OpenCV infrastructure

Alexander Alekhin
Maksim Shabunin

Library

Crossplatform

- Windows / Linux / MacOSX
- ARM
- iOS
- Android
- Windows RT
- ???

Crossplatform tools

- CMake with cross-compilation support and several back-ends “generators”:
  - make, ninja
  - nmake, VS 2010/2012/2013/etc
  - Xcode
  - etc
- GTest for C++ code testing:
  - Linux, Windows, Android, Mac OS X
  - iOS, WinRT

Structure

- opencv/modules
  - core
  - imgproc
  - …
- opencv_contrib/modules
  - face
  - xfeatures2d
  - …

- <module>
  - include
  - src
  - test
  - perf
  - doc
  - tutorials
  - samples

CMake process

- determine compiler, libraries and options
- find modules
- sort modules (by dependencies)
- setup each module
- add general stuff (documentation, bindings, ...)
- generate result (make, ninja, Visual Studio, ...)

Tests

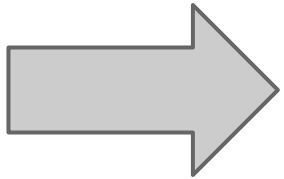
- accuracy
- performance
- java/android
- python
- build
- binary compatibility

```
[==========] Running 9230 tests from 176 test cases.
[----------] Global test environment set-up.
[----------] 6 tests from Core_Concatenation
[ RUN      ] Core_Concatenation.hconcat_empty_nonempty
[       OK ] Core_Concatenation.hconcat_empty_nonempty (0 ms)
[ RUN      ] Core_Concatenation.hconcat_nonempty_empty
[       OK ] Core_Concatenation.hconcat_nonempty_empty (0 ms)
[ RUN      ] Core_Concatenation.hconcat_empty_empty
[       OK ] Core_Concatenation.hconcat_empty_empty (0 ms)
[ RUN      ] Core_Concatenation.vconcat_empty_nonempty
[       OK ] Core_Concatenation.vconcat_empty_nonempty (0 ms)
```

```
[==========] Running 3568 tests from 120 test cases.
[----------] Global test environment set-up.
[----------] 48 tests from Size_MatType_ROp_reduceR
[ RUN      ] Size_MatType_ROp_reduceR.reduceR/0
[ PERFSTAT ]    (samples = 10, mean = 0.05, median = 0.05, stddev = 0.00 (1.7%))
[ VALUE    ]    (640x480, 8UC1, CV_REDUCE_SUM)
[       OK ] Size_MatType_ROp_reduceR.reduceR/0 (9 ms)
[ RUN      ] Size_MatType_ROp_reduceR.reduceR/1
[ PERFSTAT ]    (samples = 10, mean = 0.04, median = 0.04, stddev = 0.00 (0.4%))
[ VALUE    ]    (640x480, 8UC1, CV_REDUCE_AVG)
[       OK ] Size_MatType_ROp_reduceR.reduceR/1 (7 ms)
[ RUN      ] Size_MatType_ROp_reduceR.reduceR/2
[ PERFSTAT ]    (samples = 38, mean = 0.02, median = 0.02, stddev = 0.00 (2.7%))
[ VALUE    ]    (640x480, 8UC1, CV_REDUCE_MAX)
[       OK ] Size_MatType_ROp_reduceR.reduceR/2 (9 ms)
```

World build

```
libopencv_calib3d.so.3.0.0
libopencv_core.so.3.0.0
libopencv_features2d.so.3.0.0
libopencv_flann.so.3.0.0
libopencv_highgui.so.3.0.0
libopencv_imgcodecs.so.3.0.0
libopencv_imgproc.so.3.0.0
libopencv_ml.so.3.0.0
libopencv_objdetect.so.3.0.0
libopencv_photo.so.3.0.0
libopencv_shape.so.3.0.0
libopencv_stitching.so.3.0.0
libopencv_superres.so.3.0.0
libopencv_videoio.so.3.0.0
libopencv_video.so.3.0.0
libopencv_videostab.so.3.0.0
```



```
libopencv_world.so.3.0.0
```

3rd party libraries

```
GUI:
  QT:
  GTK+ 2.x:
  GThread :
  GtkGlExt:
  OpenGL support:
  VTK support:

Media I/O:
  ZLib:
  JPEG:
  WEBP:
  PNG:
  TIFF:
  JPEG 2000:
  OpenEXR:
  GDAL:
```

```
Video I/O:
  DC1394 1.x:
  DC1394 2.x:
  FFMPEG:
    codec:
    format:
    util:
    swscale:
    resample:
    gentoo-style:
  GStreamer:
  OpenNI:
  OpenNI PrimeSensor Modules:
  OpenNI2:
  PvAPI:
  GigEVisionSDK:
  UniCap:
  UniCap ucil:
  V4L/V4L2:
  XIMEA:
  Xine:
  gPhoto2:
```

```
Parallel framework:

Other third-party libraries:
  Use IPP:
       at:
  Use IPP Async:
  Use Eigen:
  Use Cuda:
  Use OpenCL:
```

Binary compatibility

Application
Binary
Interface

ABI compliance checker

Test Info

Library Name	opencv
Version #1	3.0.0
Version #2	3.0.0-261-gb09f591
CPU Type	x86_64
GCC Version	4.6
Subject	Binary Compatibility

Test Results

Total Header Files	136
Total Shared Libraries	16
Total Symbols / Types	2663 / 1513
Verdict	**Compatible**

Added Symbols (198)

imgcodecs.hpp, **libopencv_imgcodecs.so.3.0.0**
namespace **cv**
imread_reduced (String const& *filename*, int *flags*,

imgproc.hpp, **libopencv_imgproc.so.3.0.0**
namespace **cv**
spatialGradient (InputArray *src*, OutputArray *dx*, C

ocl.hpp, **libopencv_core.so.3.0.0**
namespace **cv::ocl**
attachContext (cv::String const& *platformName*, v
convertFromBuffer (void* *cl_mem_buffer*, size_t *s*
convertFromImage (void* *cl_mem_image*, cv::UM

opengl.hpp, **libopencv_core.so.3.0.0**
namespace **cv::ogl**
convertFromGLTexture2D (Texture2D const& *tex*
convertToGLTexture2D (cv::InputArray *src*, Textu
mapGLBuffer (Buffer const& *buffer*, int *accessFlag*
unmapGLBuffer (cv::UMat& *u*)

opengl.hpp, **libopencv_core.so.3.0.0**
namespace **cv::ogl::ocl**
initializeContextFromGL ( )

Documentation (doxygen)

```
/** @brief Calculates a square root of array elements.

The functions sqrt calculate a square root of each input a
In case of multi-channel arrays, each channel is processed
independently. The accuracy is approximately the same as c
std::sqrt .
@param src input floating-point array.
@param dst output array of the same size and type as src.
*/
CV_EXPORTS_W void sqrt(InputArray src, OutputArray dst);
```

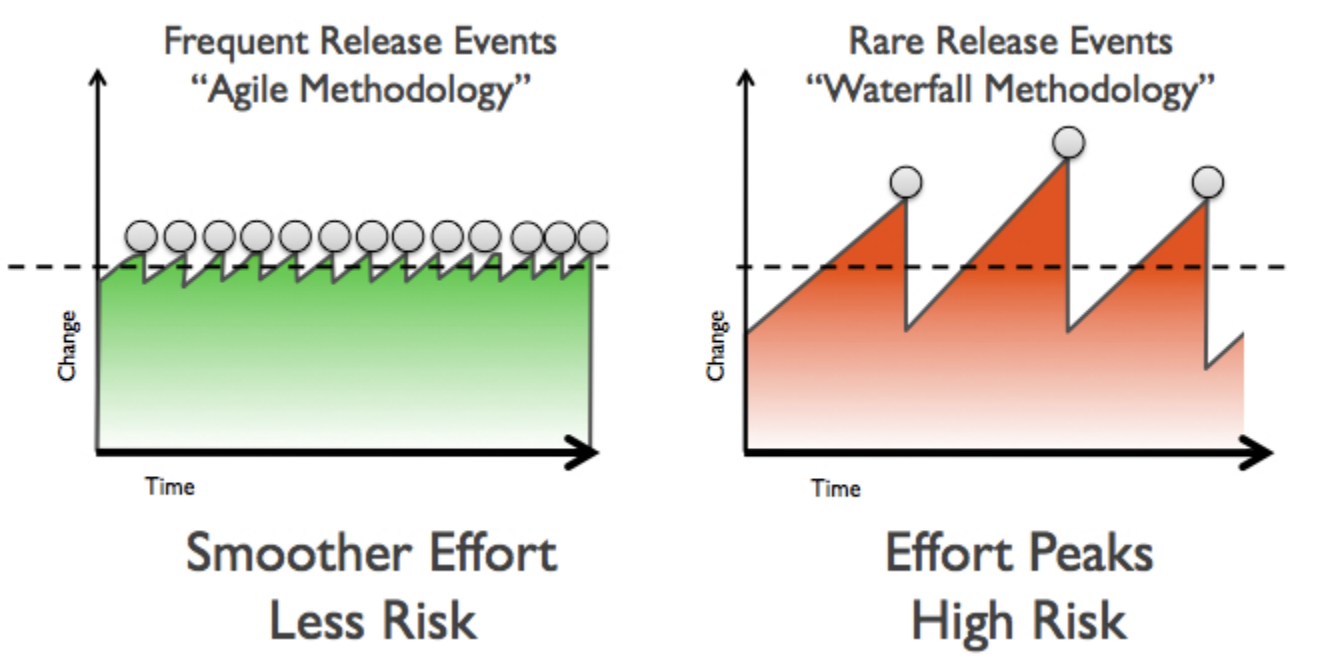
Parameters

src input floating-point array.

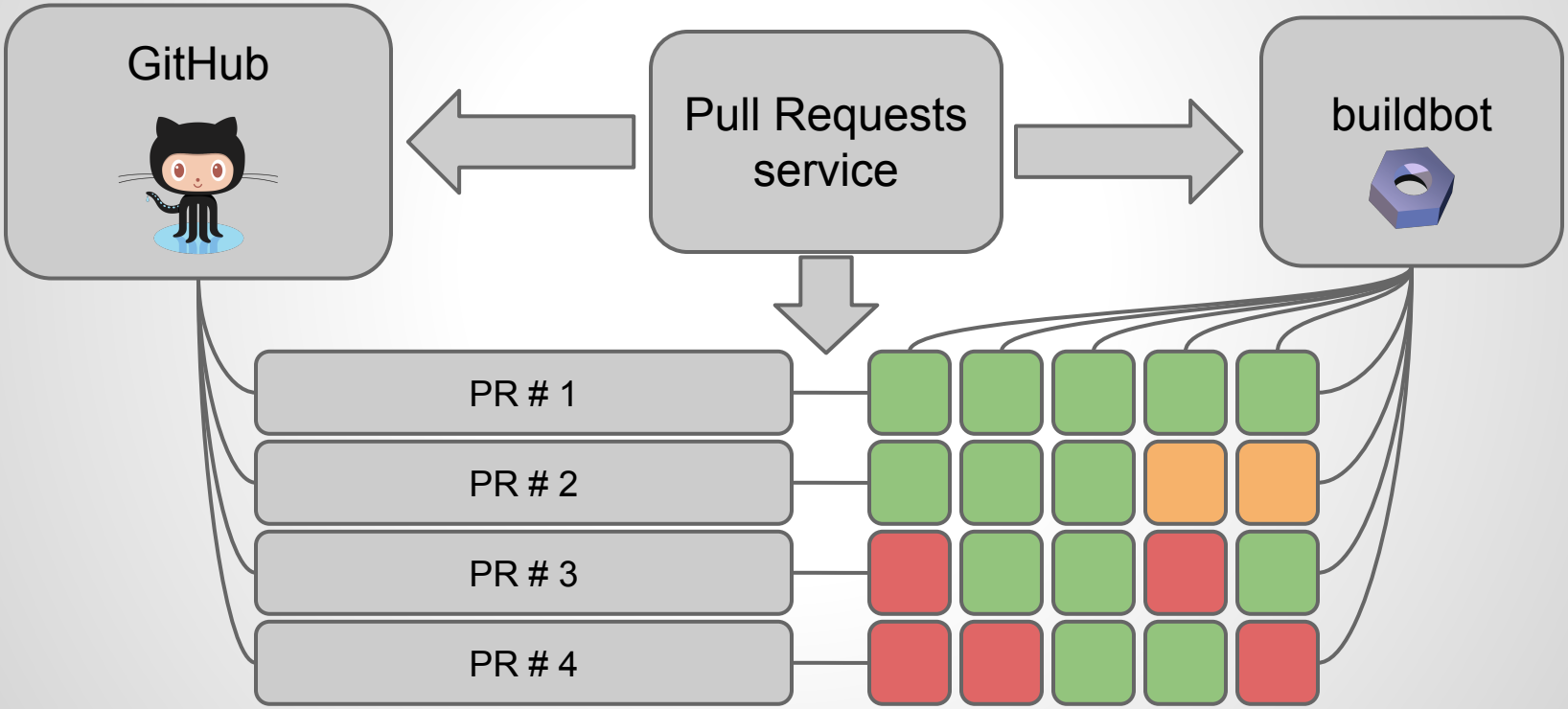
dst output array of the same size and type as src.

Continuous integration

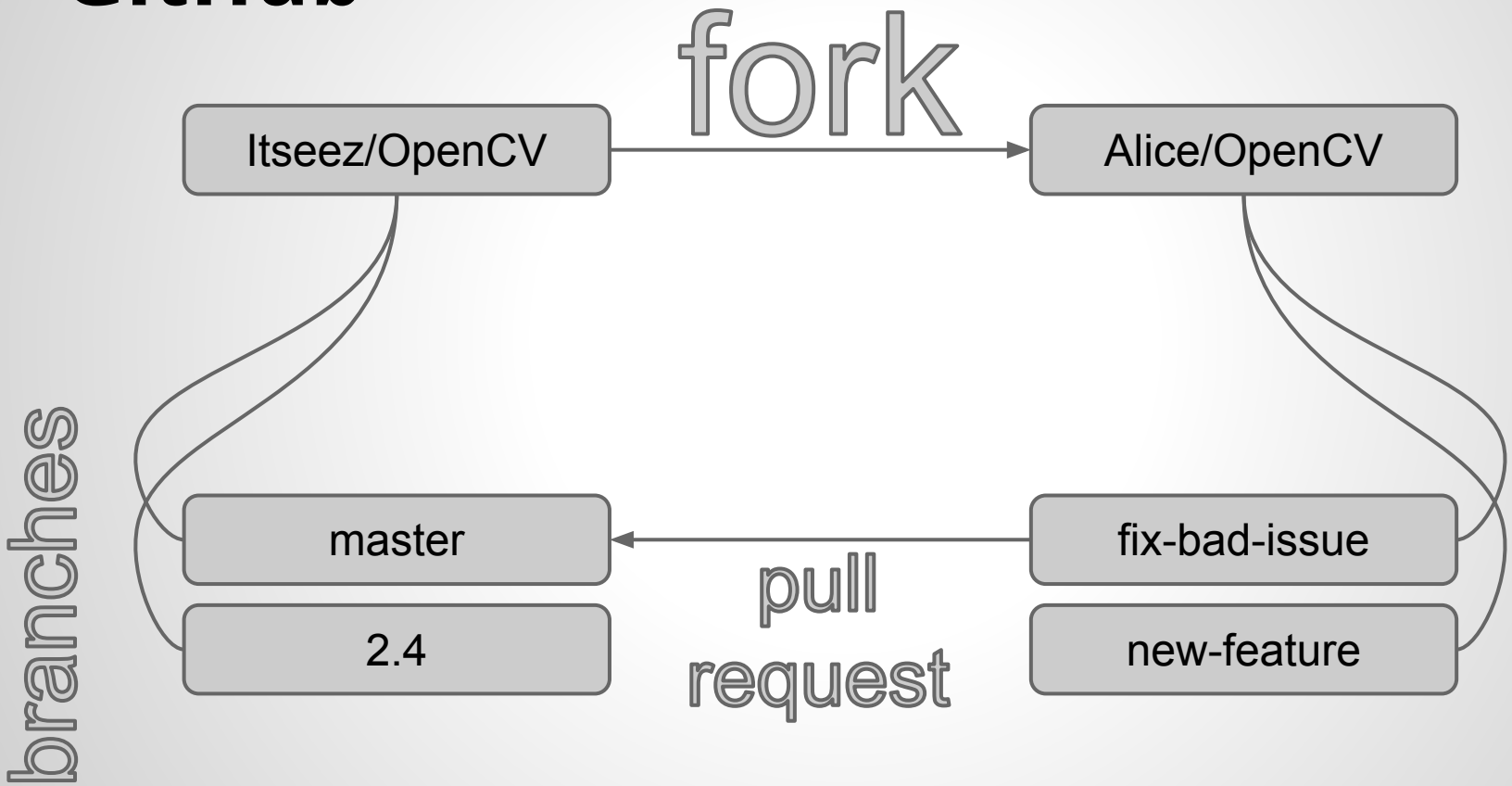
Agile vs. Waterfall



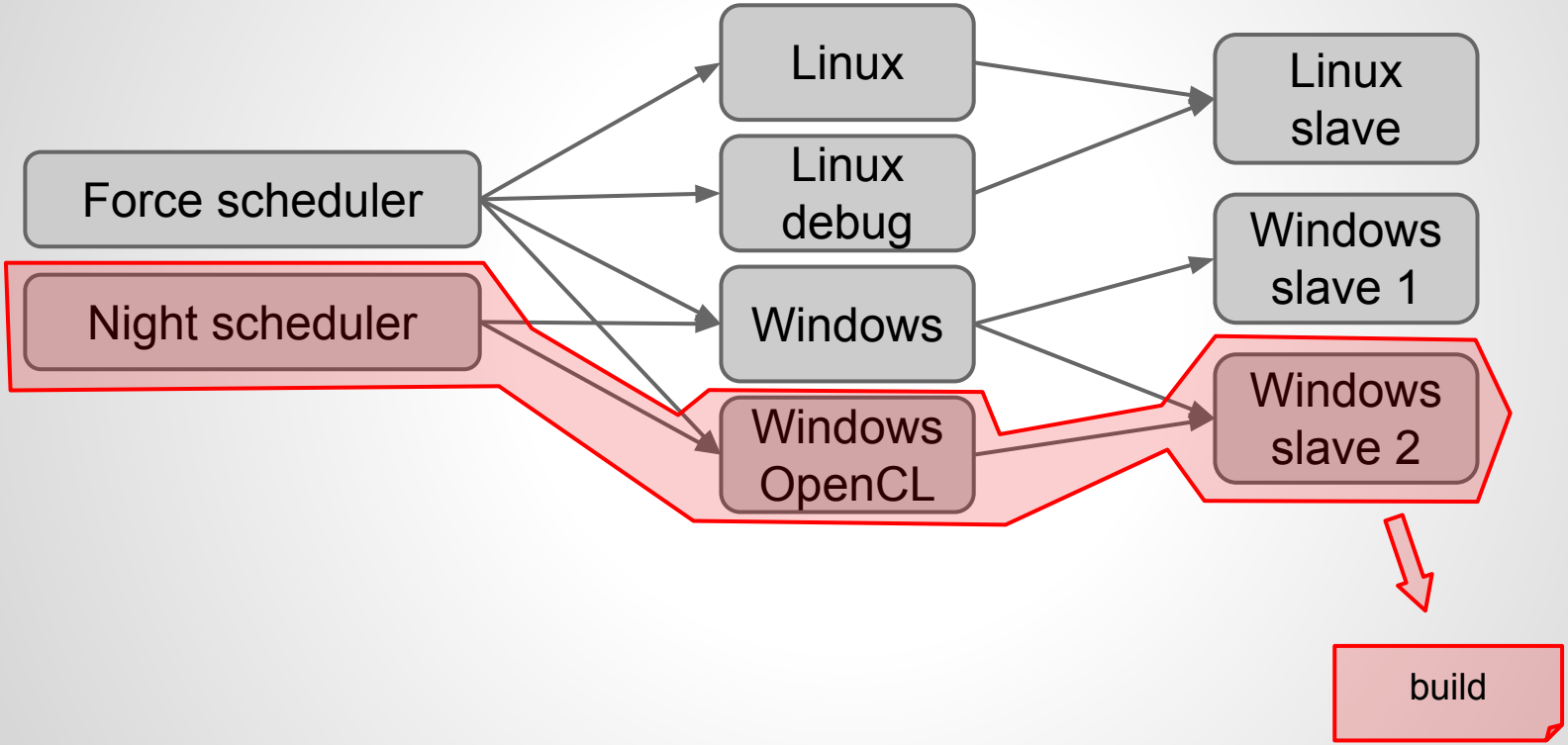
Pull Requests service



GitHub


fork
Itseez/OpenCV
Alice/OpenCV
branches
master
2.4
pull
request
fix-bad-issue
new-feature

Buildbot


Force scheduler
Night scheduler
Linux
Linux debug
Windows
Windows OpenCL
Linux slave
Windows slave 1
Windows slave 2
build

Buildbot configuration

```
# step 1: make clean; this fails if the slave has no local copy, but
# is harmless and will only happen the first time
makeclean = ShellCommand(name = "make clean",
                         command = ["make", "clean"],
                         description = "make clean")

# step 2: svn update (here updates trunk, see the docs for more
# on how to update a branch, or make it more generic).
checkout = SVN(baseURL = 'svn://myrepo/projects/coolproject/trunk',
               mode = "update",
               username = "foo",
               password = "bar",
               haltOnFailure = True )

# step 3: make all
makeall = ShellCommand(name = "make all",
                       command = ["make", "all"],
                       haltOnFailure = True,
                       description = "make all")
```

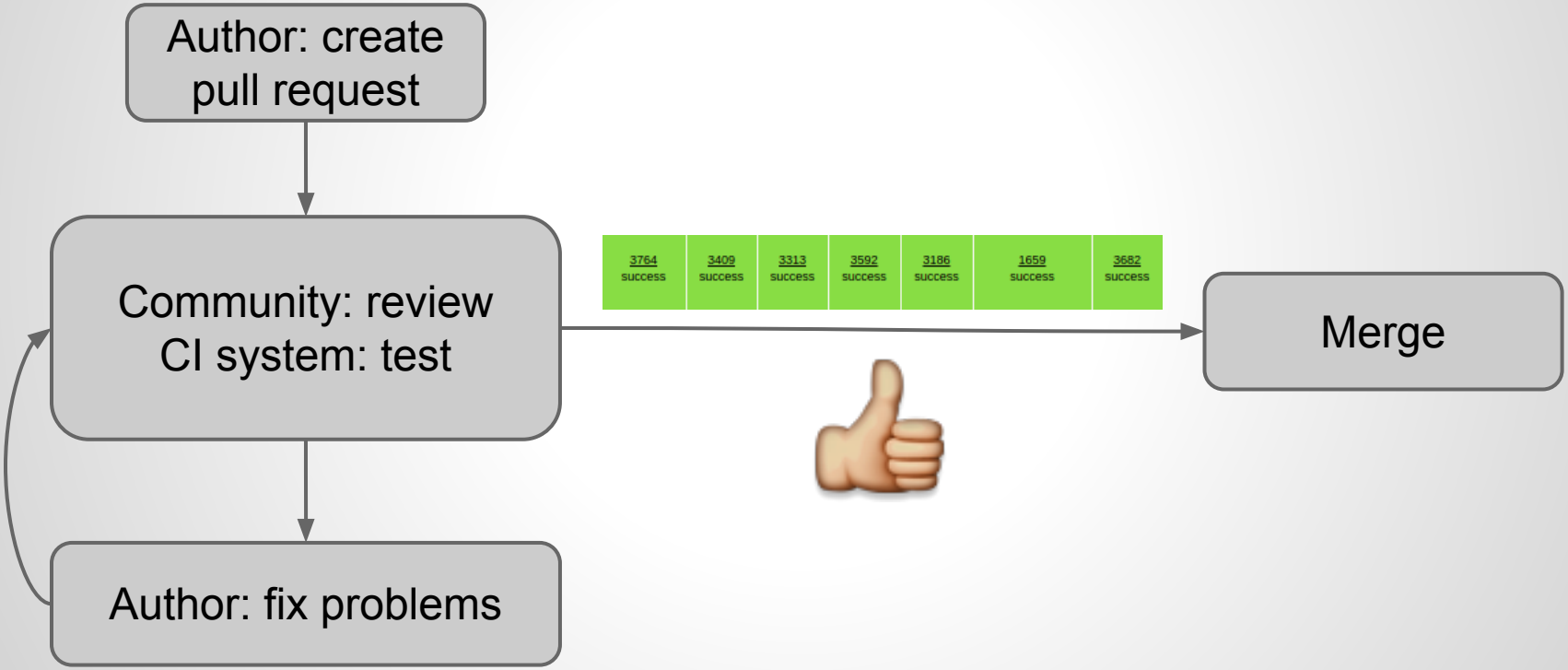
```
# create the build factory and add the steps to it
f_simplebuild = BuildFactory()
f_simplebuild.addStep(makeclean)
f_simplebuild.addStep(checkout)
f_simplebuild.addStep(makeall)
f_simplebuild.addStep(makepackages)
f_simplebuild.addStep(uploadpackages)

# finally, declare the list of builders. In this case, we o
c['builders'] = [
    BuilderConfig(name = "simplebuild", slavenames = ['slav
]
```

Build slaves

- Different OS: Windows / Linux / Mac OS
- Bare metal / Virtual Machine / Container
- Special HW (CUDA, OpenCL)
- Attached external devices (Android)

Contribution process



Other

Java wrapper details

```
package org.vamp_plugins;


public class Plugin
{
    public native String getIdentifier();
    public native String getName();
    public native String getDescription();
    public native int getPluginVersion();
}
```

```
jstring
Java_org_vamp_1plugins_Plugin_getIdentifier(JNIEnv *env, jobject obj)
{
    Plugin *p = getHandle<Plugin>(env, obj);
    return env->NewStringUTF(p->getIdentifier().c_str());
}
```

```
jint
Java_org_vamp_1plugins_Plugin_getPluginVersion(JNIEnv *env, jobject obj)
{
    Plugin *p = getHandle<Plugin>(env, obj);
    return p->getPluginVersion();
}
```

Python/Java wrappers

```
CV_EXPORTS_W void sqrt(InputArray src, OutputArray dst);
```

- advantages:
  - automatic
  - easy to add new functionality
- disadvantages:
  - C++ parsing is hard
  - documenting
  - memory management

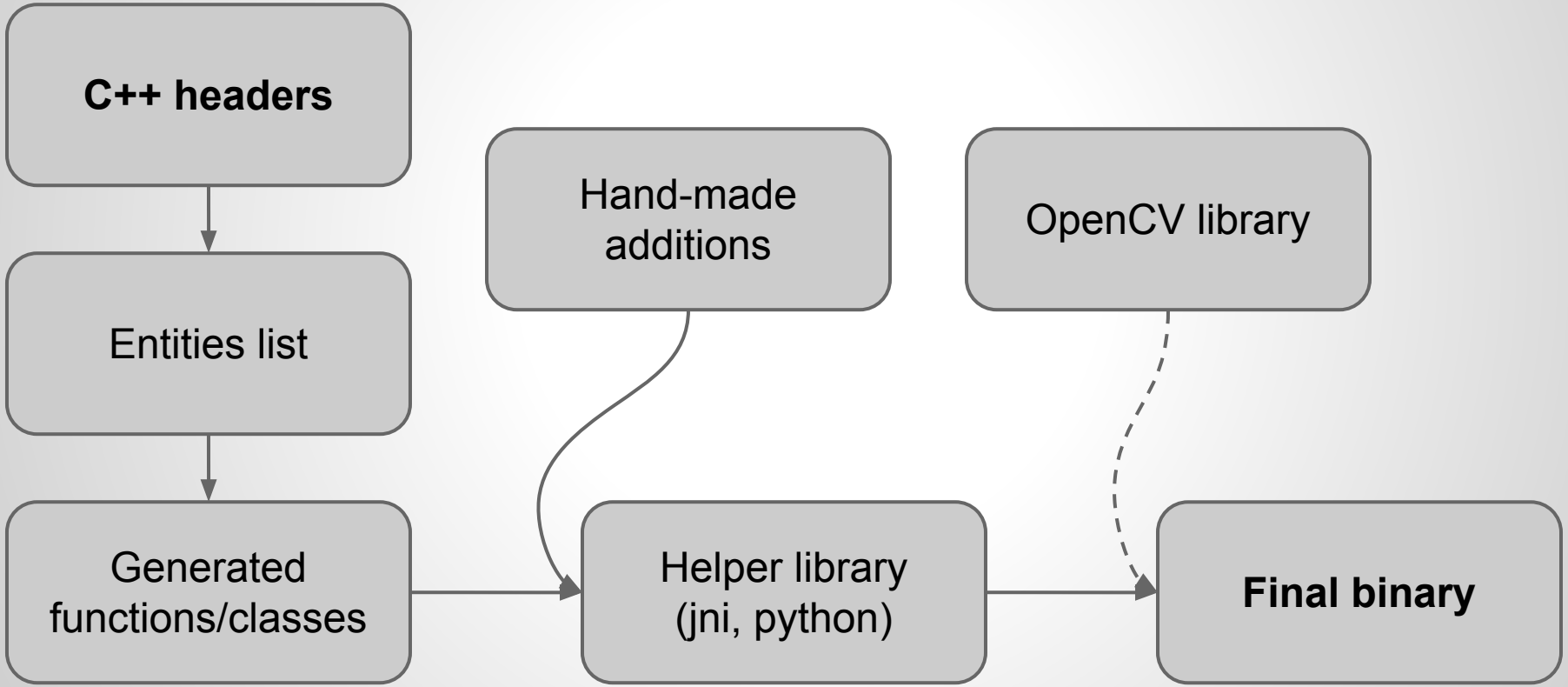
Java example

```
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.CvType;
import org.opencv.core.Scalar;
class SimpleSample {
  static{ System.loadLibrary(Core.NATIVE_LIBRARY_NAME); }
  public static void main(String[] args) {
    System.out.println("Welcome to OpenCV " + Core.VERSION);
    Mat m = new Mat(5, 10, CvType.CV_8UC1, new Scalar(0));
    System.out.println("OpenCV Mat: " + m);
    Mat mr1 = m.row(1);
    mr1.setTo(new Scalar(1));
    Mat mc5 = m.col(5);
    mc5.setTo(new Scalar(5));
    System.out.println("OpenCV Mat data:\n" + m.dump());
  }
}
```

Python example

```
import cv2
import numpy as np
import sys
src = cv2.imread(sys.argv[1], 1)
img = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
img = cv2.medianBlur(img, 5)
cimg = src.copy() # numpy function
circles = cv2.HoughCircles(img, cv2.HOUGH_GRADIENT, 1, 10
a, b, c = circles.shape
for i in range(b):
    cv2.circle(cimg, (circles[0][i][0], circles[0][i][1]),
    cv2.circle(cimg, (circles[0][i][0], circles[0][i][1]),
cv2.imshow("source", src)
cv2.imshow("detected circles", cimg)
cv2.waitKey(0)
```

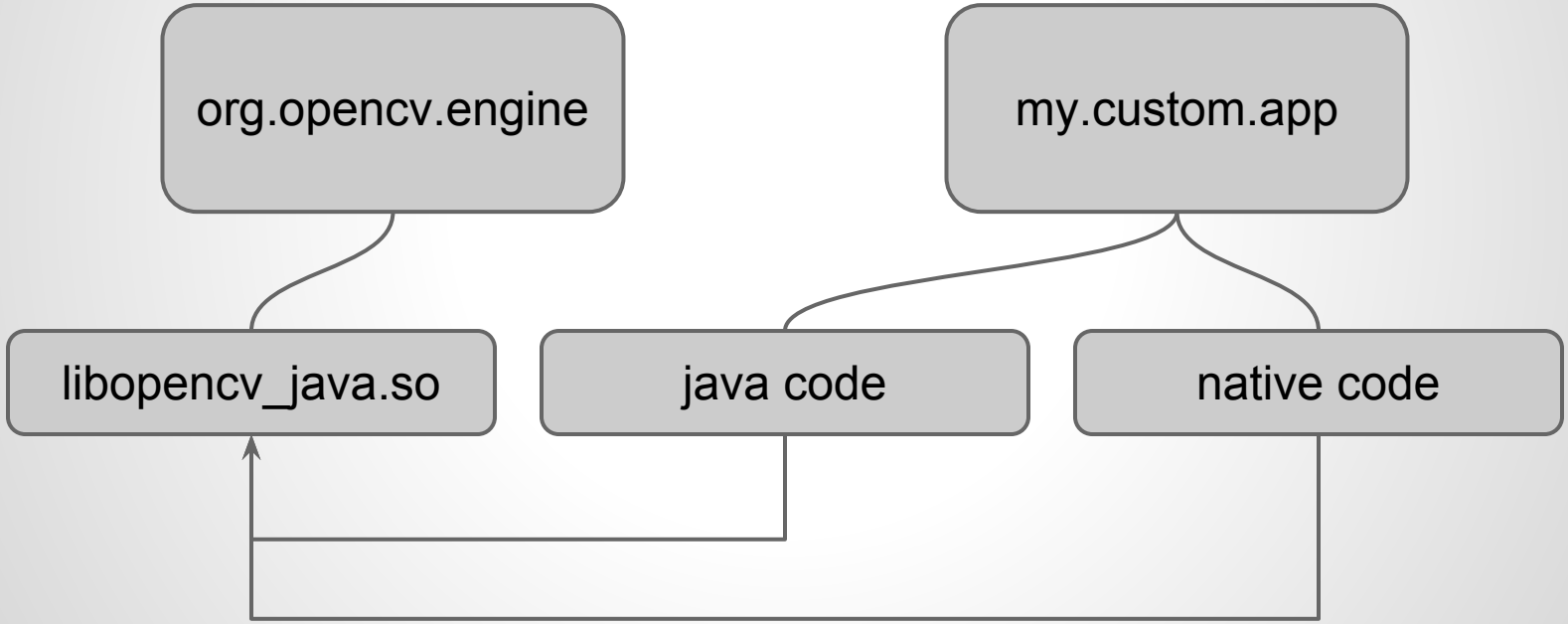
Python/Java wrappers



OpenCV for Android

- 7 native platforms (3 * arm, 2 * x86, 2 * mips)
- OpenCV SDK for Android
  - library (native + java)
  - samples
  - manager
  - documentation

OpenCV Manager for Android



www

issue tracker	code.opencv.org ⇒ github.com
wiki	code.opencv.org ⇒ github.com
forum	answers.opencv.org
online documentation	docs.opencv.org
news / social	opencv.org + FB/TW/G+
downloads	sourceforge.net/projects/opencvlibrary

Questions?