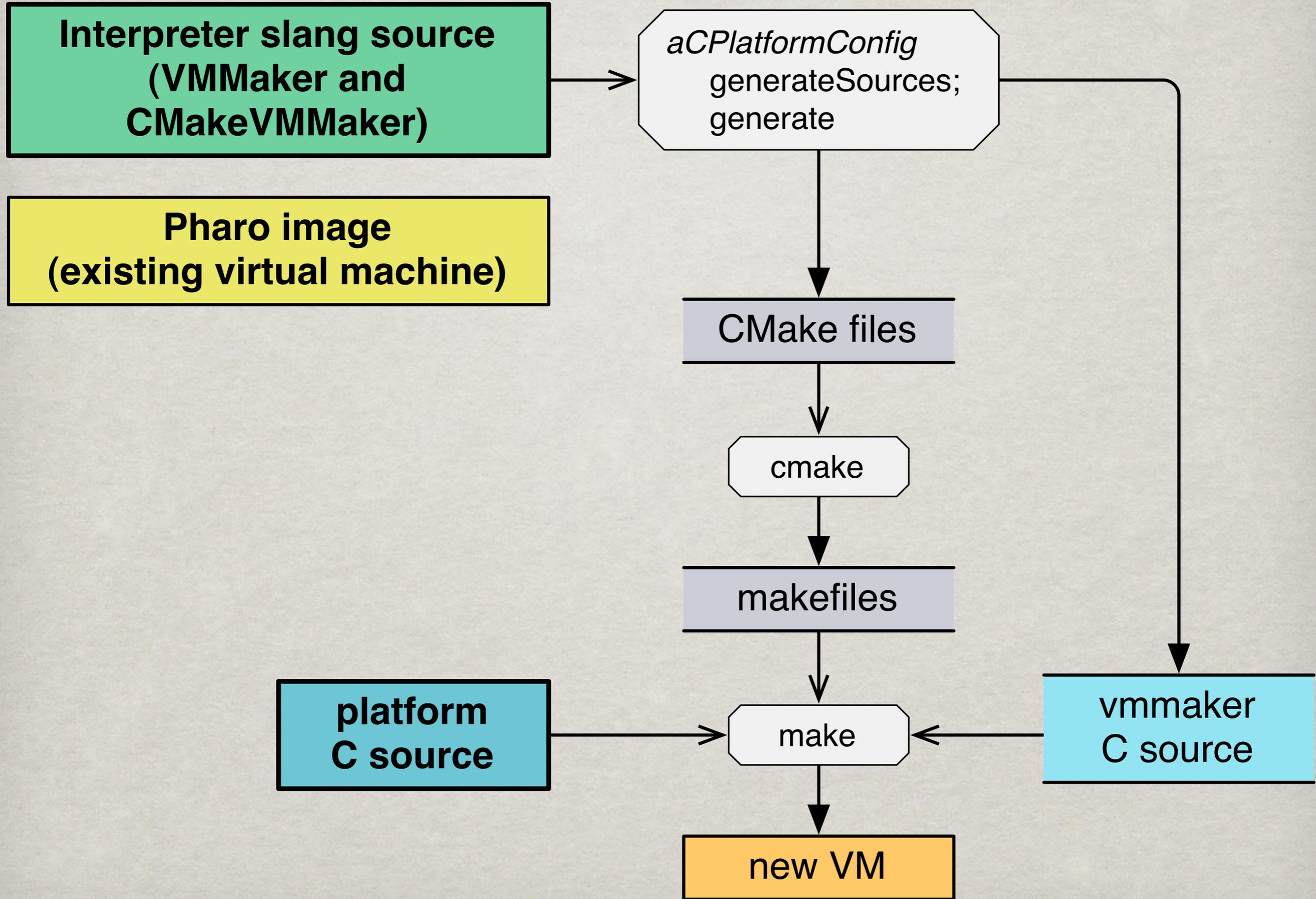
Platform-dependent code like sockets, files, etc.



VMMaker	Platform code
All VM kernel: GC, method lookup, bytecode interpretation, memory management, etc.	Platform-dependent code like sockets, files, etc.
Hand-written code in Slang (subset of Smallalk) with classes and methods.	Hand-written code in C (and some Objective-C).
Versioned with Monticello and Metacello.	Versioned with Git.
Needs to be translated to C.	It is already in C.

THE BUILDING PROCESS AND TOOLS

1. Get platform code (fork from git).
2. Get VMMaker and CMakeVMMaker (load a metacello conf).
3. Generate VMMaker C source code and CMake files.
4. Run CMake.
5. Run make.



BUILD ENVIRONMENT

☼ Git client

☼ CMake

☼ make

☼ gcc

Mac

XCode (problem in latest version)

Unix

sudo apt-get install build-essential git-core cmake

Windows

MSYS and MinGW

<http://marianopeck.wordpress.com/2011/04/10/building-the-vm-from-scratch-using-git-and-cmakevmmaker/>

ducasse

stef-squeakvm



squeak-vm → stef-squeakvm

 Clone & push urls GIT HTTP SSH `git@gitorious.org:~ducasse/squeak-vm/stef-squeakvm.git` ?

Adding this repository as a pushable origin:

```
git remote add origin git@gitorious.org:~ducasse/squeak-vm/stef-squeakvm.git
# to push the master branch to the origin remote we added above:
git push origin master
# after that you can just do:
git push
```

Cloning this repository:

```
git clone git://gitorious.org/~ducasse/squeak-vm/stef-squeakvm.git stef-squeakvm
cd stef-squeakvm
```

Add this repository as a remote to an existing local repository:

```
git remote add stef-squeakvm git://gitorious.org/~ducasse/squeak-vm/stef-squeakvm.git
git fetch stef-squeakvm
git checkout -b my-local-tracking-branch stef-squeakvm/master_or_other_branch
```

Branches: **master**Clone of: **blessed**

Commit log

Source tree

Merge requests (0)

Clone repository

Unwatch

Project: **squeak-vm**Owner: **~ducasse**Clone of: **squeak-vm/blessed.git**Created: **02 Feb 15:45**

Clone repository

Request merge

Manage collaborators

Edit repository

Committers

ducasse (creator)

Repository clones

No clones on Gitorious yet of this repository

Activities 

Wednesday February 02 2011

REPOSITORY

15:45

ducasse cloned **squeak-vm/blessed**New repository is in `stef-squeakvm`

WHY GIT?

Git is a distributed source management tool while SVN is centralized one.

This means that git works better for the developing and managing of open-source projects which are driven by community rather than by a single or selected team of developers.

With gitorious.org, one can easily create its own branch publicly available without needing to ask for VIP's permission.

1) GETTING PLATFORM CODE

```
git clone git://gitorious.org/cogvm/blessed.git
```

or...

```
git clone git://gitorious.org/~marianopeck/cogvm/marianopecks-blessed.git
```

SOURCES DIRECTORY STRUCTURE

/ <root dir>

platforms/ common code for all platforms

Cross/

Mac OS/

RiscOS/

unix/

win32/

platform-specific source code
& configuration files/tools

src/ VMMaker generated code

build/ directory used for cmake configuration files

results/ where built VMs are stored

image/ possible place to put the generator Pharo image

codegen-scripts/ useful and needed scripts

2) GETTING VMMAKER

1. Take a Pharo image.
2. Place it into /image (or any) subdirectory.
3. Evaluate the code from /codegen-scripts/
LoadVMMaker.st. Example:

Gofer new

```
squeaksource: 'PharoVM';  
package: 'ConfigurationOfPharoVM';  
load.
```

((Smalltalk at: #ConfigurationOfPharoVM) project version: '2.0-1') load.

-> This will also install CMakeVMMaker!!! <-

3) GENERATING SOURCES

<MyConfiguration> generateSources

(translates VMMaker Slang to C)

<MyConfiguration> generate

(generates CMake files)

Depending on what you
want to build:

```
CPlatformConfig
CUnixConfig
  CogUnixConfig
  CogDebugUnixConfig
  CogUnixNoGLConfig
  StackInterpreterUnixConfig
  StackInterpreterDebugUnixConfig
  FixedVerSIDebugUnixConfig
MacOSConfig
  CocoaIOSConfig
  CocoaIOSCogConfig
  CocoaIOSCogJitConfig
  CocoaIOSCogStackConfig
CogMacOSConfig
  StackInterpreterMacOSConfig
```

4) COMPILING

cd build

../codegen-scripts/extract-commit-info.sh

cmake . (on Windows: cmake -G"MSYS Makefiles" .)

make

.. run ..

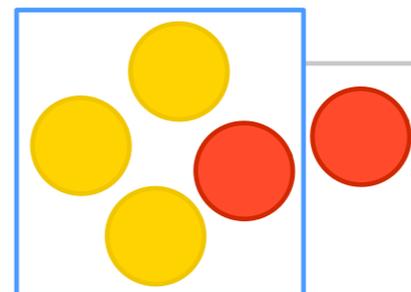
```
/Users/mariano/PhD/Presentaciones/Fuel/PharoConf2012/buildingVM/blessed/src/plugins/TestOSAPugin/TestOSAPugin.c
Linking C shared library /Users/mariano/PhD/Presentaciones/Fuel/PharoConf2012/buildingVM/blessed/results/CogVM.app
[100%] Built target TestOSAPugin
Scanning dependencies of target SqueakSSL
[100%] Building C object SqueakSSL/CMakeFiles/SqueakSSL.dir/Users/mariano/PhD/Presentaciones/Fuel/PharoConf2012/b
[100%] Building C object SqueakSSL/CMakeFiles/SqueakSSL.dir/Users/mariano/PhD/Presentaciones/Fuel/PharoConf2012/b
Linking C shared library /Users/mariano/PhD/Presentaciones/Fuel/PharoConf2012/buildingVM/blessed/results/CogVM.app
[100%] Built target SqueakSSL
mariano @ Aragorn : ~/PhD/Presentaciones/Fuel/PharoConf2012/buildingVM/blessed/build
;-) cd ../results/CogVM.app/Contents/MacOS/
mariano @ Aragorn : ~/PhD/Presentaciones/Fuel/PharoConf2012/buildingVM/blessed/results/CogVM.app/Contents/MacOS
;-) ./CogVM
mariano @ Aragorn : ~/PhD/Presentaciones/Fuel/PharoConf2012/buildingVM/blessed/results/CogVM.app/Contents/MacOS
```

Thanks!

Mariano Martinez Peck
marianopeck@gmail.com

<http://marianopeck.wordpress.com/>

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



RMod

