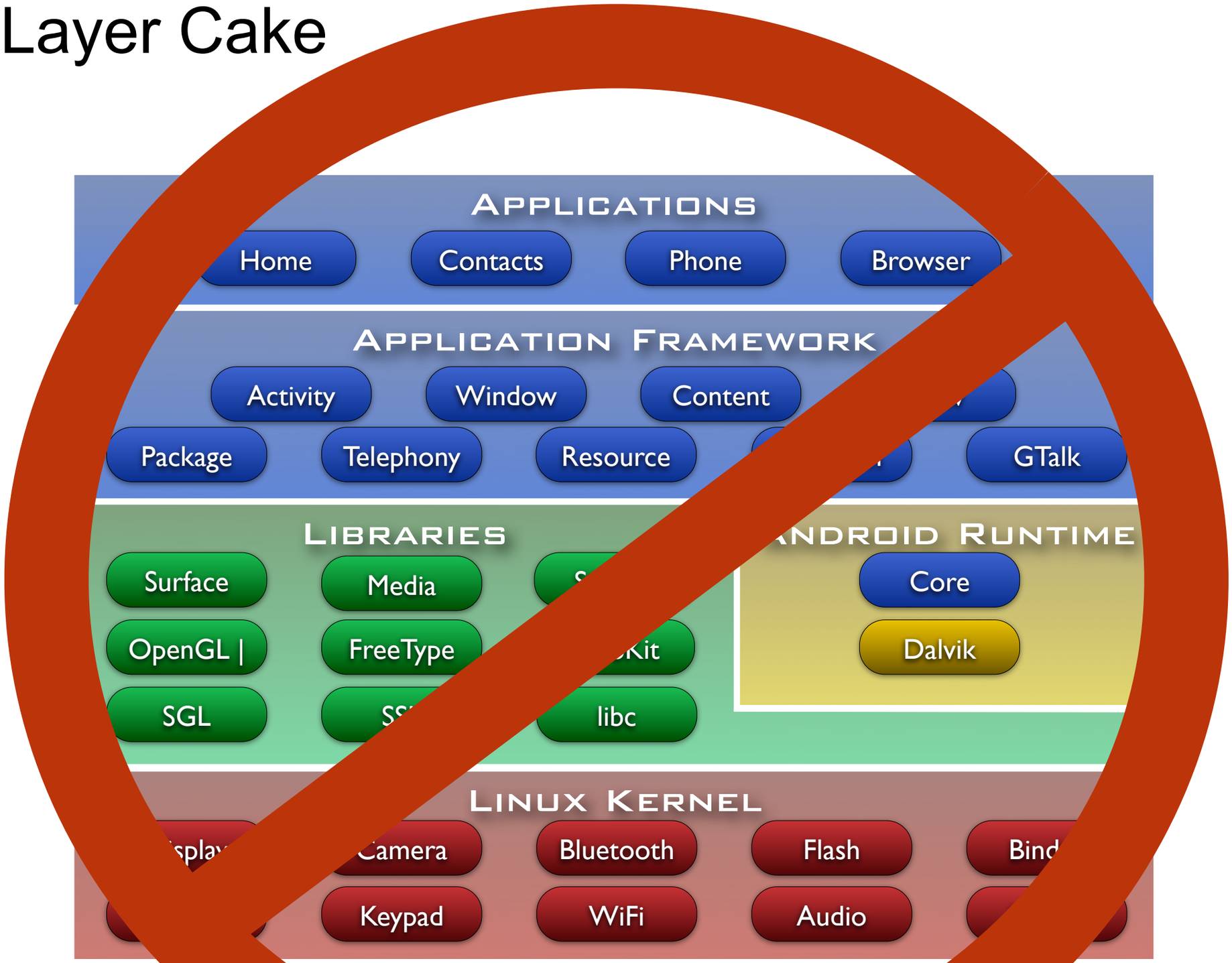
Google™ 09 I/O

How Do I Code Thee?
Let Me Count The Ways.

Dan Morrill
Google Developer Relations

What is Android?

The Layer Cake

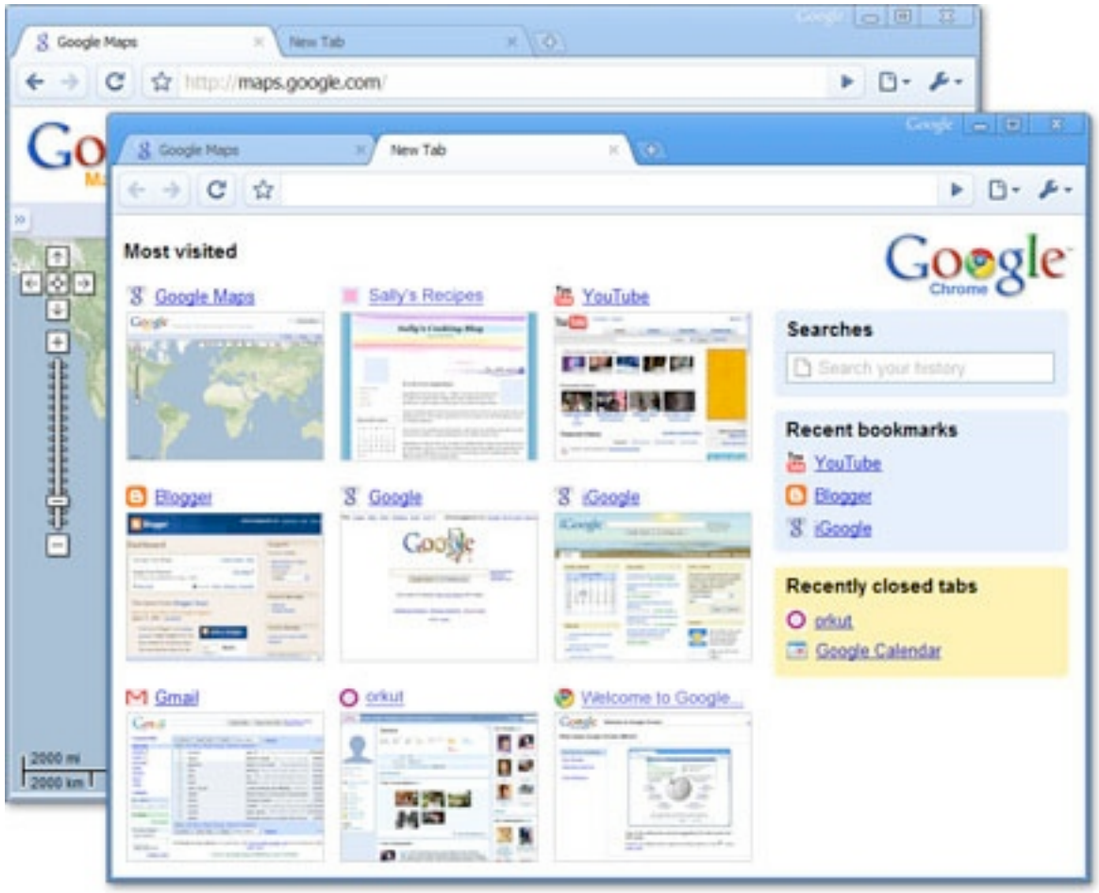


What is Android?

A chunk of code for making phone calls

Image by Evi Christodoulou; http://www.flickr.com/search/?q=startac&l=comm&ct=0

A robust network stack and Internet client



A Platform for Running Code

Image by sludgegulper; http://www.flickr.com/photos/sludgeulper/3441498831/

A Platform for Running Code

- Android has an application platform, true
- But we didn't create Android in an attempt to "own mobile"
- Android is a framework for interacting components
  - Apps can pull in pieces of other apps, the web, or even native

Image by genvessel; http://www.flickr.com/photos/genvessel/177133869/
Licensed under Creative Commons Attribution

"Give me your compiled, your ARM, your huddled bytecodes yearning to run free, the wretched refuse of your teeming store. Send these, the threadless, scripted stuff to me, I lift my lamp beside the coder’s door."

Agenda

- Introduce the Three Kinds
- Demonstrate how to use them
- Useful comparisons & statistics
- Future directions

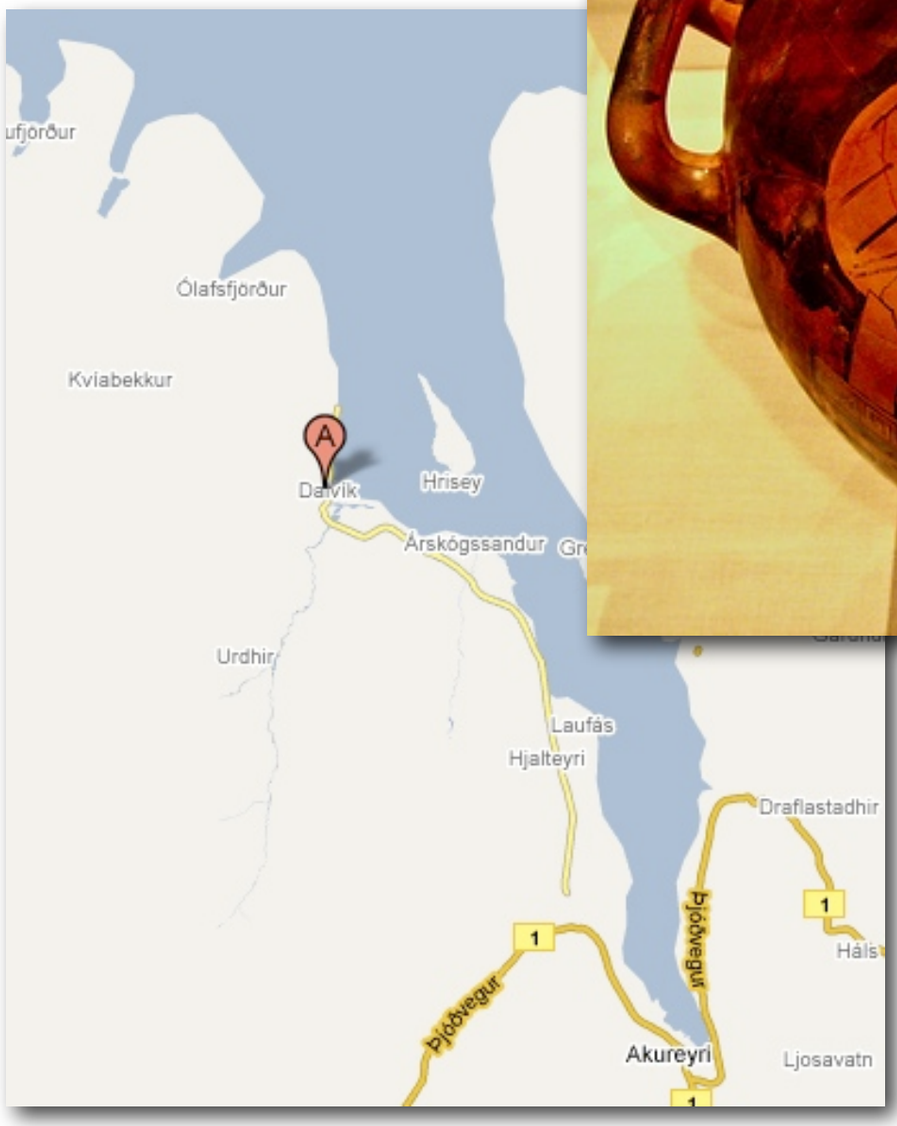
Anti-Agenda

- I will not...
  - Teach you how to write apps
  - Rehash stuff that's covered elsewhere
  - Pass judgment

Three Delicious Flavors

The (current) Three Kinds of Android Code

- Managed Code
- Ajax
- Native Code



Dalvik Code

Wait Wait, Hold Up

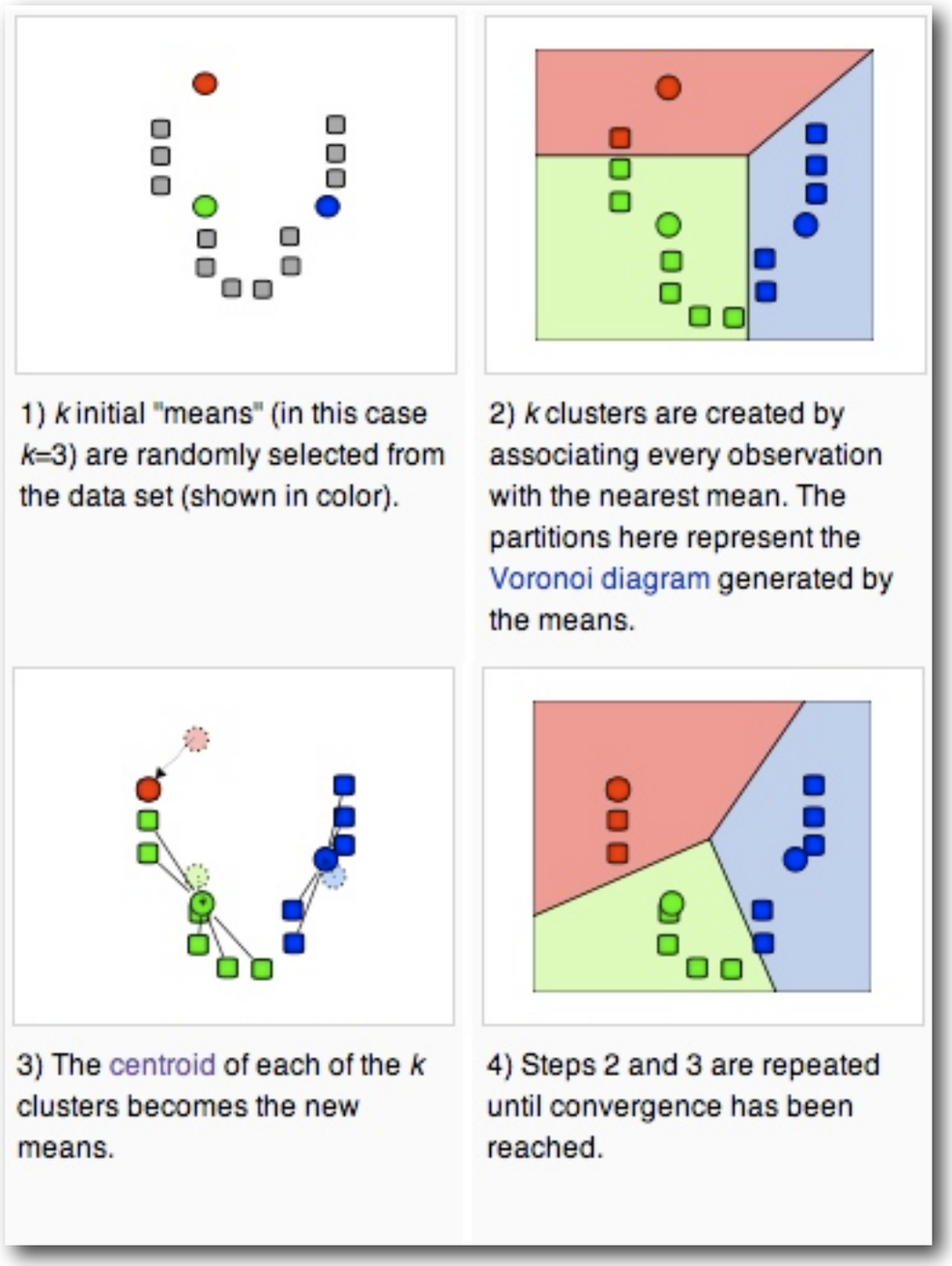
I Need a Victim

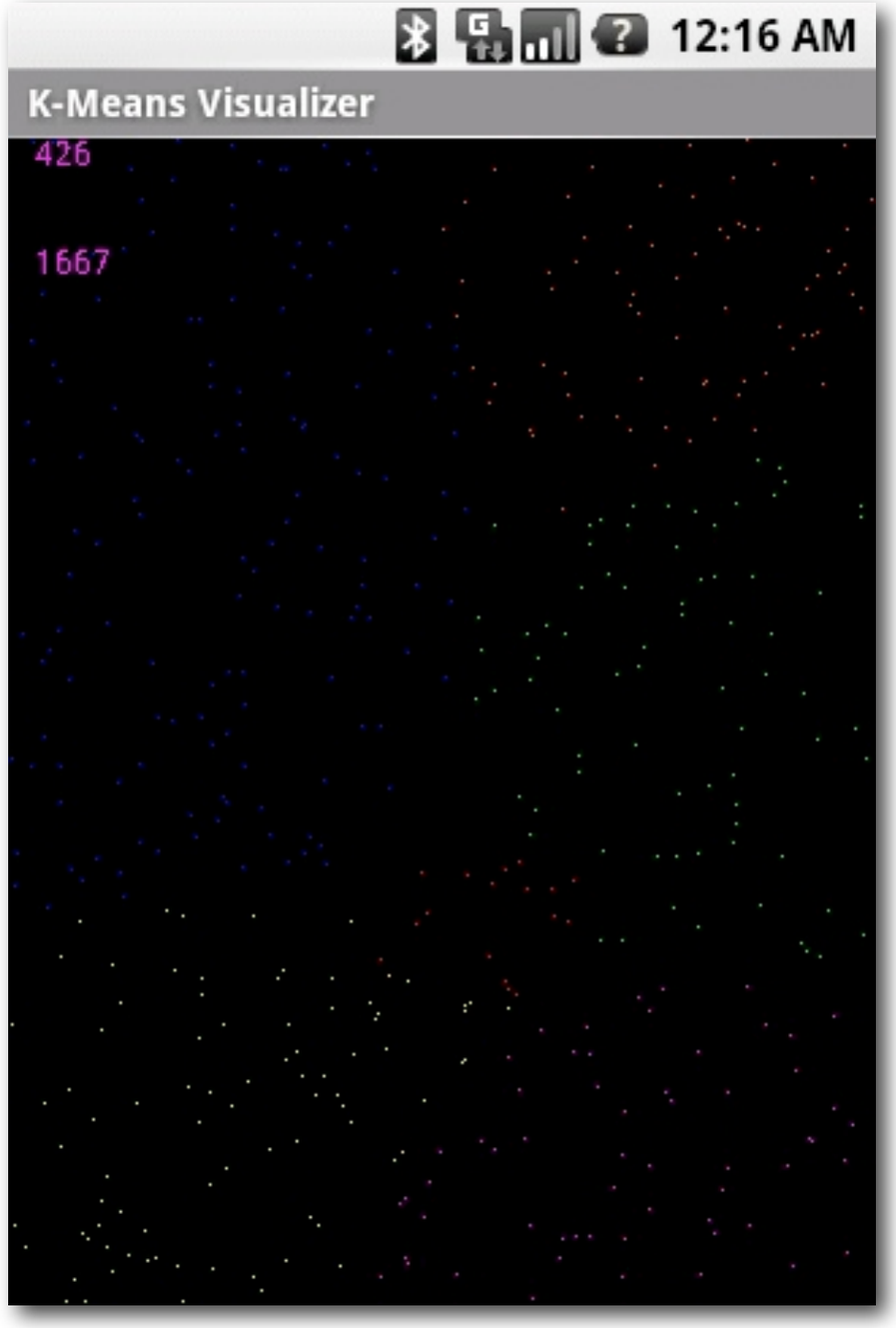
- Interesting but computationally significant
- Efficiently externalizable
- Must make for an interesting visualization

How about.... k-means clustering?

- Used to identify clusters in sets of points
- We'll use a simple 2D version
  - 500 randomly-generated points
  - 6 clusters (ROYGBV)
- Easy to visualize on screen
- 3 implementations
  - "Pure" Dalvik app (i.e. 100% Java source code)
  - Same visualization code with C algorithm implementation
  - JavaScript

K-Means Clustering in 2,000 Words



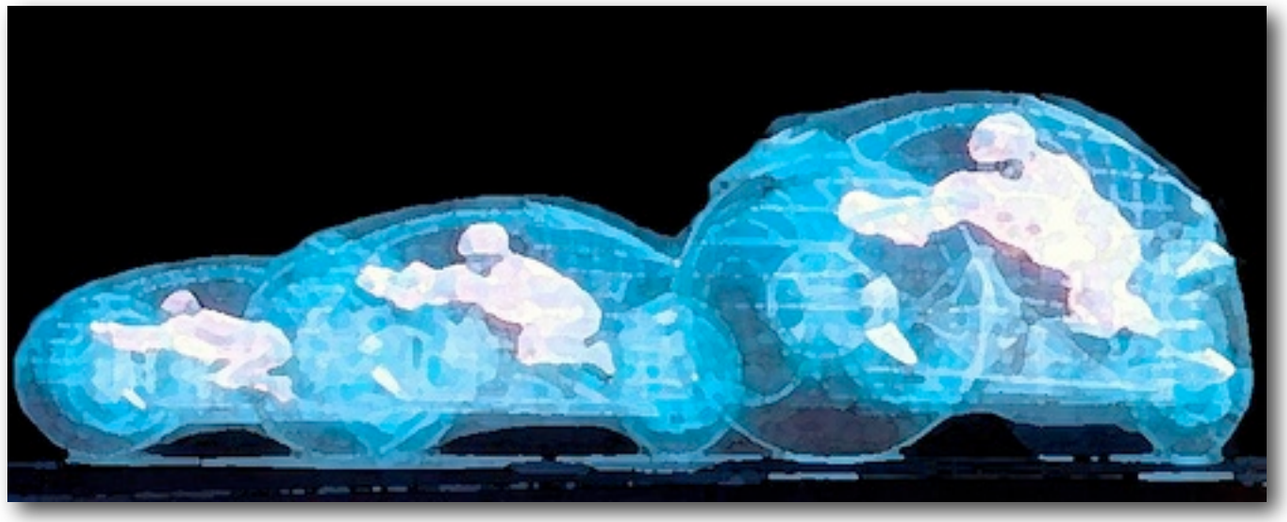


Source: http://en.wikipedia.org/wiki/K-means_clustering

Dalvik Code

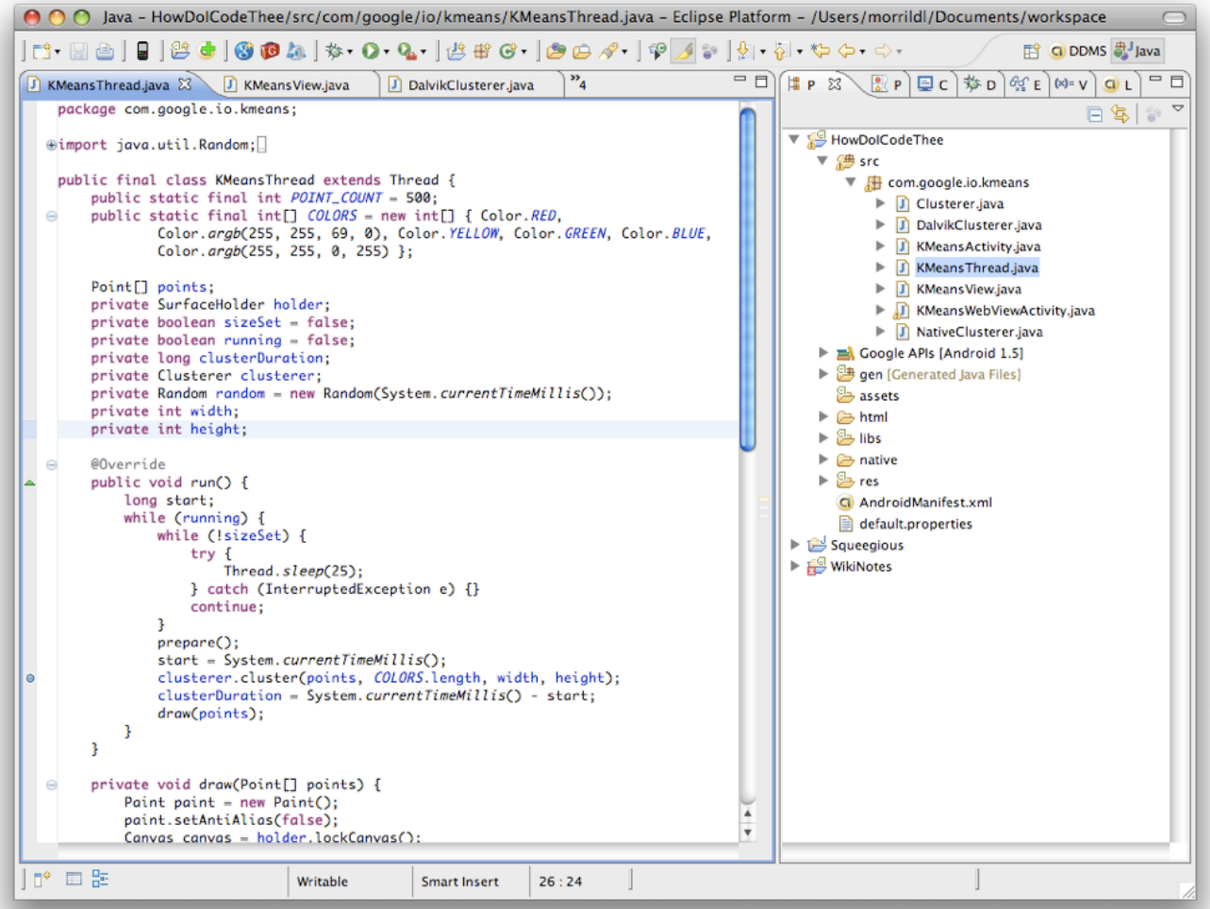
Dalvik - Overview

- Dalvik is a virtual machine, similar to the JVM or .Net CLR
- Memory-protected, garbage-collected, & lifecycle-managed
- Optimized for embedded machines
  - Built to reduce much of the need for a JITC
- Custom bytecode format; included translator from Java bytecode



Dalvik - Writing Apps

- Apps run on Dalvik, which provides core framework APIs
- APIs are backed by system infrastructure in native code
  - e.g. OpenGL, Binder IPC, media, etc.
- Popular tools are supported, and an Eclipse plugin is available



Dalvik - What Can You Do?

- Rich UIs
- Background services
- Shared components
- Tight integration with system events and UI
- ...but there are other sessions covering standard app dev in detail

Dalvik - What Can't You Do?

- Not much!
- Dalvik is the primary app platform -- the crossroads
- That does not necessarily mean it's right for your app, though
  - Some apps need raw speed
  - Some apps don't need tight integration

Dalvik - Demo

```
package com.google.io.kmeans;

import java.util.Random;

public final class KMeansThread extends Thread {
    public static final int POINT_COUNT = 500;
    public static final int[] COLORS = new int[] { Color.RED,
            Color.argb(255, 255, 69, 0), Color.YELLOW, Color.GREEN, Color.
            Color.argb(255, 255, 0, 255) };

    Point[] points;
    private SurfaceHolder holder;
    private boolean sizeSet = false;
    private boolean running = false;
    private long clusterDuration;
    private Clusterer clusterer;
    private Random random = new Random(System.currentTimeMillis());
    private int width;
    private int height;

    @Override
    public void run() {
        long start;
        while (running) {
            while (!sizeSet) {
                try {
                    Thread.sleep(25);
                } catch (InterruptedException e) {}
                continue;
            }
            prepare();
            start = System.currentTimeMillis();
            clusterer.cluster(points, COLORS.length, width, height);
            clusterDuration = System.currentTimeMillis() - start;
            draw(points);
        }
    }

    private void draw(Point[] points) {
        Paint paint = new Paint();
        paint.setAntiAlias(false);
        Canvas canvas = holder.lockCanvas();
        try {
            long start = System.currentTimeMillis();
            canvas.drawARGB(255, 0, 0, 0); // black the screen
            for (Point point : points) {
                paint.setColor(COLORS[point.cluster]);
                canvas.drawPoint((int)point.x, (int)point.y, paint);
            }
```

Dalvik - Examples

- Core system apps
- Currently (nearly) all apps on the Android Market

Dalvik - Future

- Improved (faster, better) garbage collection
- Just-In-Time compilation
- Optimizations in core libraries
- Additional APIs -- Bluetooth, P2P, and more

Ajax (Web) Apps

Ajax - Overview

- Apps are broken up into convenient declarative layouts & code
- JavaScript code mutates the DOM to create UI effects
- Network access is available via XMLHttpRequest
- Recently, <canvas> allows JavaScript to do direct painting

Ajax on Android - Overview

- Android's Browser is based on WebKit + SquirrelFish
- WebKit v528.5, equivalent to Safari 4 beta
  - Reports as 3.1.2 in the user-agent string -- oops!
- Includes Gears 0.5.17.0 (including location)
- Includes support for the <canvas> tag

Ajax - What Can You Do?

- Old sk00l
  - You can, of course, just build static web page apps

Ajax - What Can You Do?

- But you can also build dynamic UIs, via DOM or <canvas>
- You can fetch & store data on your origin server
- With HTML5, you can also:
  - Access location
  - Run code outside the main thread
  - Store data & pages locally

Ajax on Android - What Can't You Do?

- Android doesn't currently support HTML5, but does have Gears
- Background processing
  - Code only runs when Browser is open and your page is loaded
- Access system & framework APIs [1]

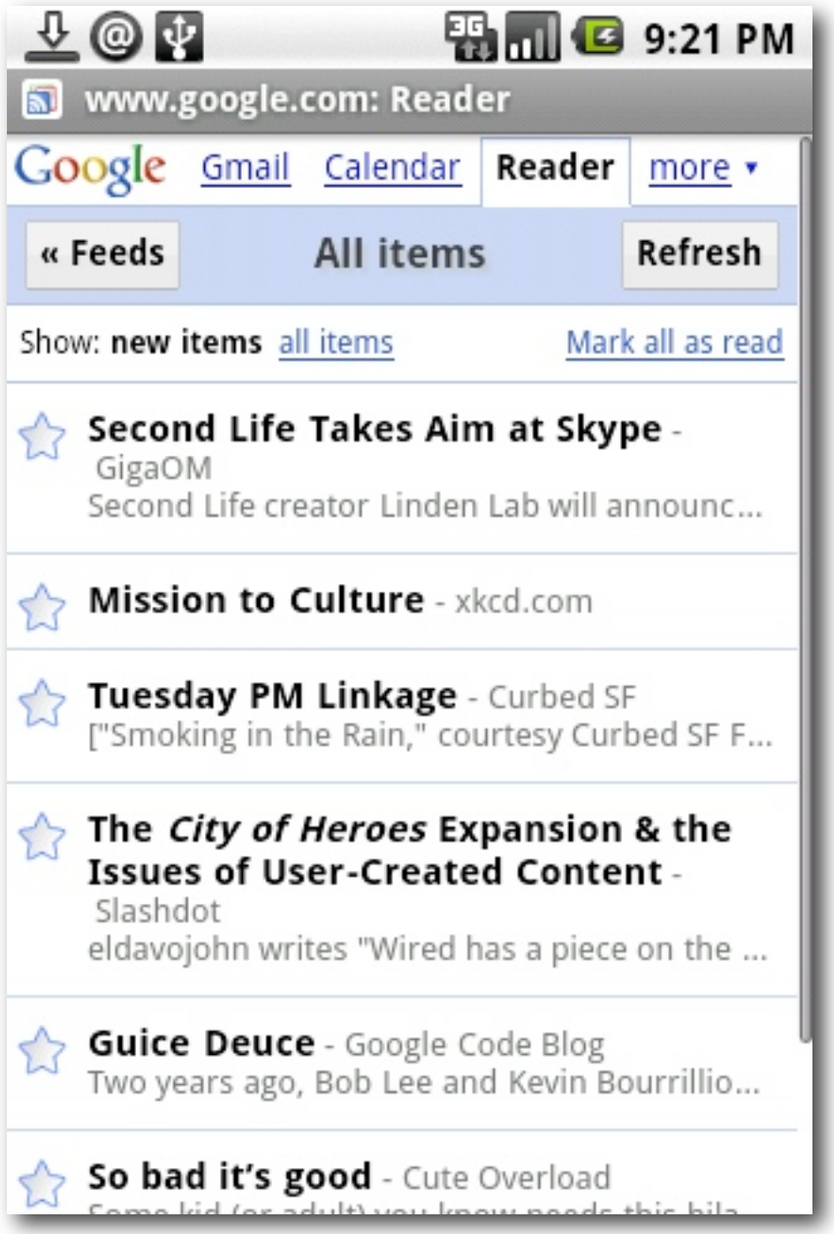
[1] - Hold that thought, though.

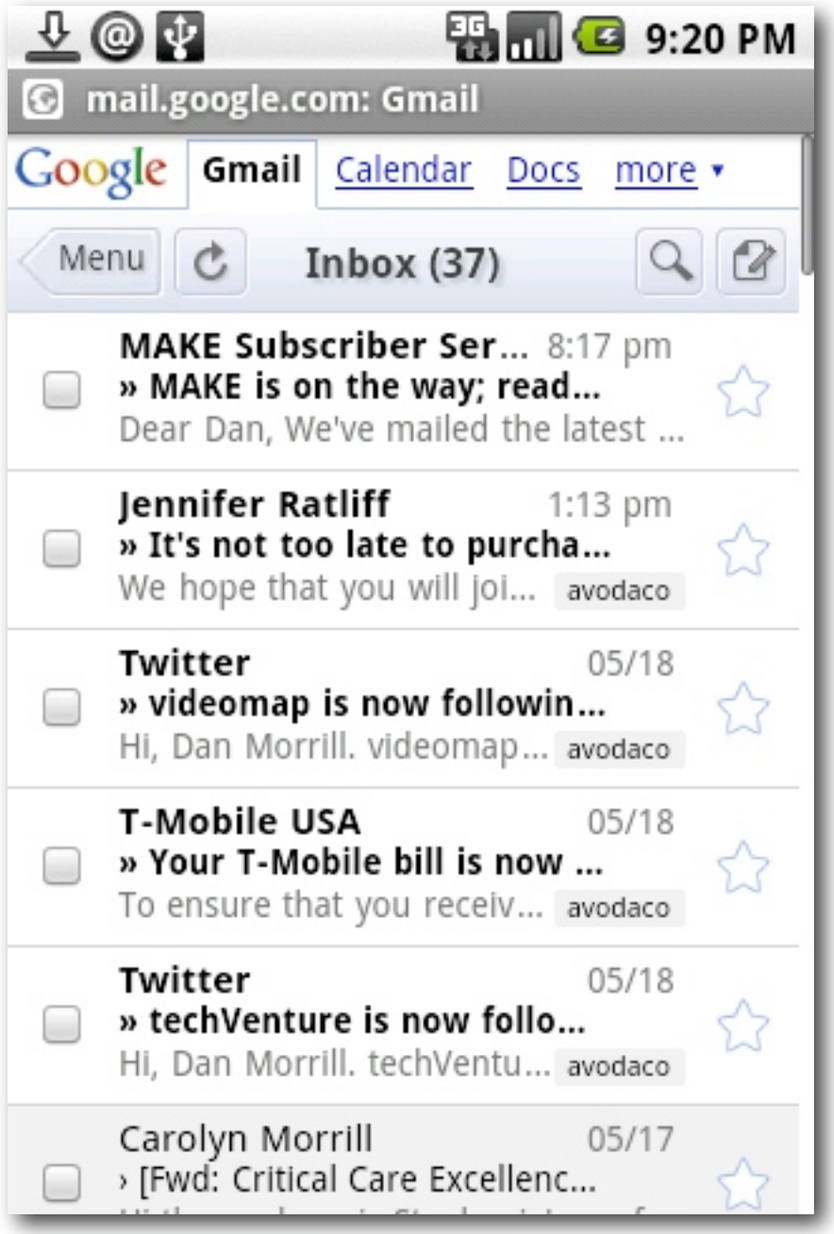
Ajax - Demo

```
var means = new Array();
var distances = {};

function computeDistance(a, b) {
    return Math.sqrt(Math.pow(a.x - b.x, 2) + Math.pow(a.y - b.y, 2));
}

function cluster() {
    var converged = false;
    var dirty = false;
    var distance = 0.0;
    var curMinDistance = 0.0;
    var sumX = new Array();
    var sumY = new Array();
    var clusterSizes = new Array();
    var cluster = new Array();
    var loopCount = 0;

    while (!converged) {
        dirty = false;
        for (var i = 0; i < points.length; i = i + 1) {
            point = points[i];
            curMinDistance = distances[hash(point)];
            for (var j = 0; j < means.length; j = j + 1) {
                mean = means[j];
                distance = computeDistance(point, mean);
                if (distance < curMinDistance) {
                    dirty = true;
                    curMinDistance = distance;
                    point.cluster = j;
                }
            }
        }
        if (!dirty) {
            converged = true;
            break;
        }
        for (var i = 0; i < means.length; i = i + 1) {
            sumX[i] = 0;
            sumY[i] = 0;
            clusterSizes[i] = 0;
        }
        for (var i = 0; i < points.length; i = i + 1) {
            sumX[points.cluster] = sumX[points.cluster] + points.x;
            sumY[points.cluster] = sumY[points.cluster] + points.y;
            clusterSizes[points.cluster] = clusterSizes[points.cluster] + 1;
        }
        for (var i = 0; i < means.length; i = i + 1) {
            means[i].x = sumX[i] / clusterSizes[i];
            means[i].y = sumY[i] / clusterSizes[i];
        }
        loopCount = loopCount + 1;
        if (loopCount > MAX_LOOP_COUNT) {
            converged = true;
        }
    }
}

function hash(booga) {
```

Ajax - Examples

- Google Reader for mobile
- Gmail (web version)
- Many mobile web apps (of various fruit flavors)



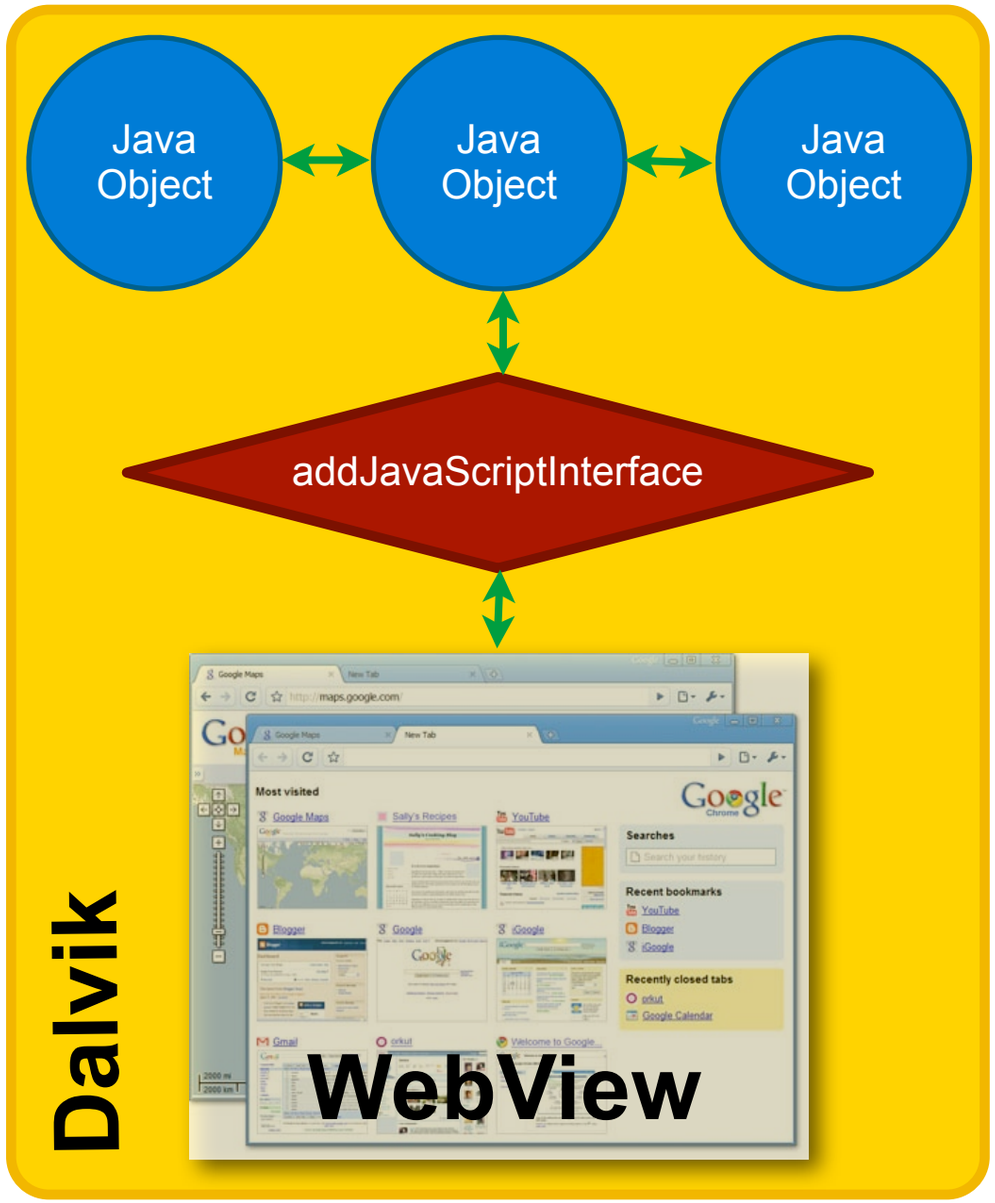


Ajax - Future

- HTML5 will keep getting better
- Android will continue to include a world-class browser
  - Gears today, HTML5 in the future
  - Plans to upgrade to an even faster JSVM
- Who knows what else? I hear the future is a crazy place

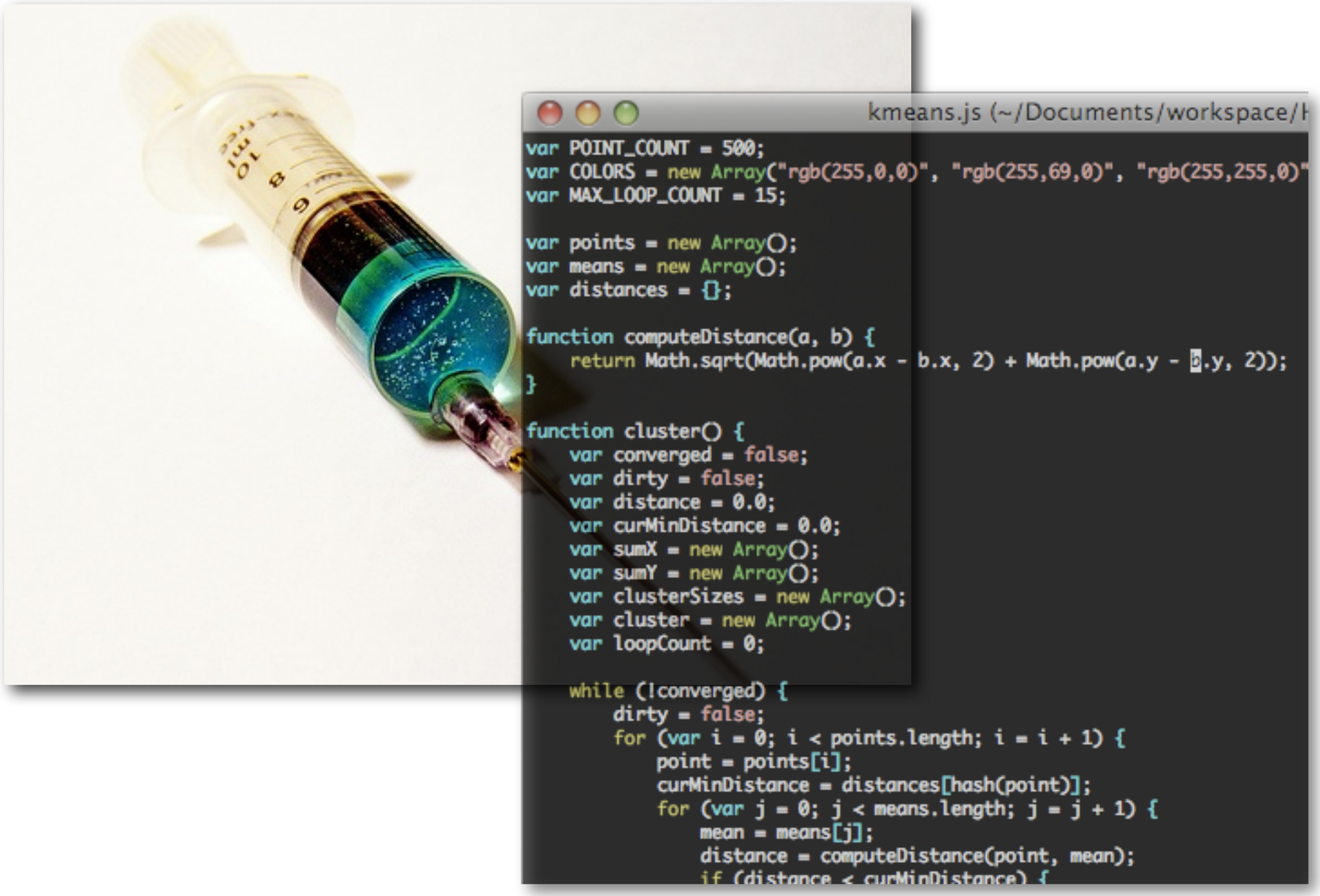
Ajax - What Else Can You Do?

- Augmented Ajax



Augmented Ajax

- Inject functionality into JavaScript



Native Code

Native - Overview

- Launched as a Dalvik application
- Loads a dynamic library (.so, ARM ELF, linked against Bionic)
- Makes calls into native code via JNI for heavy lifting
- .so files are created using the NDK complement to the SDK
- Work-in-progress NDK added to donut branch in git

Native - What Can You Do?

- Physics simulations
- Efficient/fast loading of large-ish data files
- Speed-intensive lookups, such as for IMEs
- Custom VMs (long live obsolete games!)
- Unsupported things...
  - Technically, other libraries are present, but have no guarantees
  - Use them, and you deserve the Market user ratings you will get when your app breaks

Native - What Can’t You Do?

- Current set of APIs is limited to libm and libc
- More will be added over time
- h@x0r the system -- sandboxing still applies

Native - Demo

```c
#include <stdlib.h>

double computeDistance(double aX, double aY, double bX, double bY) {
    return sqrt((aX - bX) * (aX - bX) + (aY - bY) * (aY - bY));
}

typedef struct {
    double x;
    double y;
    int cluster;
} point;

#define MAX_LOOP_COUNT 15;

JNIEXPORT void JNICALL
Java_com_google_io_kmeans_NativeClusterer_cluster(JNIEnv* env, jobject jobj,
jobjectArray points, jint numClusters, jint width, jint height) {
    int converged = 0;
    int dirty = 0;
    double curMinDistance = 0;
    int loopCount = 0;
    jsize numPoints = (*env)->GetArrayLength(env, points);
    double distances[numPoints];
    point means[numClusters];
    double sumX[numClusters];
    double sumY[numClusters];
    int clusterSizes[numClusters];
    double distance = 0;
    int cluster = 0;
    jobject point;
    jclass pointClass;
    jfieldID xFID, yFID, clusterFID;
    jdouble ptX = 0, ptY = 0;
    int i, j;

    // cache some JNI objects so we don't keep requerying them
    pointClass = (*env)->FindClass(env, "com/google/io/kmeans/Clusterer$Point");
    xFID = (*env)->GetFieldID(env, pointClass, "x", "D");
    yFID = (*env)->GetFieldID(env, pointClass, "y", "D");
    clusterFID = (*env)->GetFieldID(env, pointClass, "cluster", "I");

    // initialize the means
    for (i = 0; i < numClusters; ++i) {
        means[i].x = rand() % width;
        means[i].y = rand() % height;
        means[i].cluster = i;
    }

    // initializes the distances record
    for (i = 0; i < numPoints; ++i) {
        distances[i] = 100000000.0; // might break for larger resolutions
    }

    // the main loop
    while (!converged) {
        dirty = 0;
        for (i = 0; i < numPoints; ++i) {
            point = (*env)->GetObjectArrayElement(env, points, i);
            for (j = 0; j < numClusters; ++j) {
```

Native - Examples

- SCUMMVM
- Spotify
- Much of the system, actually...

Native - Future

- Add additional libraries, but which? TBD.
  - Let us know!
- No plans to make native code a fully-independent app model
  - ...but there's no reason to prevent developers from doing stuff

Bonus Demo

How Do You Choose?

Some Data

Modality	Total	Rendering Time	Percent Rendering
Dalvik	614	16	2.6%
JavaScript	936	303	32.4%
Native	127	15	11.2%

Some Data

Modality	Total	Rendering Time	Percent Rendering
Dalvik	614	16	2.6%
JavaScript	936	303	32.4%
Native	127	15	11.2%

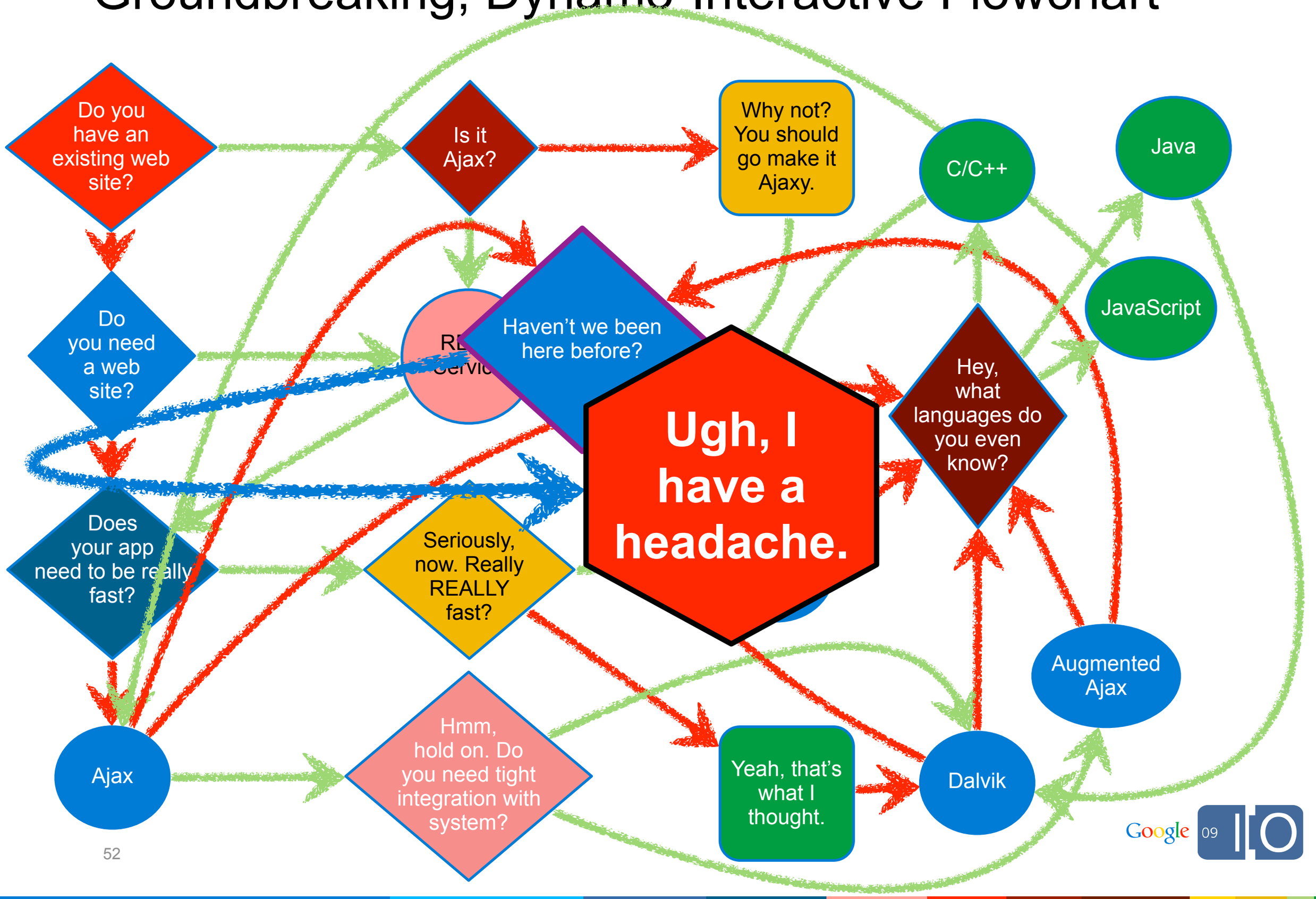
Some Data

Modality	Total	Rendering Time	Percent Rendering
Dalvik	614	16	2.6%
JavaScript	936	303	32.4%
Native	127	15	11.2%

What's that mean?

- Dalvik is very competitive with other pure-interpreted VMs
- Dalvik, as a front-end for native code, usually beats JavaScript
  - But clearly, there are use cases where that doesn't matter
- And of course, custom native code will always win the speed test

Groundbreaking, Dynamo-Interactive Flowchart



The Bottom Line

- There’s more than one way to do it
- We aren’t out to shove a paradigm down your throat (ouch?)

The Bottom Line

- The “centroid” of your app can be anywhere
  - Dalvik apps can embed the web; web apps can call to Dalvik
  - Apps can include native code pieces for better performance
- Understand your app, & decide based on your app’s needs
  - If you’re doing lots of drawing, use Dalvik
  - If you have a web presence already, use (Dalvik-enhanced?) Ajax
  - If you have large, slow algorithms, use Dalvik + native code
  - Your skills do mean something: go with what you know!

And That's Pretty Much The Long and Short of It

Source available at: http://code.google.com/p/hdict

Google™ 09 I/O