Cross-platform app with Pharo

The Pharo Launcher case

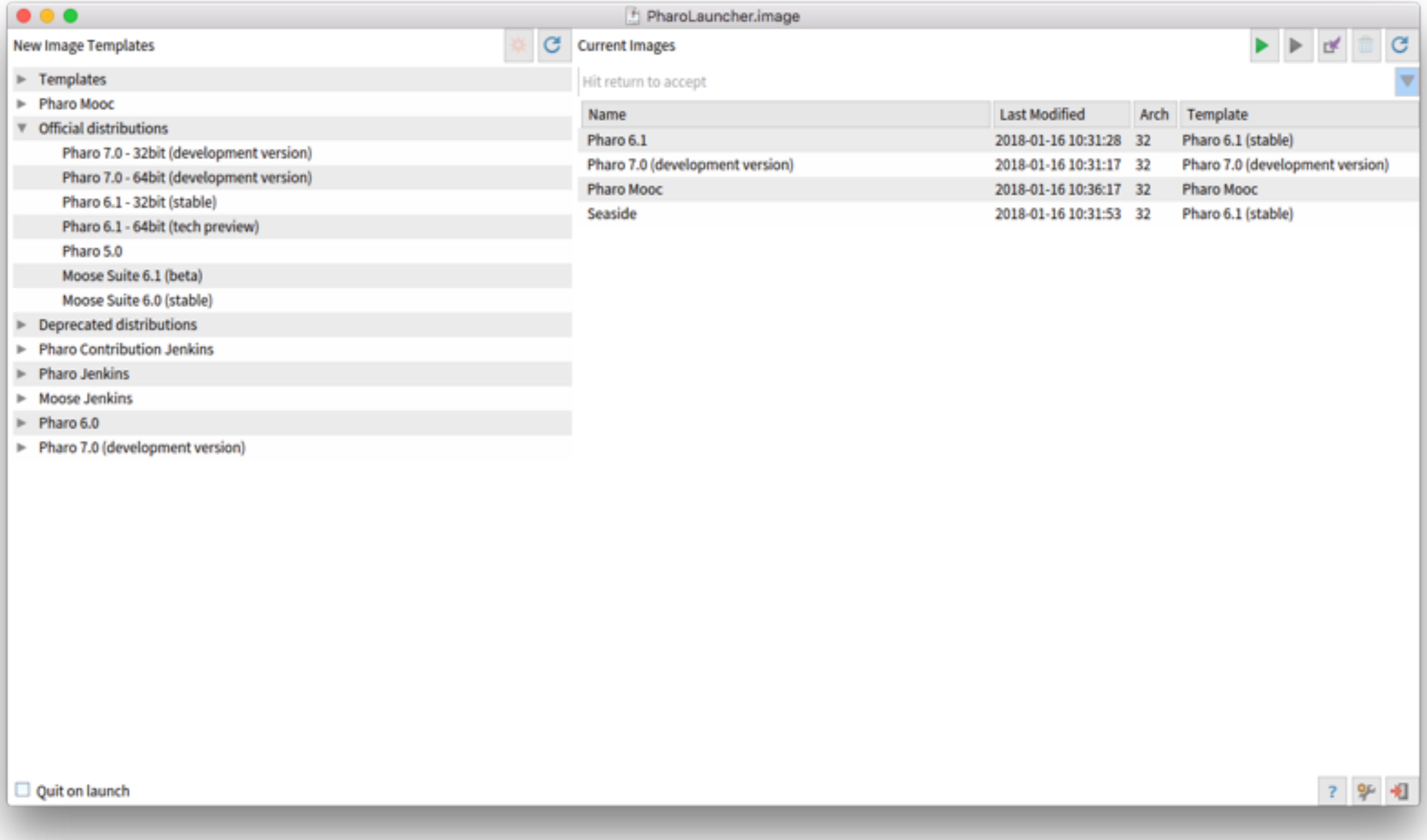
Christophe Demarey, ESUG 2018, Cagliari

Pharo Launcher

Photo: https://www.maxpixel.net/static/photo/1x/Smoke-Rocket-Nasa-Rocket-Launch-Take-Off-Side-View-67723.jpg

Pharo Launcher

- A small app to easily get, launch and manage Pharo images



PharoLauncher.image

New Image Templates

- Templates
- Pharo Mooc
- Official distributions
  - Pharo 7.0 - 32bit (development version)
  - Pharo 7.0 - 64bit (development version)
  - Pharo 6.1 - 32bit (stable)
  - Pharo 6.1 - 64bit (tech preview)
  - Pharo 5.0
  - Moose Suite 6.1 (beta)
  - Moose Suite 6.0 (stable)
- Deprecated distributions
- Pharo Contribution Jenkins
- Pharo Jenkins
- Moose Jenkins
- Pharo 6.0
- Pharo 7.0 (development version)

Current Images

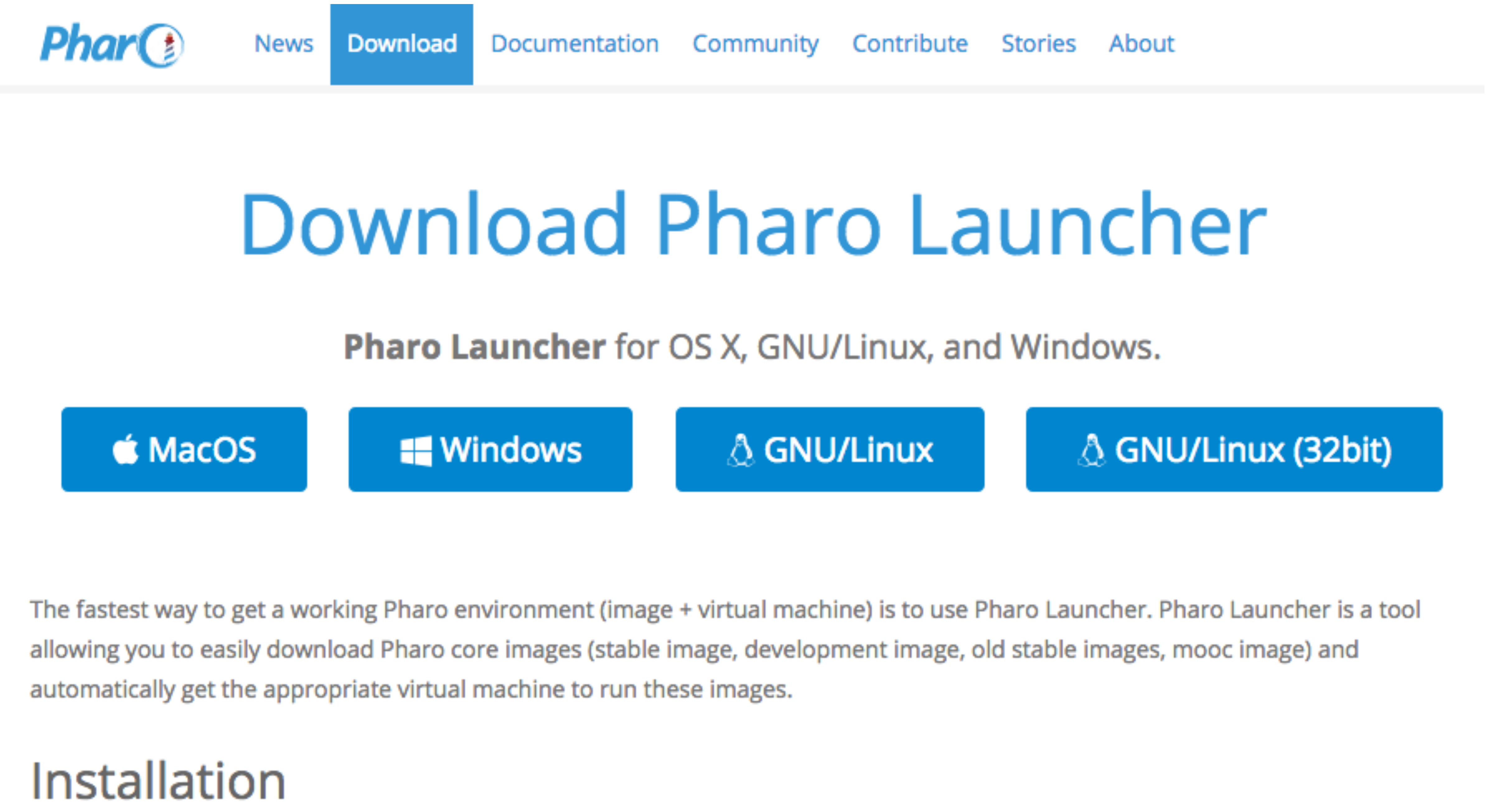
Hit return to accept

Name	Last Modified	Arch	Template
Pharo 6.1	2018-01-16 10:31:28	32	Pharo 6.1 (stable)
Pharo 7.0 (development version)	2018-01-16 10:31:17	32	Pharo 7.0 (development version)
Pharo Mooc	2018-01-16 10:36:17	32	Pharo Mooc
Seaside	2018-01-16 10:31:53	32	Pharo 6.1 (stable)

□ Quit on launch

Pharo Launcher

- Is now the default download for Pharo



Download Pharo Launcher

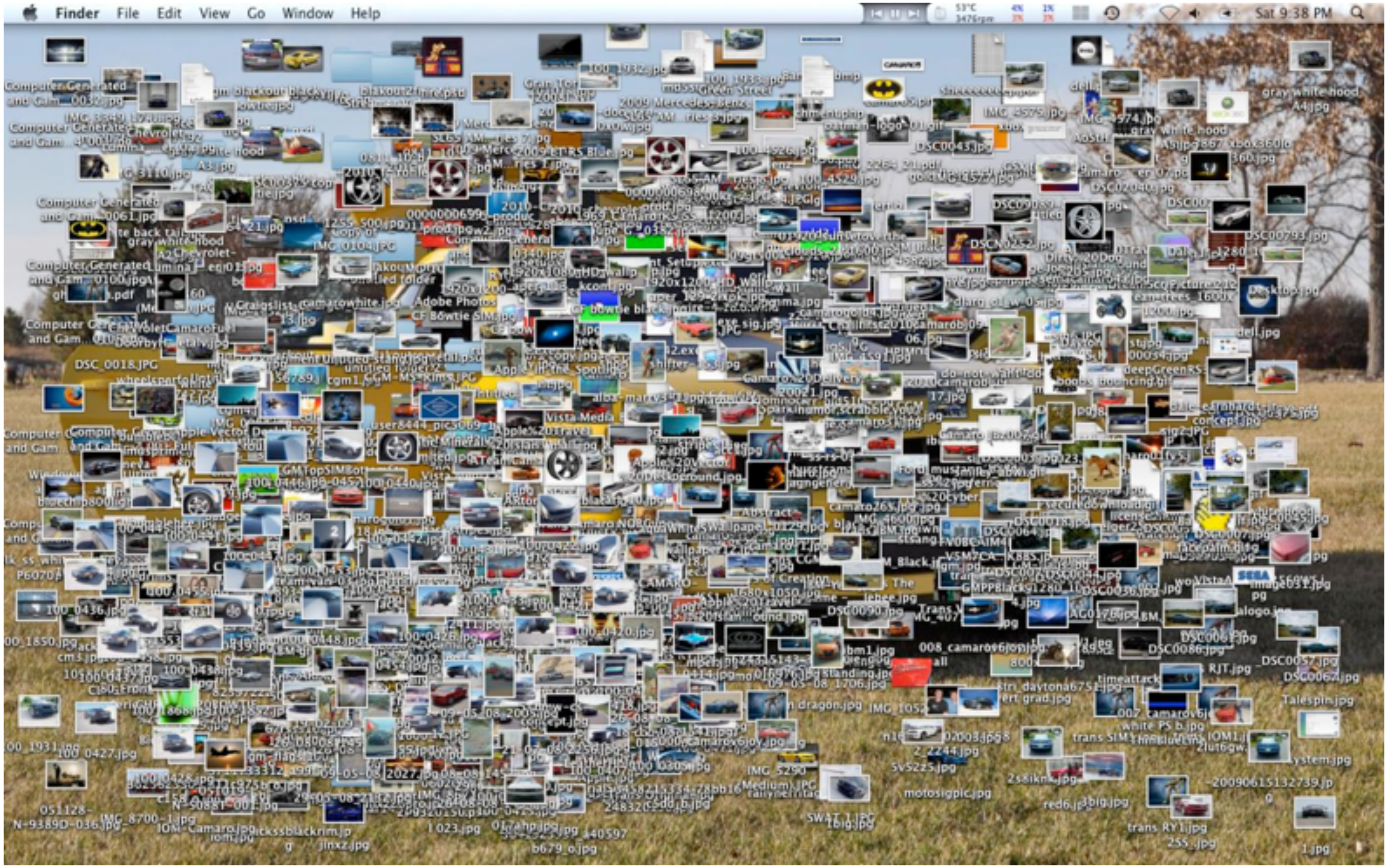
Pharo Launcher for OS X, GNU/Linux, and Windows.

The fastest way to get a working Pharo environment (image + virtual machine) is to use Pharo Launcher. Pharo Launcher is a tool allowing you to easily download Pharo core images (stable image, development image, old stable images, mooc image) and automatically get the appropriate virtual machine to run these images.

Installation

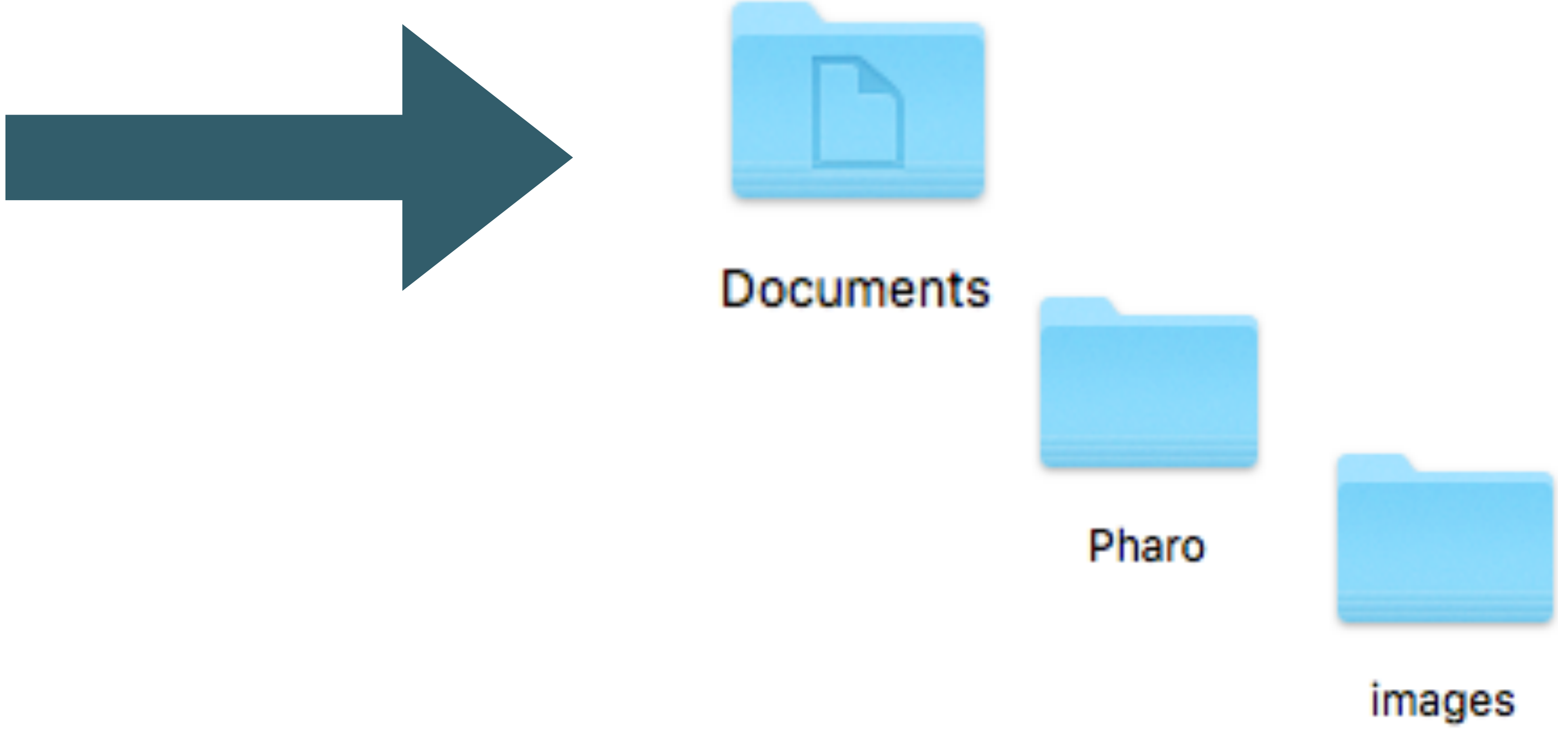
Why Pharo Launcher ?

- Pharo images spread over your file system



Why Pharo Launcher ?

- Pharo images spread over your file system



Why Pharo Launcher ?

- Which VM to use to open my image?

6.0
1.2
Linux
4.0
Windows
2.0
6.1
Mac
pre-Spur
1.3
32-bit
64-bit
1.4
5.0
Spur
3.0

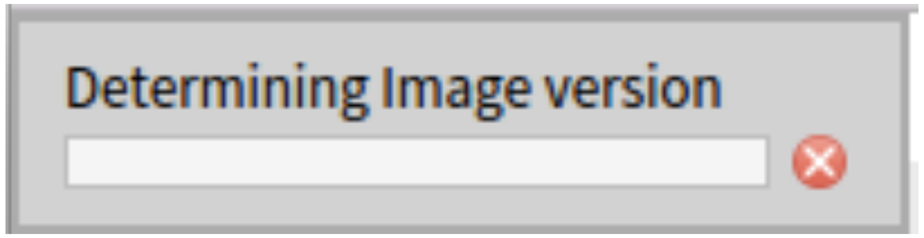
Photo: dreamstime.com

Why Pharo Launcher ?

- Which VM to use to open my image?

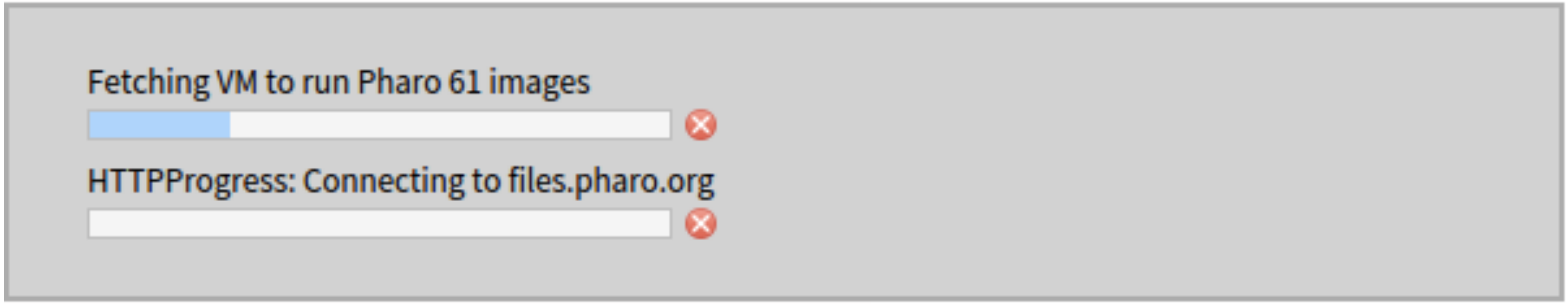
ABLE TO OPEN IMAGES > PHARO 1.0 (2010)

- Let Pharo Launcher determine the VM to use



Determining Image version

- Pharo Launcher will download it if not available locally



Fetching VM to run Pharo 61 images

HTTPProgress: Connecting to files.pharo.org

Photo: https://www.maxpixel.net/Excitement-Woman-Happy-Happiness-Excited-Fun-2649320

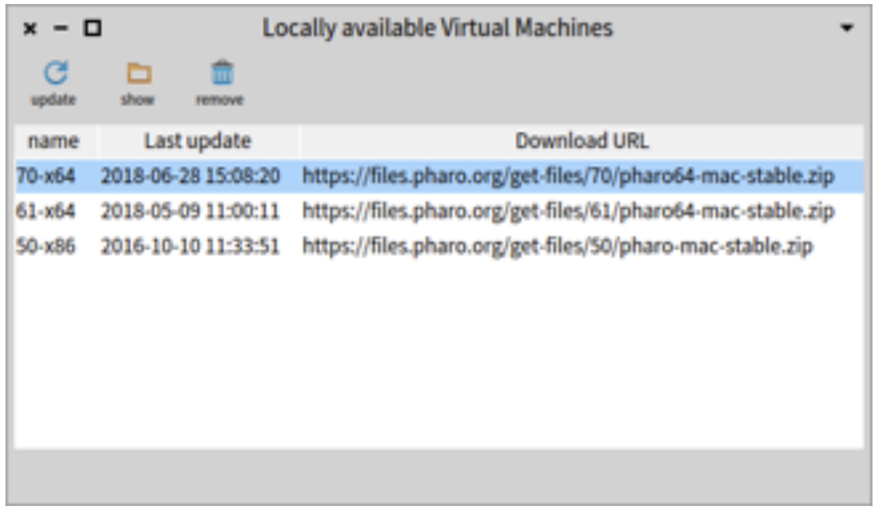
Pharo Launcher

Want to know more?

- https://github.com/pharo-project/pharo-launcher
- http://files.pharo.org/pharo-launcher/

Future

- revisit UI to make it more « newbie » friendly



name	Last update	Download URL
70-x64	2018-06-28 15:08:20	https://files.pharo.org/get-files/70/pharo64-mac-stable.zip
61-x64	2018-05-09 11:00:11	https://files.pharo.org/get-files/61/pharo64-mac-stable.zip
50-x86	2016-10-10 11:33:51	https://files.pharo.org/get-files/50/pharo-mac-stable.zip

Cross-platform app

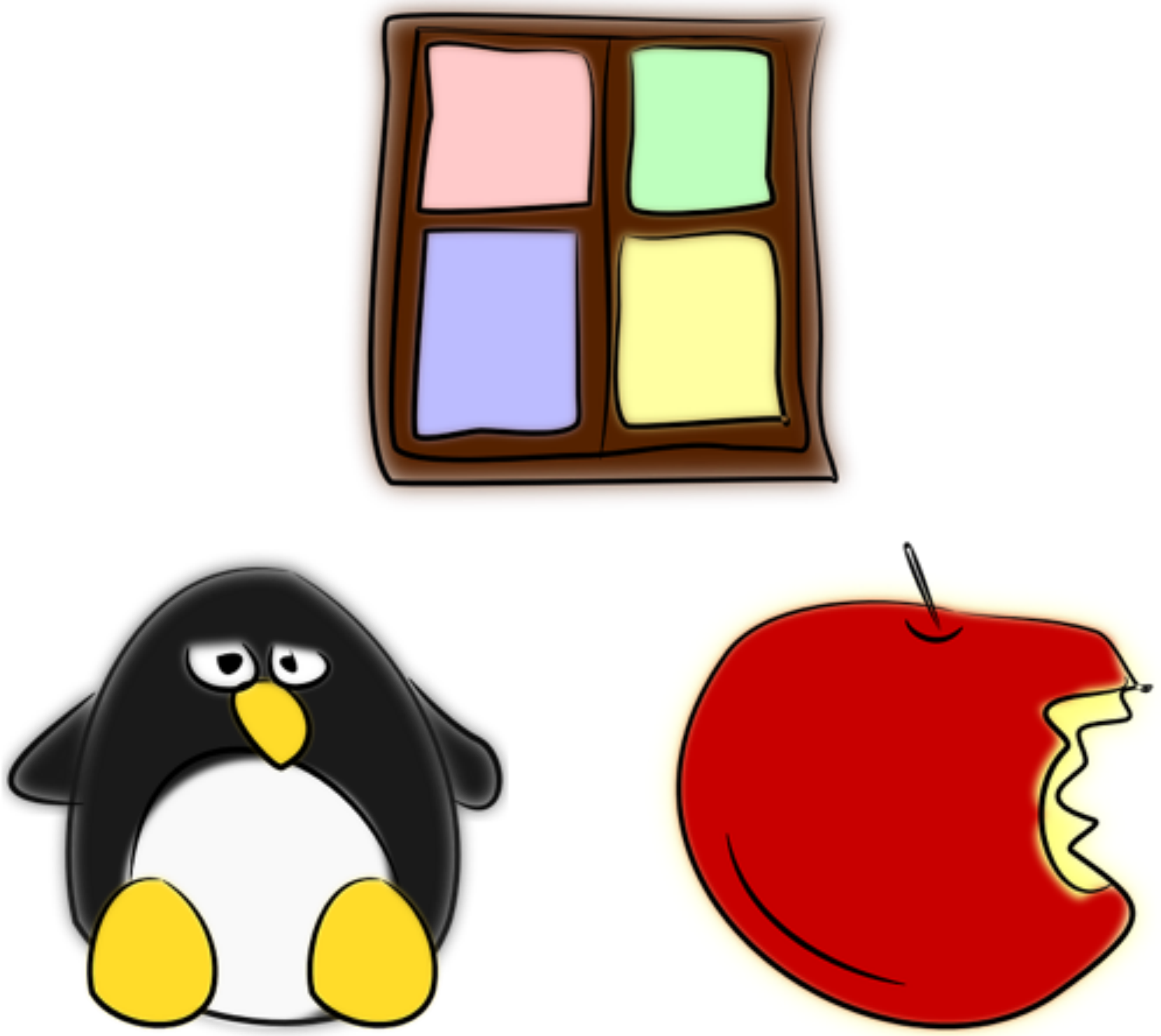
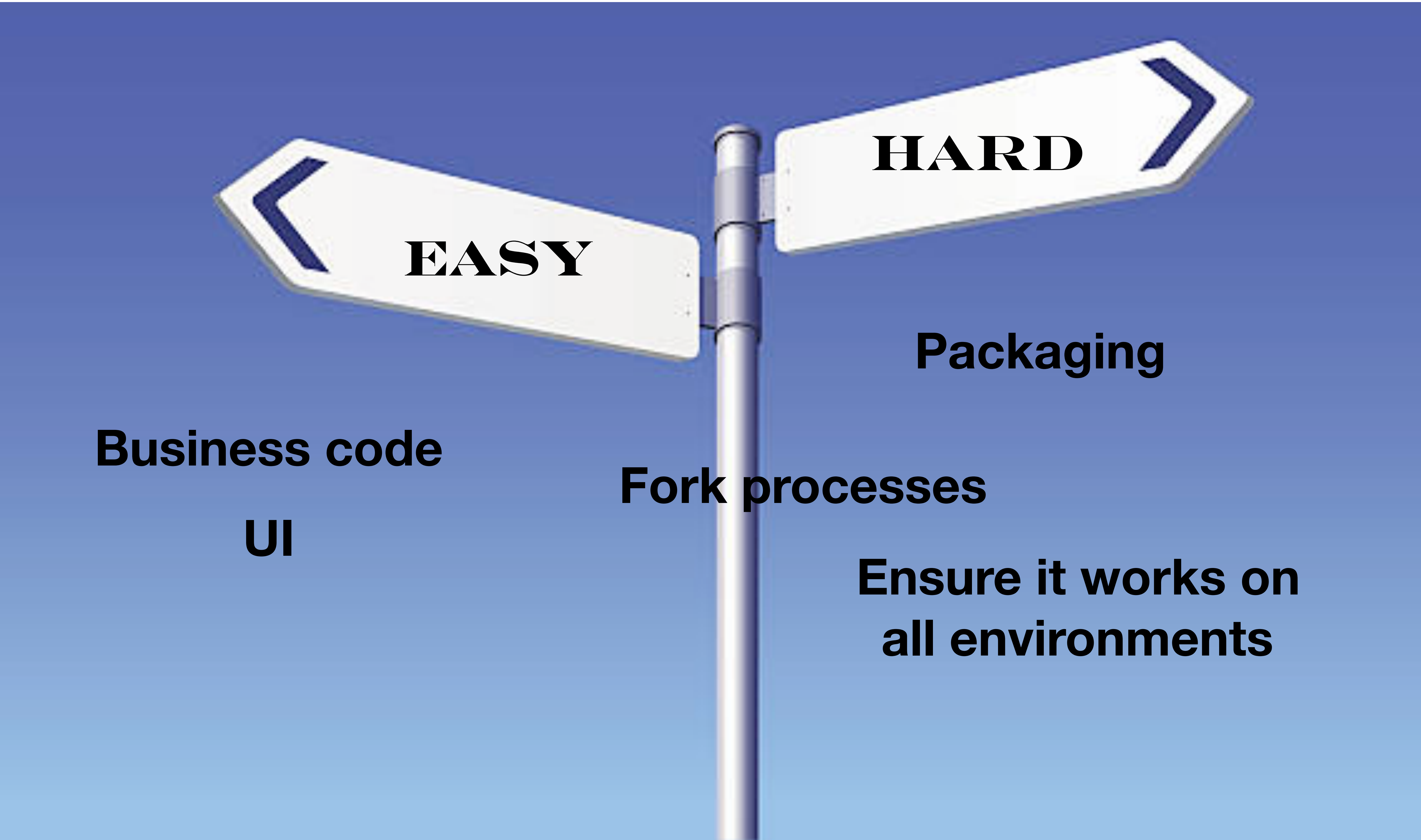


Photo: https://www.maxpixel.net/static/photo/1x/Smoke-Rocket-Nasa-Rocket-Launch-Take-Off-Side-View-67723.jpg

Pharo Launcher



Business code

UI

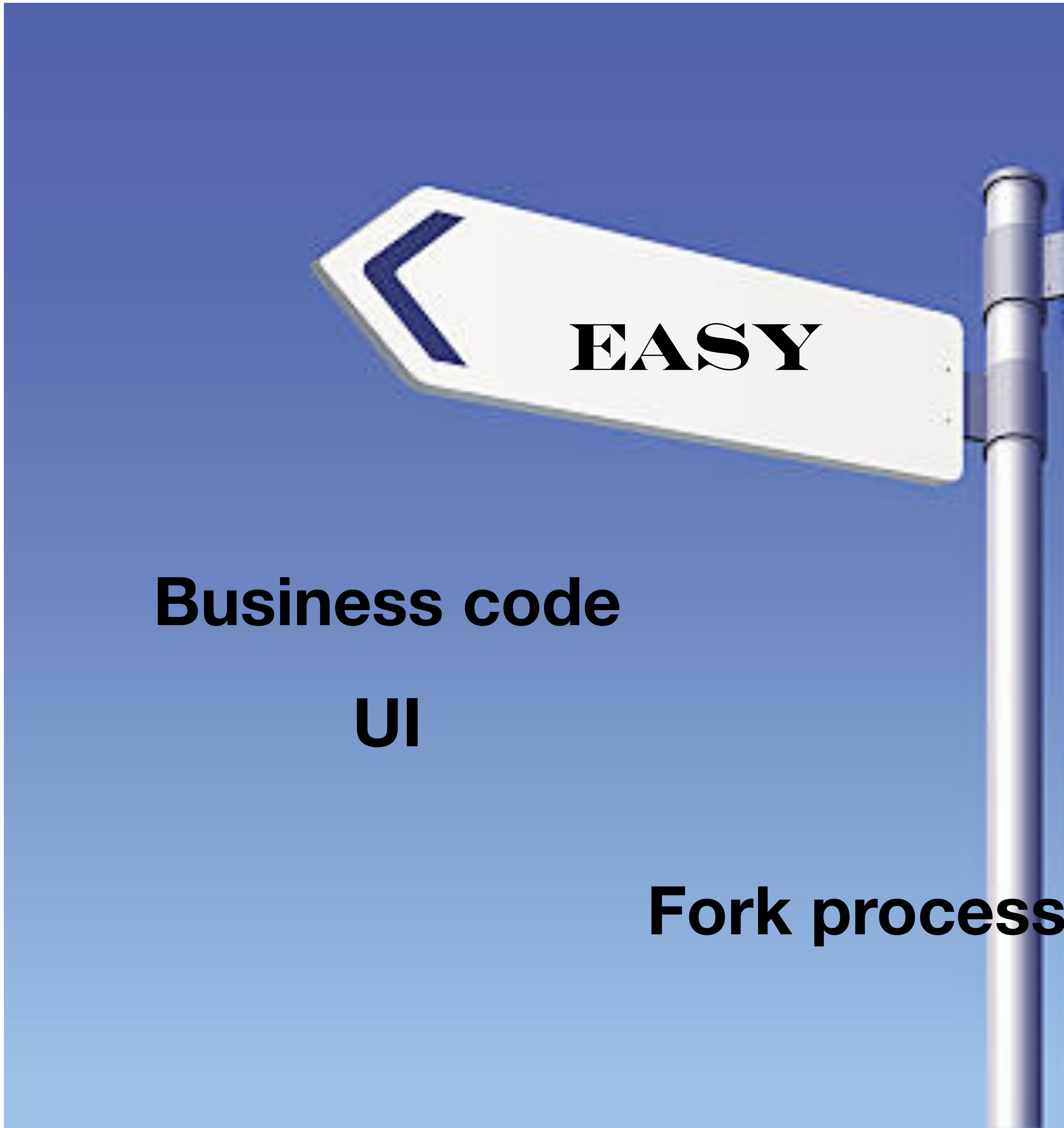
Packaging

Fork processes

Ensure it works on all environments

Pharo Launcher

Pharo provides very useful abstractions



Business code

UI

Fork process

FileLocator

- Late-bound reference to a file or directory in relation to a well-known location on the filesystem, called an origin.

- When asked to perform concrete operation, look up the current location of my origin, and resolve path against it.

examples:

```
FileLocator home
FileLocator desktop
FileLocator workingDirectory
```

FileReference

- abstraction on a file or folder
- independent of the underlying filesystem
- can be combined with FileLocator

```
FileLocator home / ‘foo’                {home}/foo

(FileLocator home / ‘foo’) fullName   '/Users/demarey/foo'
```

OSPlatform

Get information on the underlying Operating System

- OSPlatform current family #Unix #Windows #MacOS #MacOSX
- OSPlatform current isUnix32
- OSPlatform current isWindows64

Environment variables

Read and set values of system environment variables

- OSEnvironment current getEnv: ‘PATH’
- OSEnvironment current setEnv: ‘FOO’ value: ‘bar’

Fork processes

Pharo Launcher needs to fork processes
and manage them

- wait for a process completion, exit code, capture std output

Fork processes

OSProcess

VM plugin, huge, lot of features

- OSProcess command: 'ls -l /etc'
- OSProcess waitForCommand: 'unzip foo.zip'

OSSubProcess

small, FFI-based

-
  ```
  OSSUnixSubprocess new
    command: '/bin/ls';
    arguments: #('-la' '/Users');
     runAndWait
  ```

ProcessWrapper

focused on Windows, comes with a DLL

- ProcessWrapper new startWithShellCommand: 'dir C:\'

Fork processes

Pharo Launcher needs to fork processes
and manage them

- wait for a process completion, exit code, capture std output
- use of OSProcess on Linux / Mac
- use of ProcessWrapper on Windows
  - got problems with some processes not being run or frozen => run the command in background

Pharo Launcher

Packaging

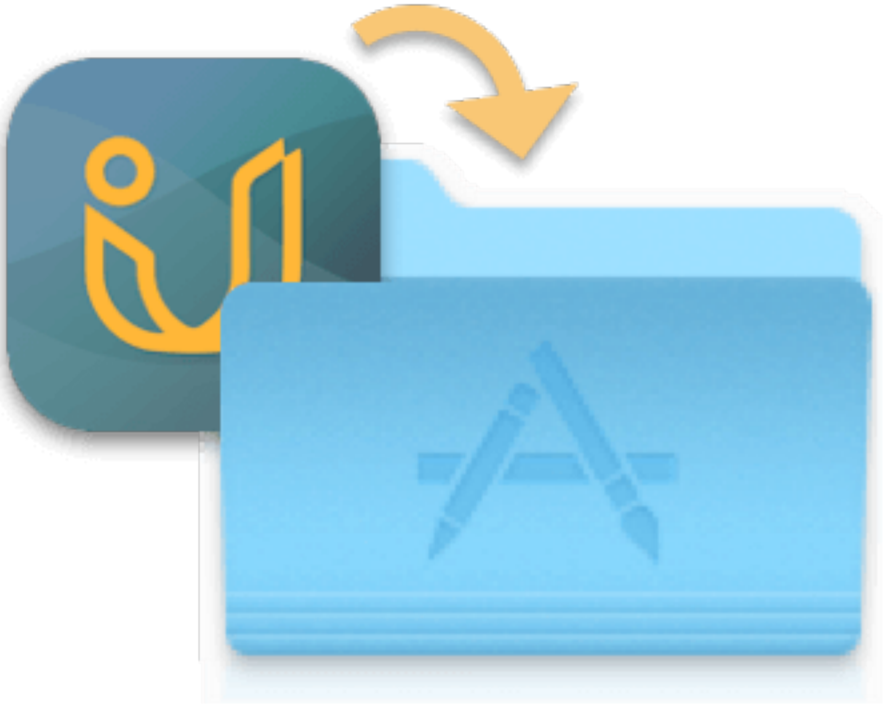
Fork processes

Ensure it works on all environments

« Read Only » app

Problem up to Pharo 6.1 when trying to write files where the application is installed

- pharo-local folder
  - ombu-sessions
  - package-cache
  - play-cache
  - play-stash
- PharoDebug.log
- stdio



« Read Only » app

Solution (since Pharo7.0)

App that does not write into the installation folder but rather in the user app data folder.

Tool to ease the conversion of a dev image to a production image

GitHub

https://github.com/VincentBlondeau/Cruiser

Cruiser:
A Tool to Package Pharo Applications

Vincent BLONDEAU

9/11/2018

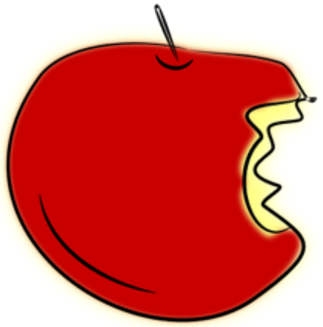
Lam RESEARCH

Lam Research Corp. 1

App packaging

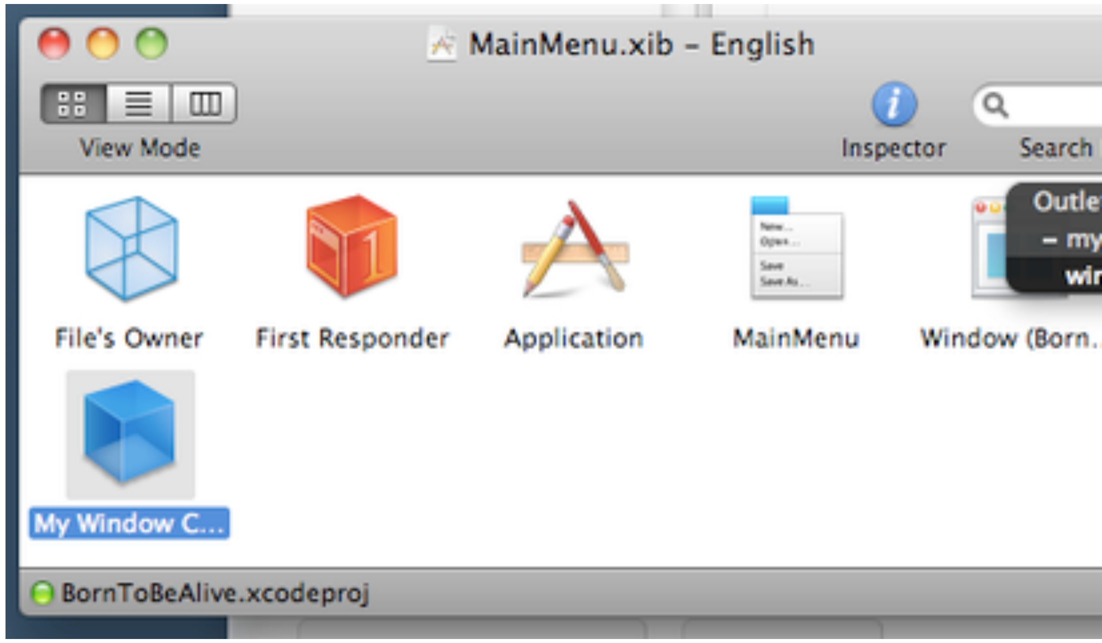
Needed for a smooth installation

OS X packaging



Build an OS X app

- re-use of material provided by the VM
- need a nib / xib file
- need a plist file
    - some properties are customizable



Not possible to embed an OS X app into an OS X app

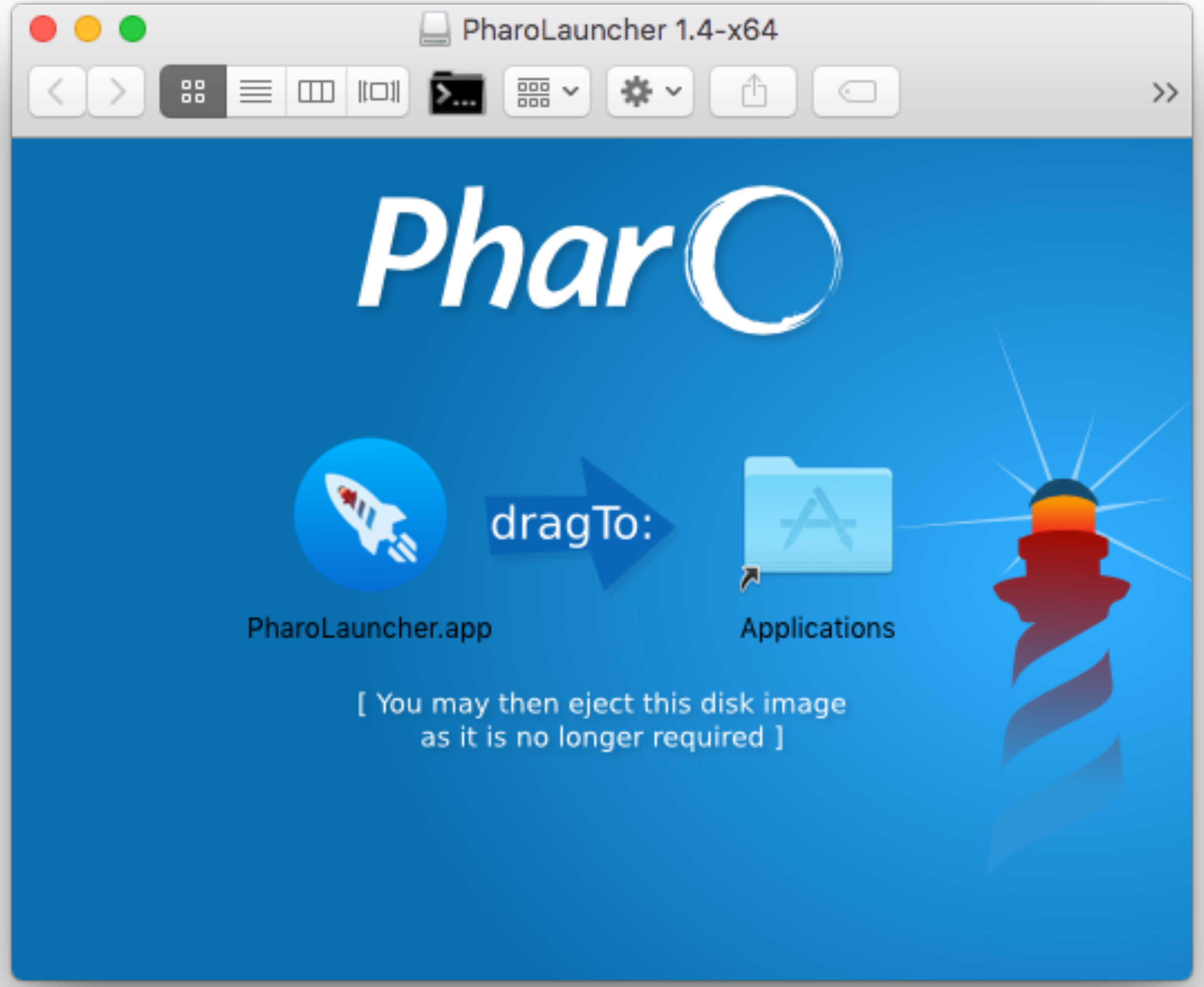
- => need to use VM app
- possible to have it prepared to become a Pharo app?

PharoLauncher.app

OS X packaging

DMG file

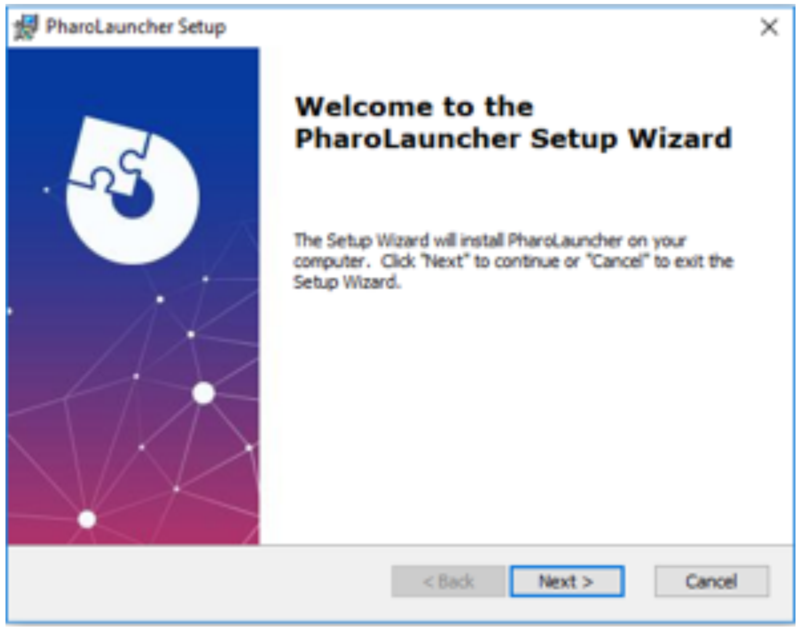
- prepare icons and background images
- copy data
- prepare the volume



[You may then eject this disk image
as it is no longer required]

Windows packaging

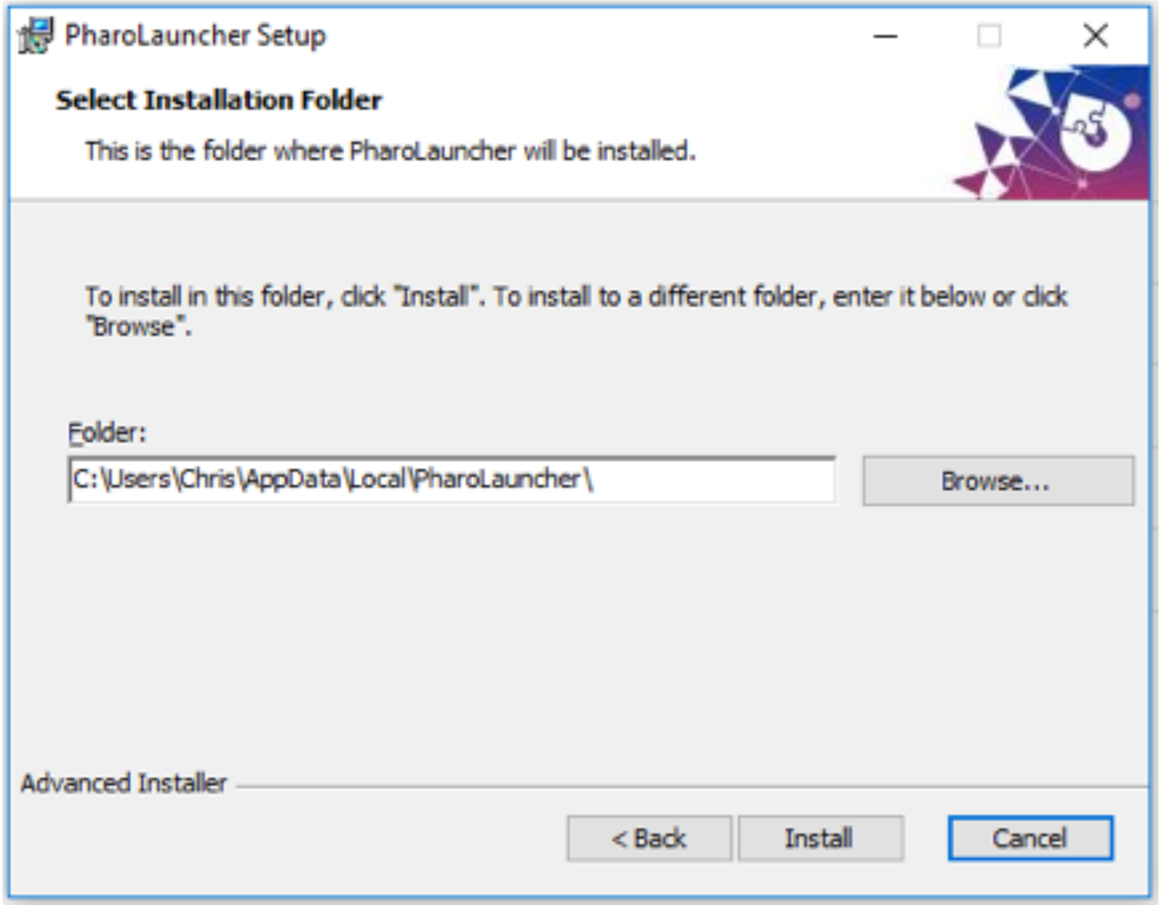
- Need an installer dealing with UAC (user account control)
  - installation without admin privileges
  - installation in a specific folder



PharoLauncher Setup

Welcome to the PharoLauncher Setup Wizard

The Setup Wizard will install PharoLauncher on your computer. Click "Next" to continue or "Cancel" to exit the Setup Wizard.



PharoLauncher Setup

Select Installation Folder

This is the folder where PharoLauncher will be installed.

To install in this folder, click "Install". To install to a different folder, enter it below or click "Browse".

Folder: C:\Users\Chris\AppData\Local\PharoLauncher\

Advanced Installer

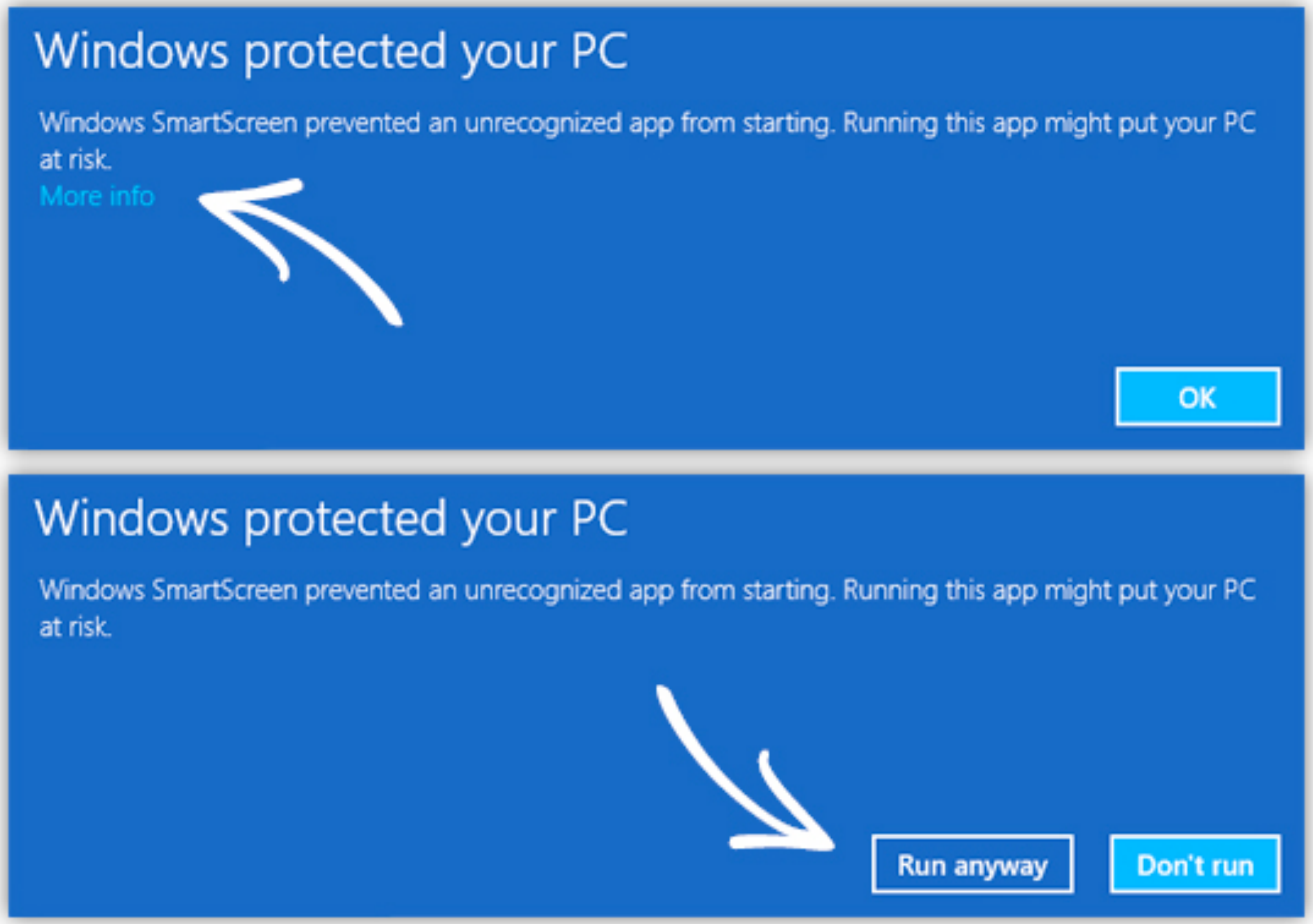
free for basic usage

Advanced Installer

Linux packaging

- Distribution via a zip file
- Better option would be to build distribution-specific packages, i.e. .deb, .rpm, etc.
  - take a lot of time to set up
  - first needed for Pharo itself

App trusted by the Operating System

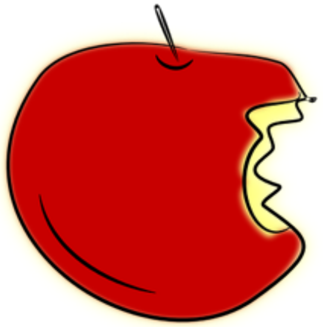


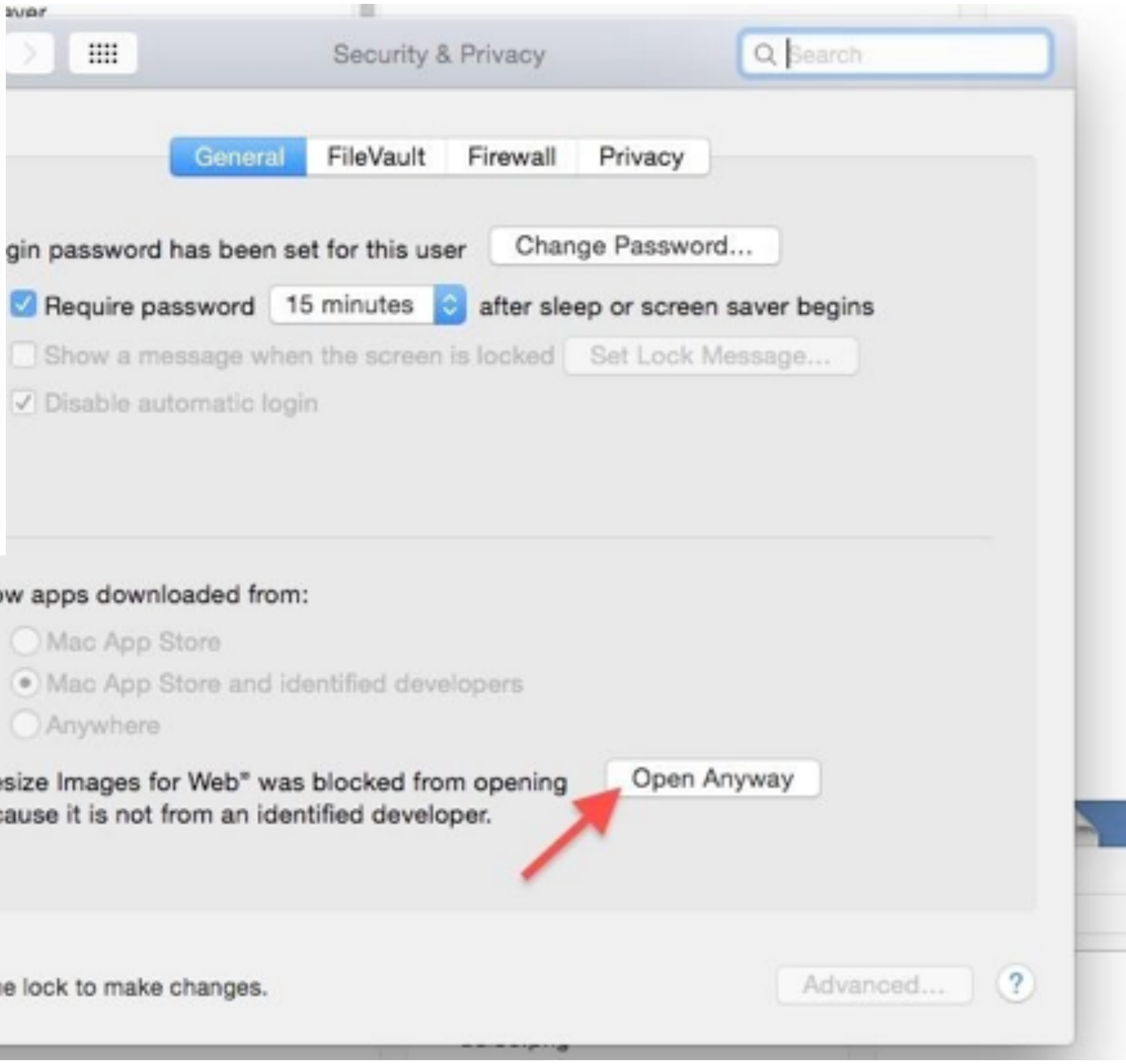
Windows protected your PC

Windows SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

Windows protected your PC

Windows SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.





gin password has been set for this user Change Password...

☑ Require password 15 minutes after sleep or screen saver begins

□ Show a message when the screen is locked Set Lock Message...

☑ Disable automatic login

Allow apps downloaded from:

○ Mac App Store

● Mac App Store and identified developers

○ Anywhere

"Resize Images for Web" was blocked from opening because it is not from an identified developer. Open Anyway

Click the lock to make changes.

App trusted by the Operating System

Solution: sign executables and installer

Need a certificate issued by a trusted authority

- developer certificate on OS X
- publicly issued code signing certificate

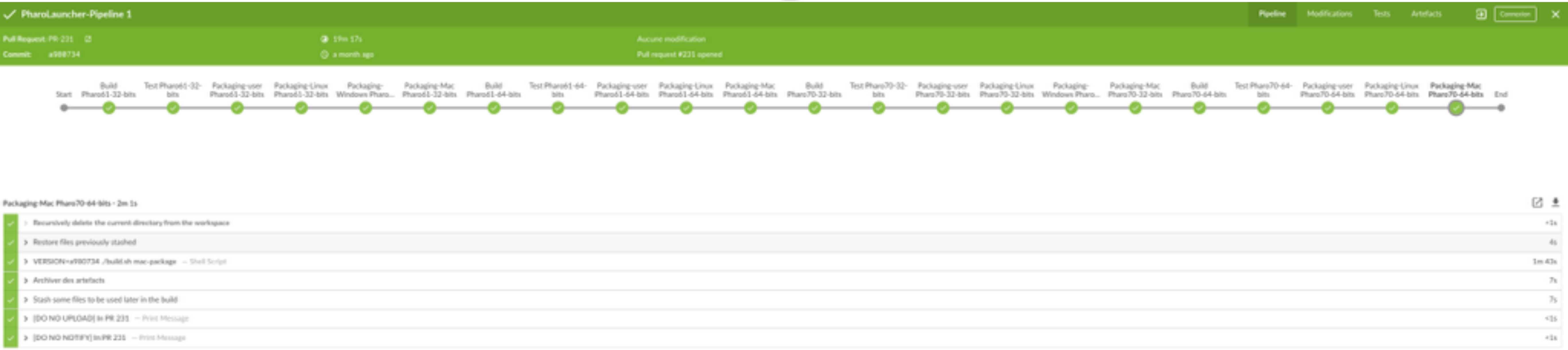
Sign executable with tools provided by the OS

Automate the process (need to store certificates safely)

Pharo Launcher signing documentation:
https://github.com/pharo-project/pharo-launcher/tree/development/signing

Testing / Building on all platforms

- How to ensure that new versions still work on all supported platforms?


Pipeline
Modifications
Tests
Artefacts
Connexion
Start
Build Pharo61-32-bits
Test Pharo61-32-bits
Packaging-user Pharo61-32-bits
Packaging-Linux Pharo61-32-bits
Packaging-Windows Pharo...
Packaging-Mac Pharo61-32-bits
Build Pharo61-64-bits
Test Pharo61-64-bits
Packaging-user Pharo61-64-bits
Packaging-Linux Pharo61-64-bits
Packaging-Mac Pharo61-64-bits
Build Pharo70-32-bits
Test Pharo70-32-bits
Packaging-user Pharo70-32-bits
Packaging-Linux Pharo70-32-bits
Packaging-Windows Pharo...
Packaging-Mac Pharo70-32-bits
Build Pharo70-64-bits
Test Pharo70-64-bits
Packaging-user Pharo70-64-bits
Packaging-Linux Pharo70-64-bits
Packaging-Mac Pharo70-64-bits
End

✓ PharoLauncher-Pipeline 1

Pull Request: PR-231
Commit: a988734

19m 17s
a month ago

Aucune modification
Pull request #231 opened

Packaging-Mac Pharo70-64-bits - 2m 1s

```
✓ › Recursively delete the current directory from the workspace                 <1s
✓ › Restore files previously stashed                                             4s
✓ › VERSION=a980734 ./build.sh mac-package — Shell Script                    1m 43s
✓ › Archiver des artefacts                                                       7s
✓ › Stash some files to be used later in the build                               7s
✓ › [DO NO UPLOAD] in PR 231 — Print Message                                    <1s
✓ › [DO NO NOTIFY] in PR 231 — Print Message                                    <1s
```

- How to test installers?
  - by hand ?
  - better solution?

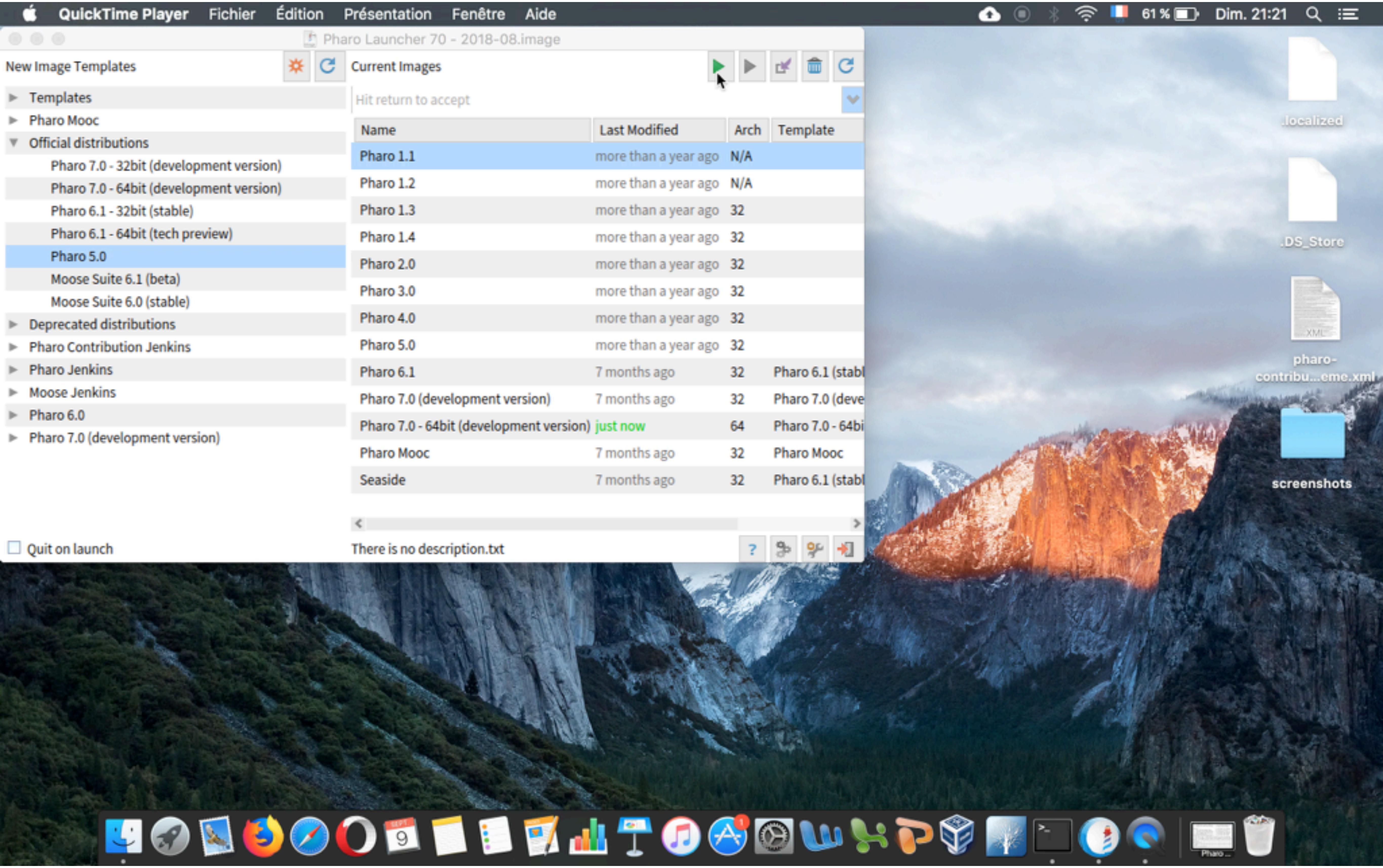
Issues

- How to reproduce issues report?
  - most functionalities covers by unit / functional tests
  - most problems comes from environment
    - different OS version
    - different configuration
    - etc.
    => need same OS version,
    same configuration to reproduce

Conclusion

Conclusion

- Writing a cross-platform app is doable!
- some classes ease a lot writing a cross-platform app
  => use them
- Testing problems
  - use CI as much as possible
  - involve users for detailed bug reports, co-investigation
- Would be nice to have an effort towards a packaging tool for Smalltalk / Pharo apps



Pharo Launcher 70 - 2018-08.image

New Image Templates

- Templates
- Pharo Mooc
- Official distributions
  - Pharo 7.0 - 32bit (development version)
  - Pharo 7.0 - 64bit (development version)
  - Pharo 6.1 - 32bit (stable)
  - Pharo 6.1 - 64bit (tech preview)
  - Pharo 5.0
  - Moose Suite 6.1 (beta)
  - Moose Suite 6.0 (stable)
- Deprecated distributions
- Pharo Contribution Jenkins
- Pharo Jenkins
- Moose Jenkins
- Pharo 6.0
- Pharo 7.0 (development version)

□ Quit on launch

Current Images

Hit return to accept

Name	Last Modified	Arch	Template
Pharo 1.1	more than a year ago	N/A	
Pharo 1.2	more than a year ago	N/A	
Pharo 1.3	more than a year ago	32	
Pharo 1.4	more than a year ago	32	
Pharo 2.0	more than a year ago	32	
Pharo 3.0	more than a year ago	32	
Pharo 4.0	more than a year ago	32	
Pharo 5.0	more than a year ago	32	
Pharo 6.1	7 months ago	32	Pharo 6.1 (stabl
Pharo 7.0 (development version)	7 months ago	32	Pharo 7.0 (deve
Pharo 7.0 - 64bit (development version)	just now	64	Pharo 7.0 - 64bi
Pharo Mooc	7 months ago	32	Pharo Mooc
Seaside	7 months ago	32	Pharo 6.1 (stabl

There is no description.txt