# C++ in the Post-PC Era

email@DiegoDagum.com
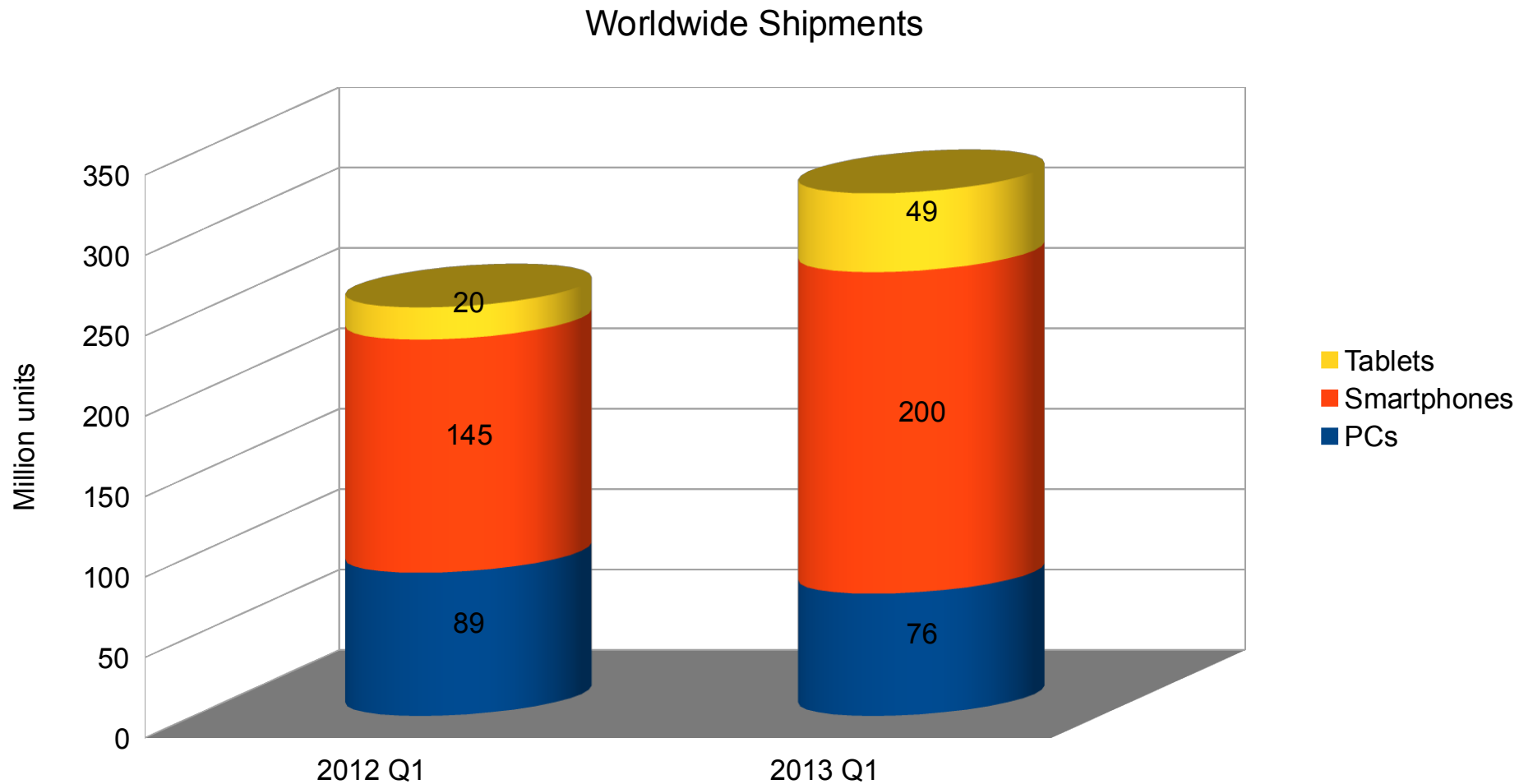Software Architect
*and Post-PC C++ Activist*
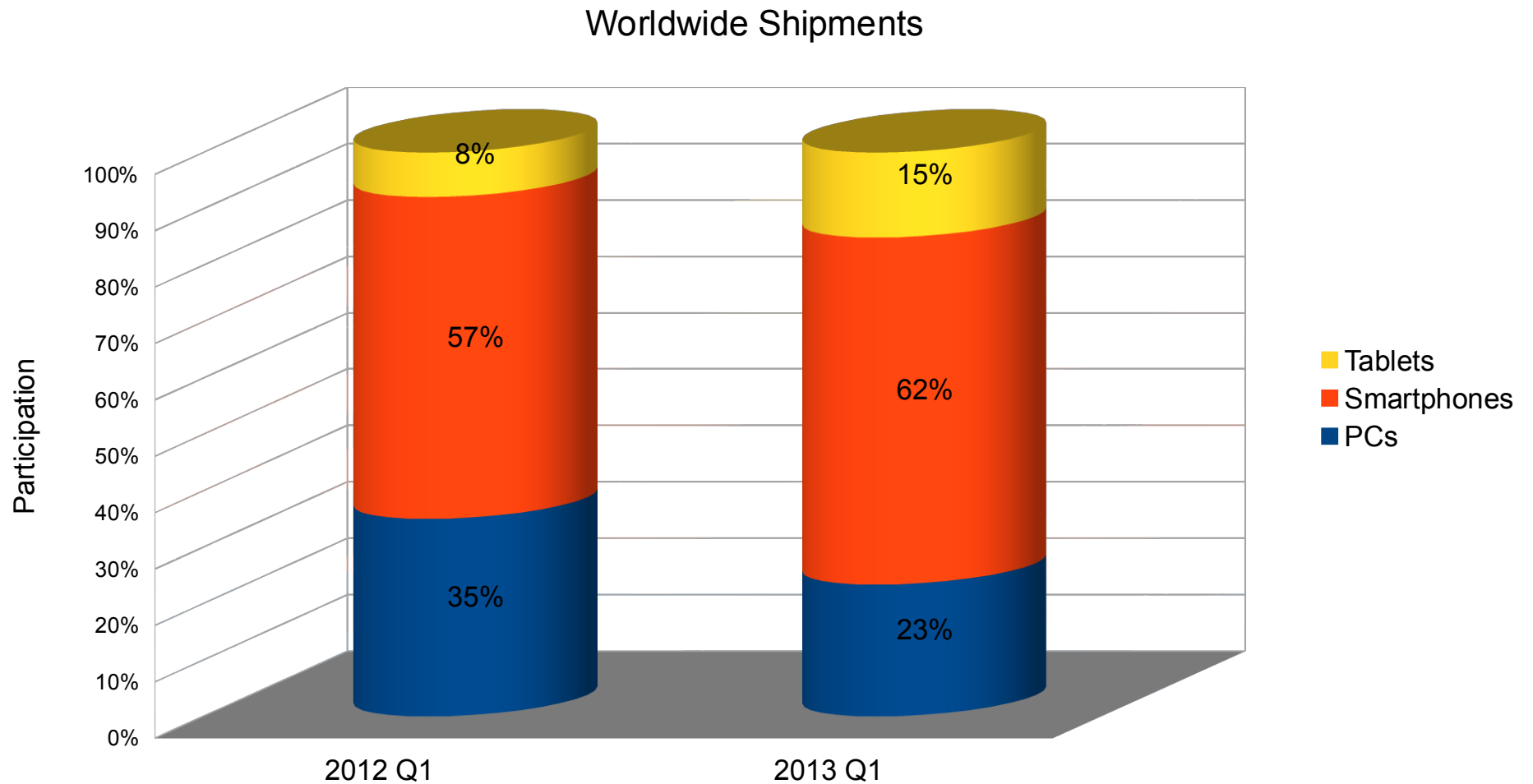
# What We Shall Discuss Tonight

- The post-PC era has arrived
- C++ at the leading post-PC platforms
- `post_pc_apps != pc_apps`
- What's wrong today with C++
- *FULLfilling* the C++ promise
  (full as IN FULL, not as in FOOL)

# A *Techtonic* Shift

## Worldwide Shipments

Million units

- 350
- 300
- 250
- 200
- 150
- 100
- 50
- 0

**2012 Q1**
- Tablets: 20
- Smartphones: 145
- PCs: 89

**2013 Q1**
- Tablets: 49
- Smartphones: 200
- PCs: 76

Legend:
- Tablets
- Smartphones
- PCs

*Sources: IDC, Juniper Research*

# A *Techtonic* Shift (cont.)

# A *Techtonic* Shift (cont.)

Tablet shipments:

http://www.idc.com/getdoc.jsp?containerId=prUS24093213

PCs: http://www.idc.com/getdoc.jsp?containerId=prUS24065413

Smartphones:

http://www.juniperresearch.com/viewpressrelease.php?pr=374

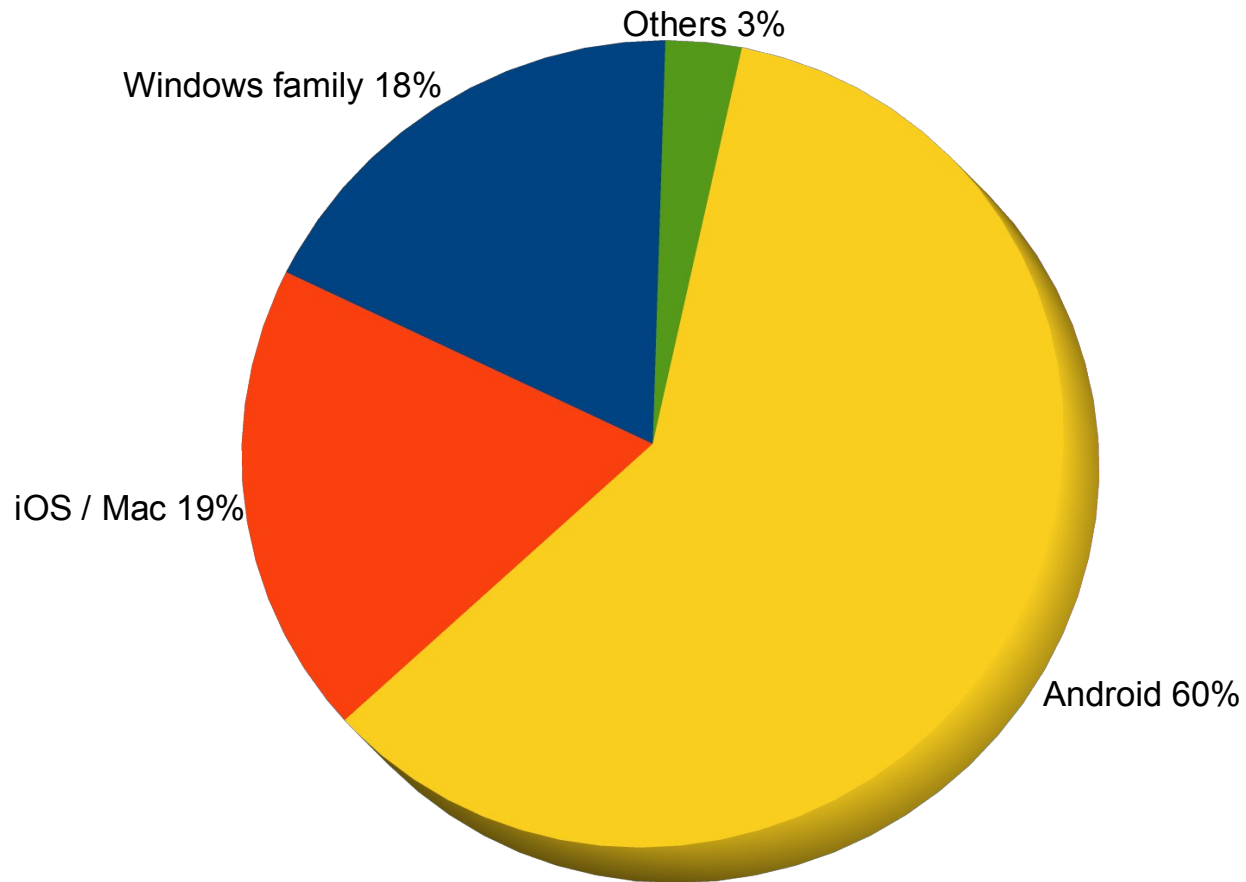http://www.idc.com/getdoc.jsp?containerId=prUS23455612

- Smartphones overtook PCs in 2011

# A *Techtonic* Shift (cont.)

- Tablets to overtake PCs this year or next

- Tablets to overtake smartphones in 3 years

- Android and iOS 91% smartphone share
  http://www.idc.com/getdoc.jsp?containerId=prUS23946013

- Android and iOS 92% tablet share
  http://www.strategyanalytics.com/default.aspx?mod=pressreleaseviewer&a

- Android 59% of PCs+tablets+smartphones
  http://www.zdnet.com/canalys-android-powers-59-percent-of-smartphones

# A *Techtonic* Shift (cont.)

Platform Participation (Tablets, smartphones and PCs)



Others 3%

Windows family 18%

iOS / Mac 19%

Android 60%

*Source: Canalys*

# C++ in iOS

# Cocoa, Cocoa Touch ~~and C++~~

| | |
|---|---|
| **Cocoa Touch** | UIKit |
| **Media** | Core Graphics<br>Open GL ES<br>Core Animation |
| **Core Services** | Core Data<br>Foundation |
| **Core OS** | |
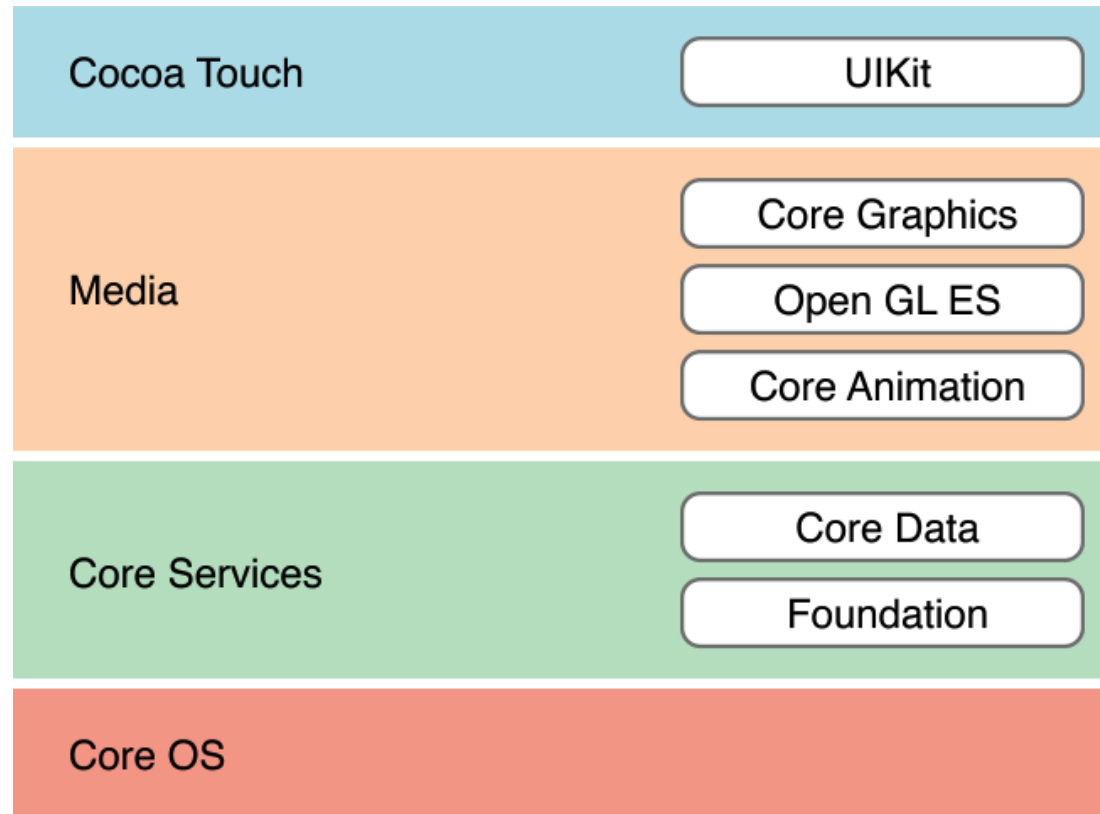
**Built on Objective-C**

Much of Cocoa Touch is implemented in Objective-C, an object-oriented language that is compiled to run at incredible speed, yet employs a truly dynamic runtime making it uniquely flexible. Because Objective-C is a superset of C, it is easy to mix C and even C++ into your Cocoa Touch applications.

# What's Implied in "*and even C++*"



About Paul | Newsletter |

REGISTER | LOG IN

Paul Thurrott's
SUPERSITE FOR WINDOWS

Windows    Mobile    Cloud    Windows Server    Office    Music & Videos

Spin up a WordPress
site in seconds.
Once you get it, you'll get it.

Windows Azure
FREE 90-DAY TRIAL

HOME > SOFTWARE DEVELOPMENT > NEW TILE-BASED SHELL, APP MODEL, AND APP STORE COMING IN WINDOWS 8?

**New Tile-Based Shell, App Model, and App Store Coming in Windows 8?**

Paul Thurrott | Paul Thurrott's Supersite for Windows          Jan. 5, 2011

EMAIL  SHARE  Twe  +1  Recommen          COMMENTS 3

**SuperSite for Windows Community**

Sign up for the WinInfo Daily UPDATE newsletter.

email address    sign up

I agree to terms of use &

**Windows 8 Rumor**
New tile-based shell, app model, app store

Advertisement

I don't normally like to publish rumors that I can't verify with at least a second source, but based on the corroborating evidence I received along with this tip, I do believe it's genuine. If not, then it's an internal attempt to get me to look silly publicly. :)

With that bit of morality out of the way, here goes.

**Windows 8 will include a new tile-based user interface that's codenamed Mosh.** Assuming this is true, I have to believe that this UI will be an alternative UI, and not a full replacement, or will appear only on low-end tablet-like devices aimed at the iPad. It sounds like something that will silence the critics who want the Windows Phone OS on a tablet.

**IT/Dev Connections**

Vegas
ember 30th - October 4th

You'll have the opportunity to experience:
• 120 Technical Sessions
• Networking with Peers
xpert Speakers

**Windows 8 will also include a new app model codenamed Jupiter that will target a new Windows Marketplace app store.** The app store will provide access to new, Silverlight based "immersive" applications that are deployed as AppX packages (.appx). The Windows and Office teams are betting very heavily on this new app type, according to my source, and development has already begun using a beta version of Visual Studio 2012. These apps can be written in C#, Visual Basic, and even C++.

# Errata

- Where it says "*and even C++*", it should say "*uneven C++*"

  - *Uneven* when compared to other included languages in the same platform (segmentation)

  - *Uneven* compared to other platforms C++ approaches (fragmentation)

```
// TODO: answer the question
// How much portable are C++ apps?
```

# post_pc_apps != pc_apps

# `post_pc_apps != pc_apps`

- Post-PC apps are very good at a couple of things. They don't try to be anything.

- Most successful ones feature NUI (natural user interfaces)

  – Touch, accelerometer

  – Camera

- Augmented reality

  – Camera

  – Sensors: Location, magnetometer

# `post_pc_apps != pc_apps` (cont.)

- Graphics (cross-cutting)
  - GPU

- C++ needs low-level access to the key APIs that make post-PC apps unique

- Otherwise, its participation in post-PC apps becomes irrelevant

# `post_pc_apps != pc_apps` (cont.)

- Multiplicity
  - Installed base version
  - Devices
- (Automated) testing in actual devices, not emulators
- Discoverability (aka "Marketing")
- Monetization strategy
- … Ah! Last but not least, time-to-market

# Supporting Multiple iOS Versions

iOS Version Share in iPhone

iOS 4.x 1% iOS 5.x 7%

iOS 6.x 92%

iOS Version Share in iPad

iOS 4.x 2%

iOS 5.x 15%

iOS 6.x 83%

*Source: http://david-smith.org/iosversionstats/*
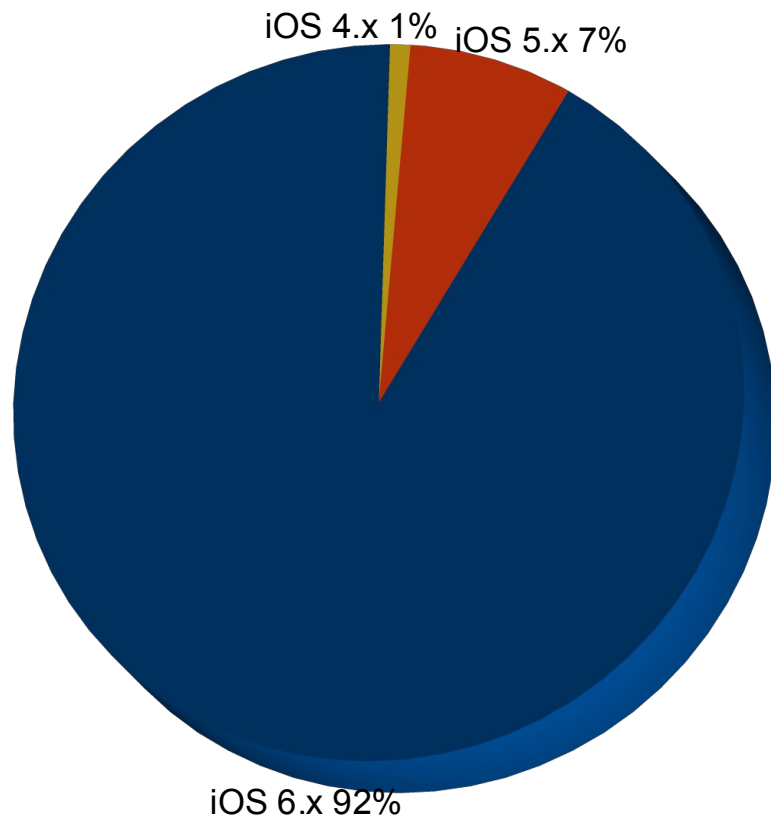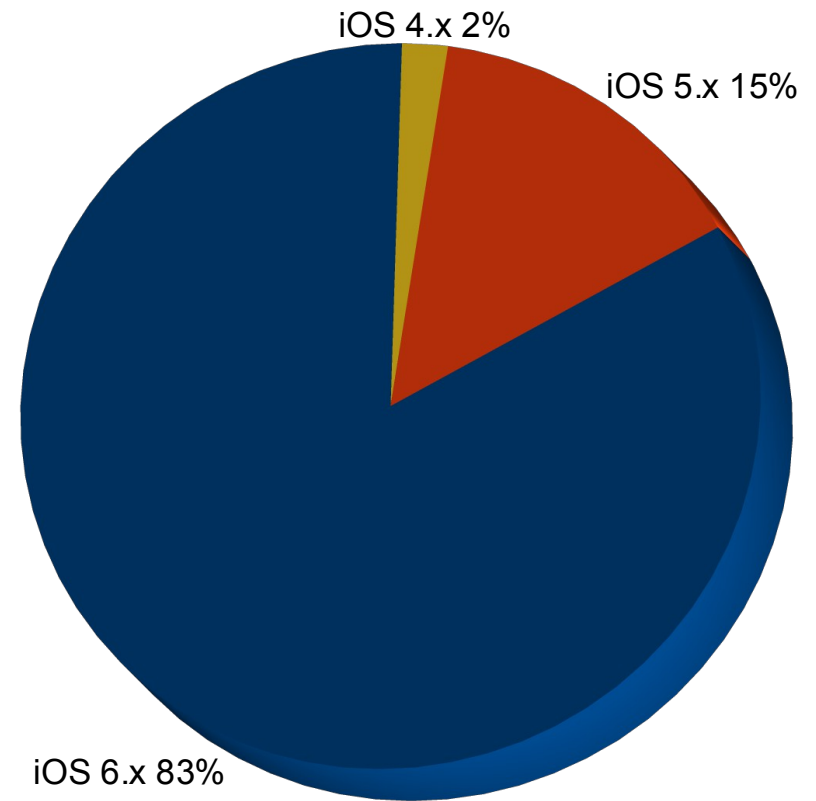
# The App Store Opportunity
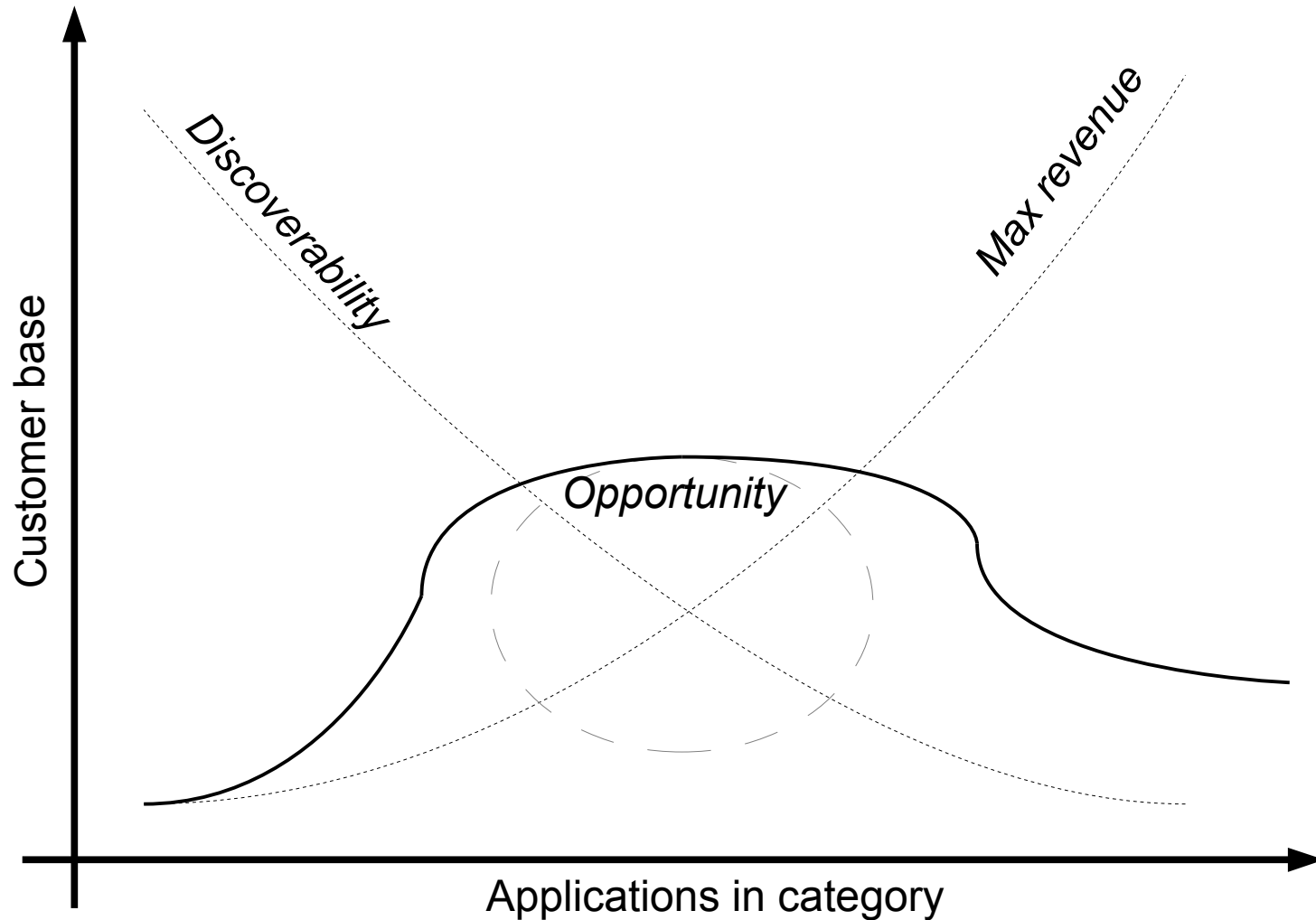
How to make money with post-PC applications?
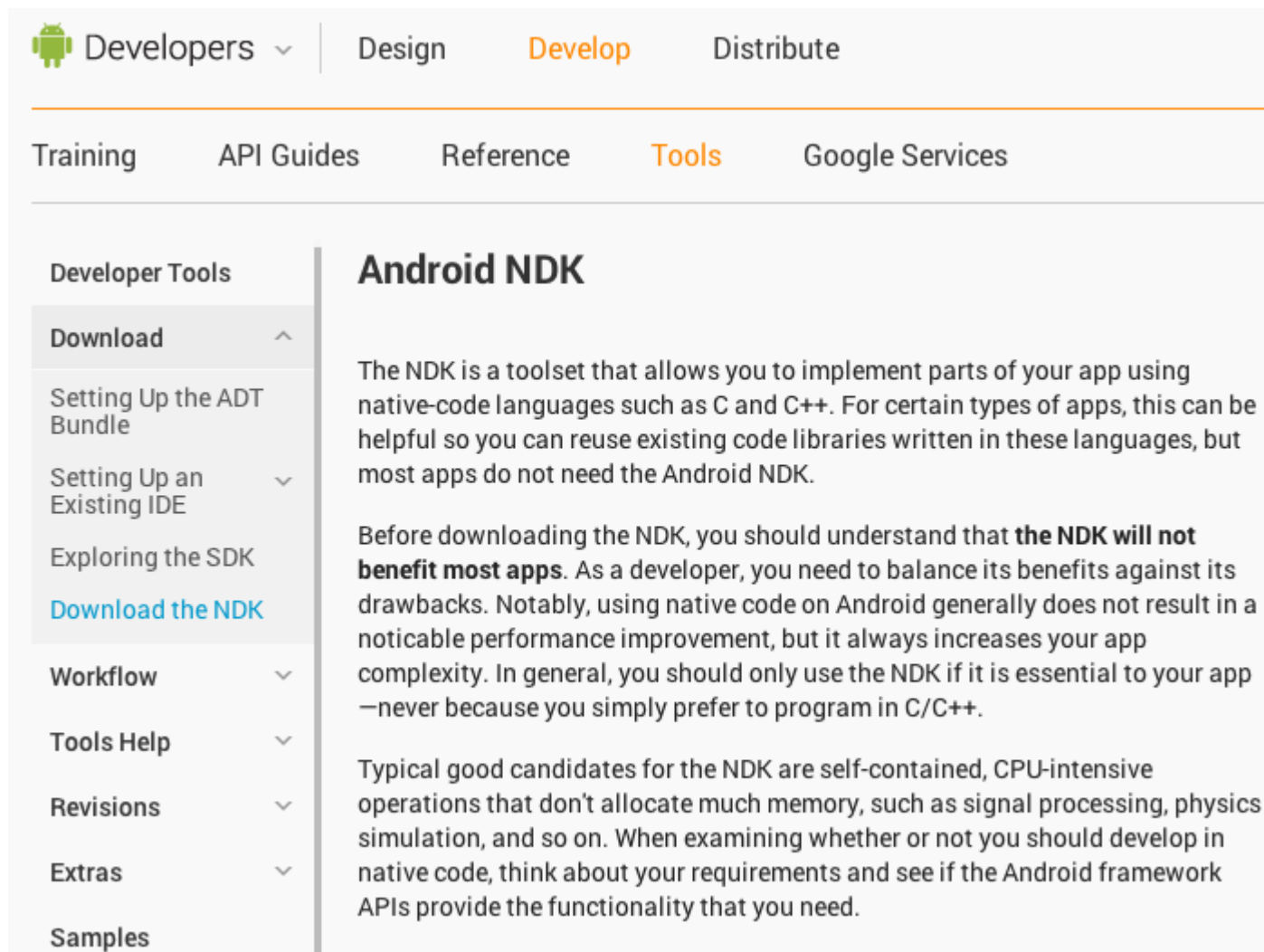
A: Direct sale

B: Free with ads

C: "Freemium"

D: B & C combined

# C++ in Android



Developers ⌄  Design  **Develop**  Distribute

Training   API Guides   Reference   **Tools**   Google Services

Developer Tools

Download ⌃

Setting Up the ADT Bundle

Setting Up an Existing IDE ⌄

Exploring the SDK

Download the NDK

Workflow ⌄

Tools Help ⌄

Revisions ⌄

Extras ⌄

Samples

## Android NDK

The NDK is a toolset that allows you to implement parts of your app using native-code languages such as C and C++. For certain types of apps, this can be helpful so you can reuse existing code libraries written in these languages, but most apps do not need the Android NDK.

Before downloading the NDK, you should understand that **the NDK will not benefit most apps**. As a developer, you need to balance its benefits against its drawbacks. Notably, using native code on Android generally does not result in a noticable performance improvement, but it always increases your app complexity. In general, you should only use the NDK if it is essential to your app —never because you simply prefer to program in C/C++.

Typical good candidates for the NDK are self-contained, CPU-intensive operations that don't allocate much memory, such as signal processing, physics simulation, and so on. When examining whether or not you should develop in native code, think about your requirements and see if the Android framework APIs provide the functionality that you need.

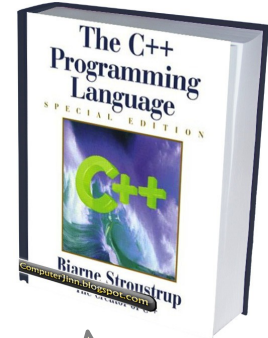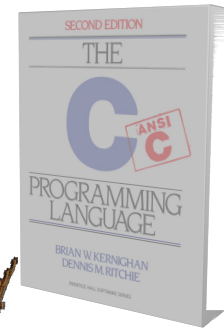# C++ in Android (cont.)

- However...
  - Emphatically YES, YOU CAN USE C++
  - Entire apps in C++ without any line of Java
    - Native APIs for sensors, activity lifecycle, etc.
    - OpenGL ES 1.1, 2.0
  - C++11: g++ by default, Clang bundled as well
    - Optional STL, exceptions

# C++ in Android (cont.)

# Android Dashboard

Platform Version

Jelly Bean 4.2.x 2%   Earlier 6%

Jelly Bean 4.1.x 26%

Gingerbread 2.3.3+ 38%

Ice Cream Sandwich 4.0.x 28%
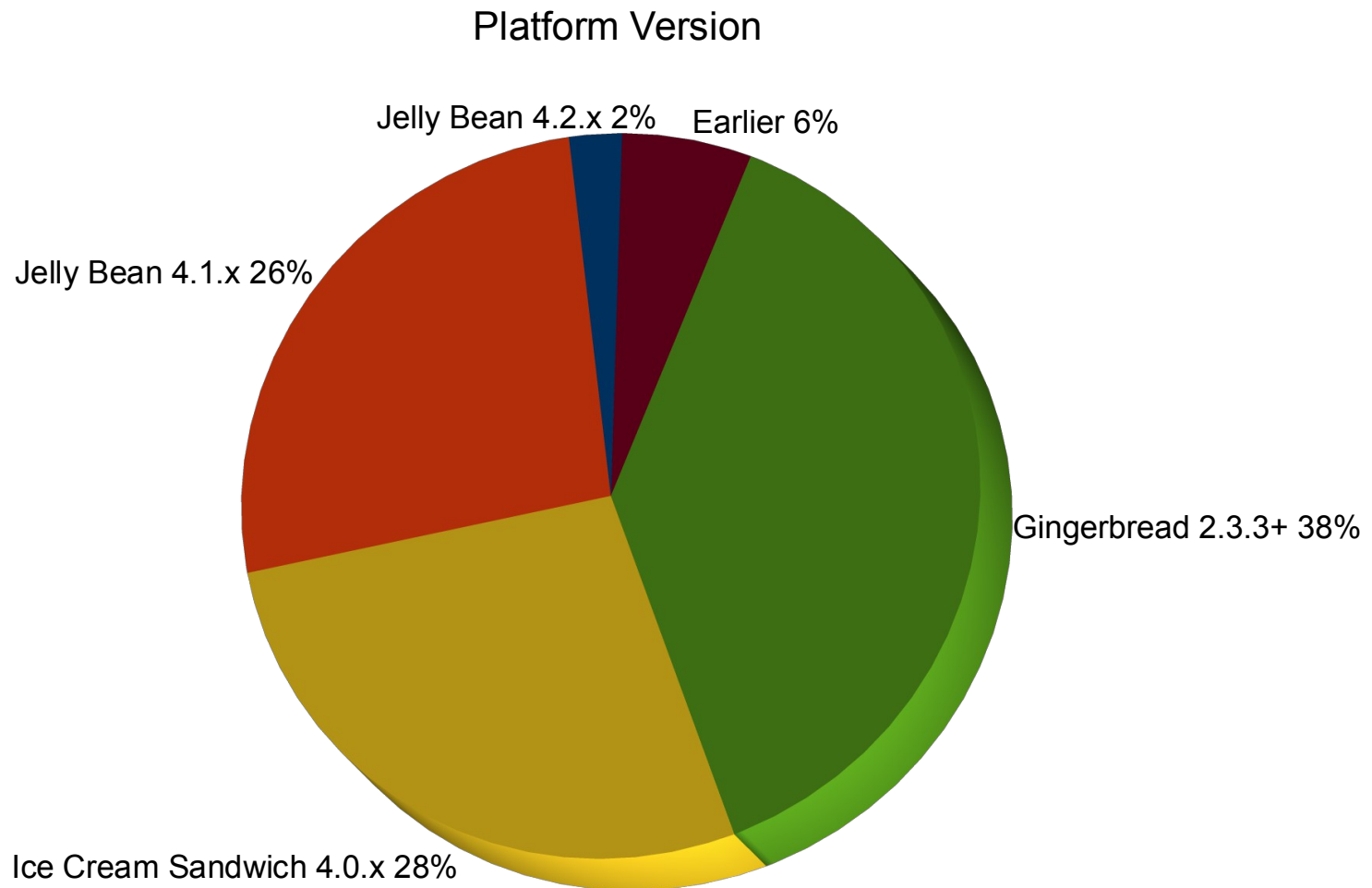
*Source: developer.android.com*

# ISO C++ Growing Technical Debt
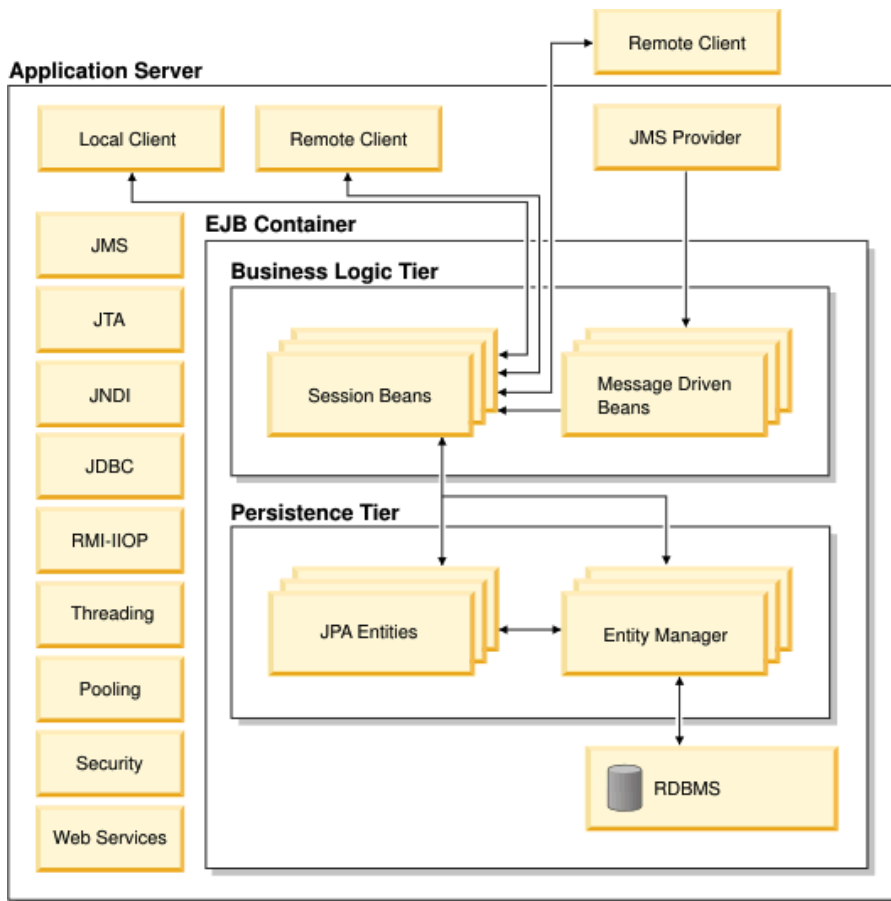
# Measuring Portability of C++ Apps

- % of ISO C++ code

- % of bridging code

- % of non-ISO C++

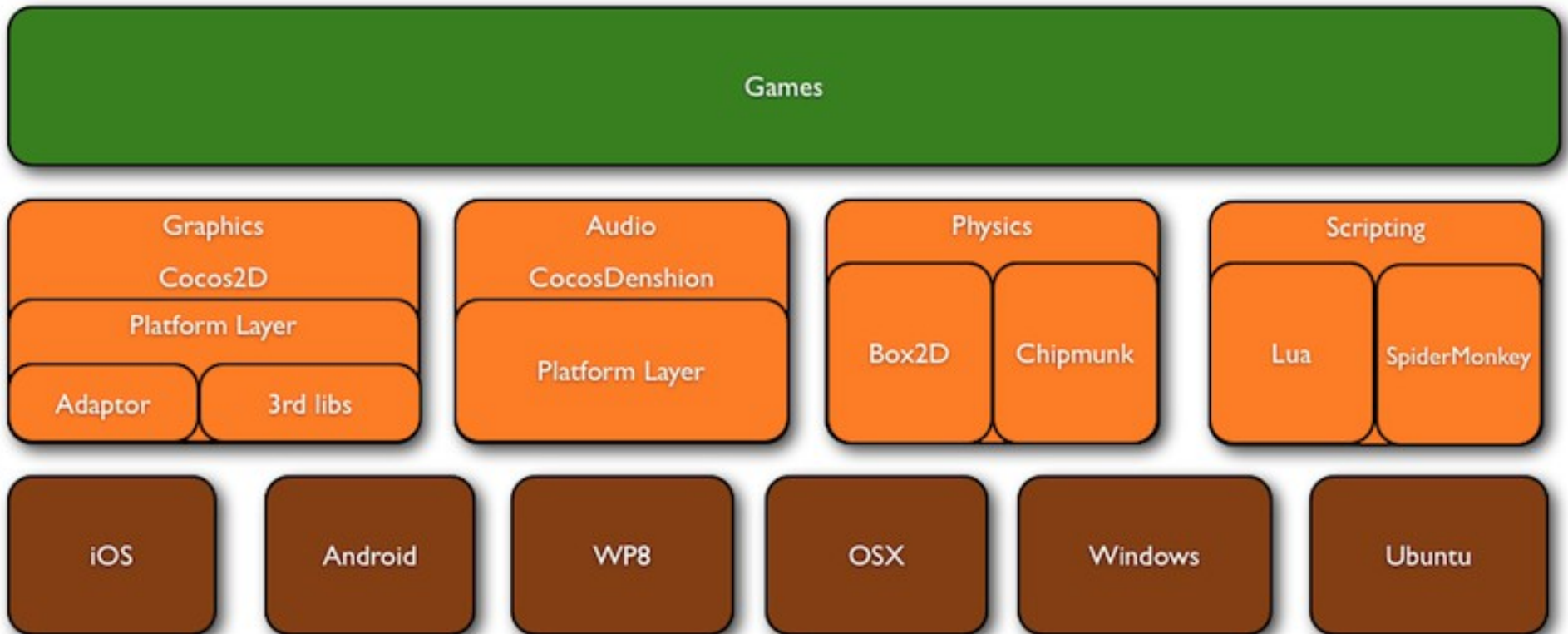The wider this area, the "*more portable*" the application
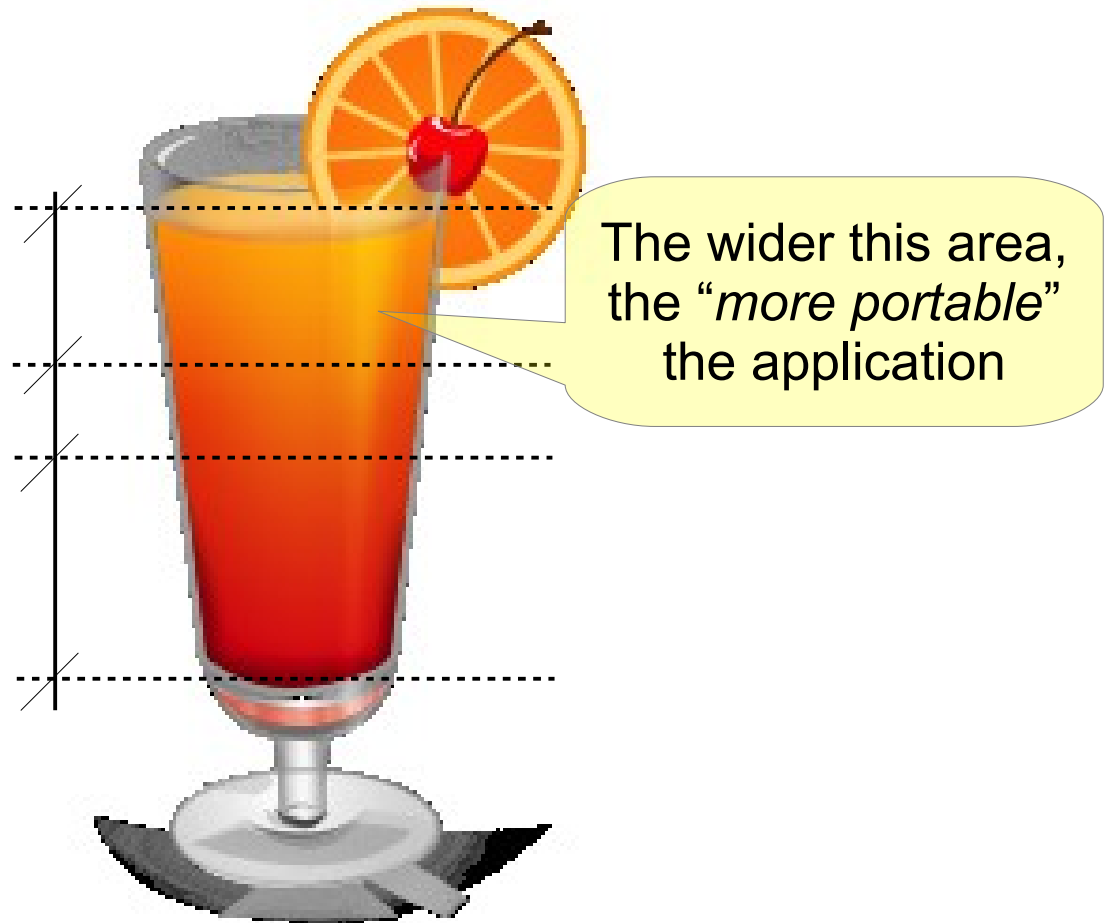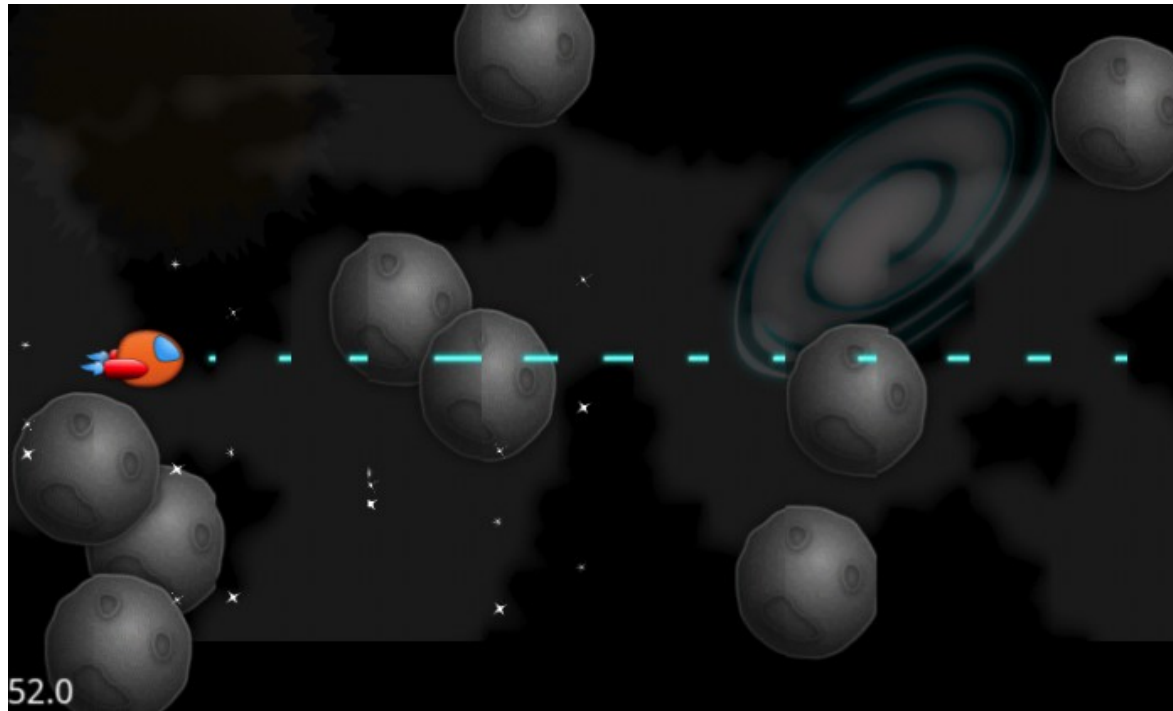
# *FULLfilling* the C++ Promise

- Cocos2d-x

- Marmalade

- Qt

- MoSync

- … any other?

# www.Cocos2d-x.org

## Cocos2d-x Architecture

# Cocos2d-x Demo



- *Credits to Jean-Yves Mengant (experienced Android / IOS developer and designer).*

www.madewith**Marmalade**.com

- Arguably the most robust

- Complete API lineup including sensor support

- Paid, not free

# www.Qt-project.org

- Founded by Nokia, yielded to Digia

- Primary targets are PC devices

  - Including Macs, Linux

- Support for iOS, Android coming up

- Default dev framework in Blackberry 10

# www.MoSync.com
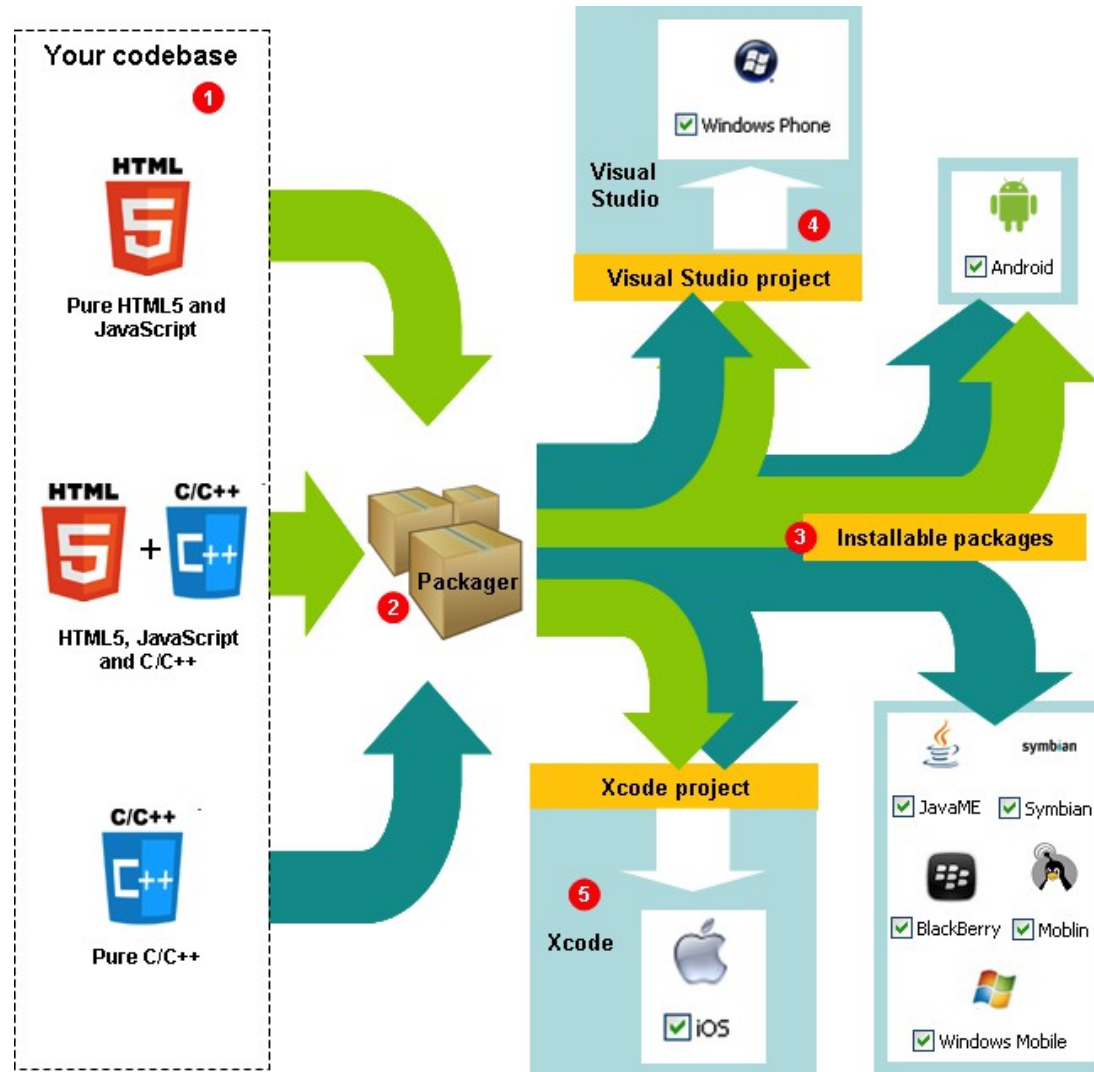
# OpenGL ES

- Framework for 3D rendering

- Not exclusively a C/C++ API (Java, Obj-C...)

- OpenGL ES vX is decimated cut of OpenGL vX

- OpenGL ES 2.0 said more complex than 1.0

- No backward compatibility from 2.0 wrt 1.0

- Supported in both, Android and iOS

# Conclusions

- You can use C++ in post-PC applications
- Dev story inferior compared to PCs
    - Platform integration isn't uniform
    - Difficult to get C++ solutions end-to-end
- ISO C++ API coverage too low
    - Vendors fill spots with their own solutions
    - Portability of modern apps easily eroded
- Emerging cross-platform, community frameworks complement the standard

# Thank You!!

email@DiegoDagum.com
@diedagum
www.linkedin.com/in/diegum