EMBEDDED SYSTEMS PROGRAMMING 2016-17

Introduction to the Platforms

LIBRARY

A collection of resources

(in our context, **classes**)

used to develop software

- Examples: the Java Class Library, the C++ Standard Template Library (STL), the C standard library...

FRAMEWORK

A **collection of libraries**
organized to provide a key functionality

A framework suggests/imposes a programming model

- Example: Apple's UIKit framework provides the classes needed to construct and manage an application's user interface for iOS

PLATFORM

A **collection of software frameworks**
(including application frameworks)
that allows software to run

A platform specifies an operating system, a set of programming languages and run-time system libraries. It may include an hardware architecture

- Examples: Android, iOS, Windows Phone

ECOSYSTEM

ECOSYSTEM

ECOSYSTEM

A **platform together with** the **community** developing **hardware and software products** for that platform

The products and the users are sometimes considered part of the ecosystem as well

- Example: Android together with Android developers, Google Play, etc.

ANDROID, IOS, WP

They share several characteristics.

- They are platforms

- They include an **operating system**, a set of **core applications** (browser, e-mail, ...) and a rich set of **libraries** to develop custom applications

- Libraries for data storage, hardware access, multimedia, 3D graphics... (More on this later)

- They are optimized for embedded devices

OPERATING SYSTEM (1/2)

RESEARCH, OPENNESS AND REGULATIONS

- Thompson and Ritchie invented Unix in the 1960s as a **research project** while at Bell Labs, a telecom company

- Under a 1958 antitrust decree, Bell Labs could not sell non-telecom technology: it was required to **license Unix to anyone who asked**

- Unix was made available to universities and firms, under licenses that included **all source code**: this fostered experimentation, innovation, adoption
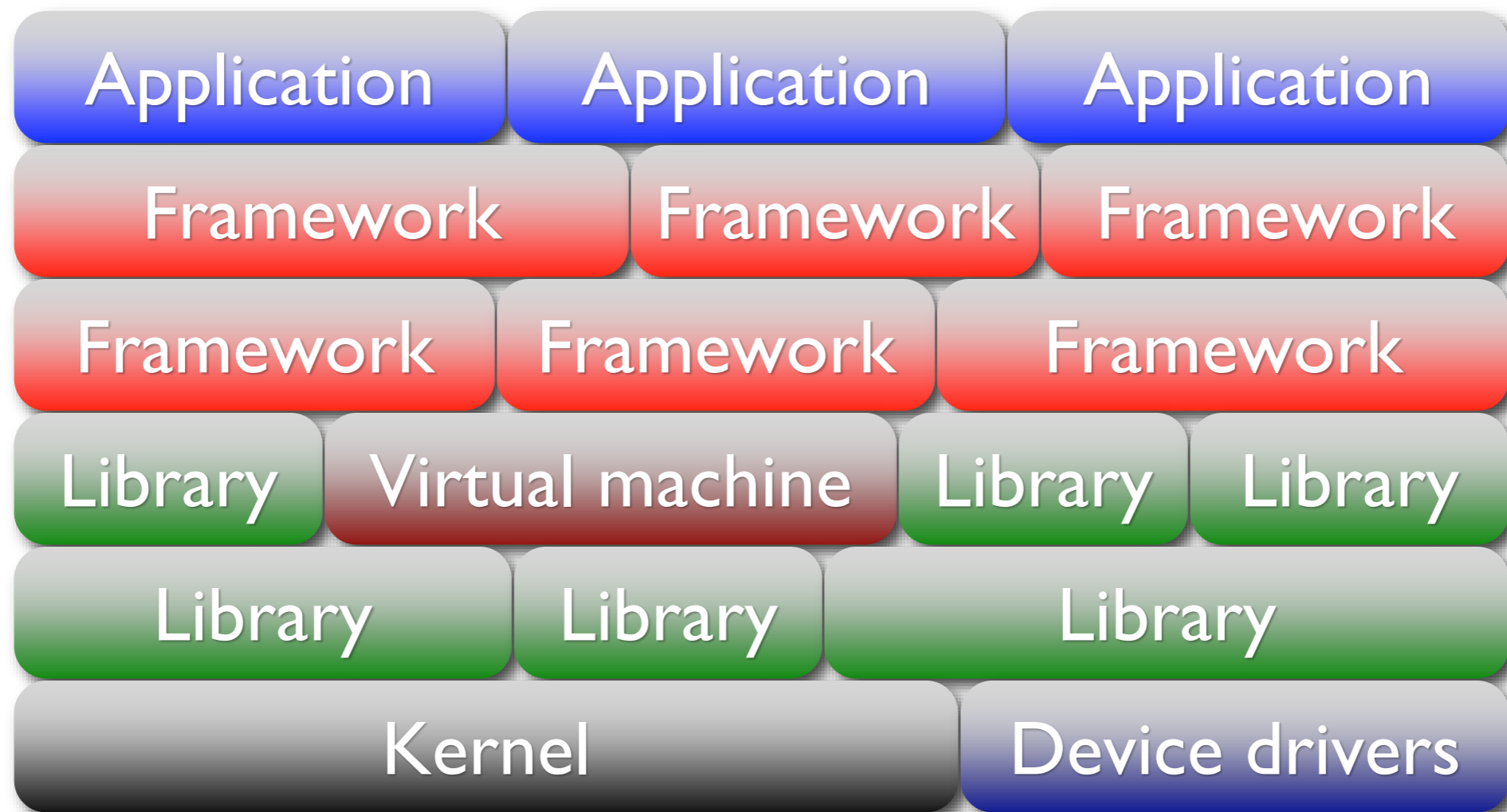
THE ROLE OF UNIX

OPERATING SYSTEM (2/2)

- **Android**: Linux (intemperately customized)

- **iOS**: BSD Unix (heavily modified)

- **Windows Phone**: based on proprietary, closed-source kernels (WinCE for WP7, Win8 for WP8)

ON TOP OF THE OS

Application	Application	Application
Framework	Framework	Framework
Framework	Framework	Framework
Library	Virtual machine	Library	Library
Library	Library	Library
Kernel	Device drivers

- There might be different libraries for the same function, at different levels of abstraction

TYPICAL LIBRARIES AND FRAMEWORKS

- For fonts, 2D and 3D graphic rendering

- For user interface (UI) management

- For network management

- For data storage

- For decoding/encoding multimedia formats

- For geolocation

TYPICAL APPLICATIONS

- Web browser, email, chat

- Calendar / appointments / todo manager

- Contacts / phonebook

- Media (music, video, ...) player

- Application market

LICENSES

- **Android**: open source*,
  download sources from http://source.android.com/

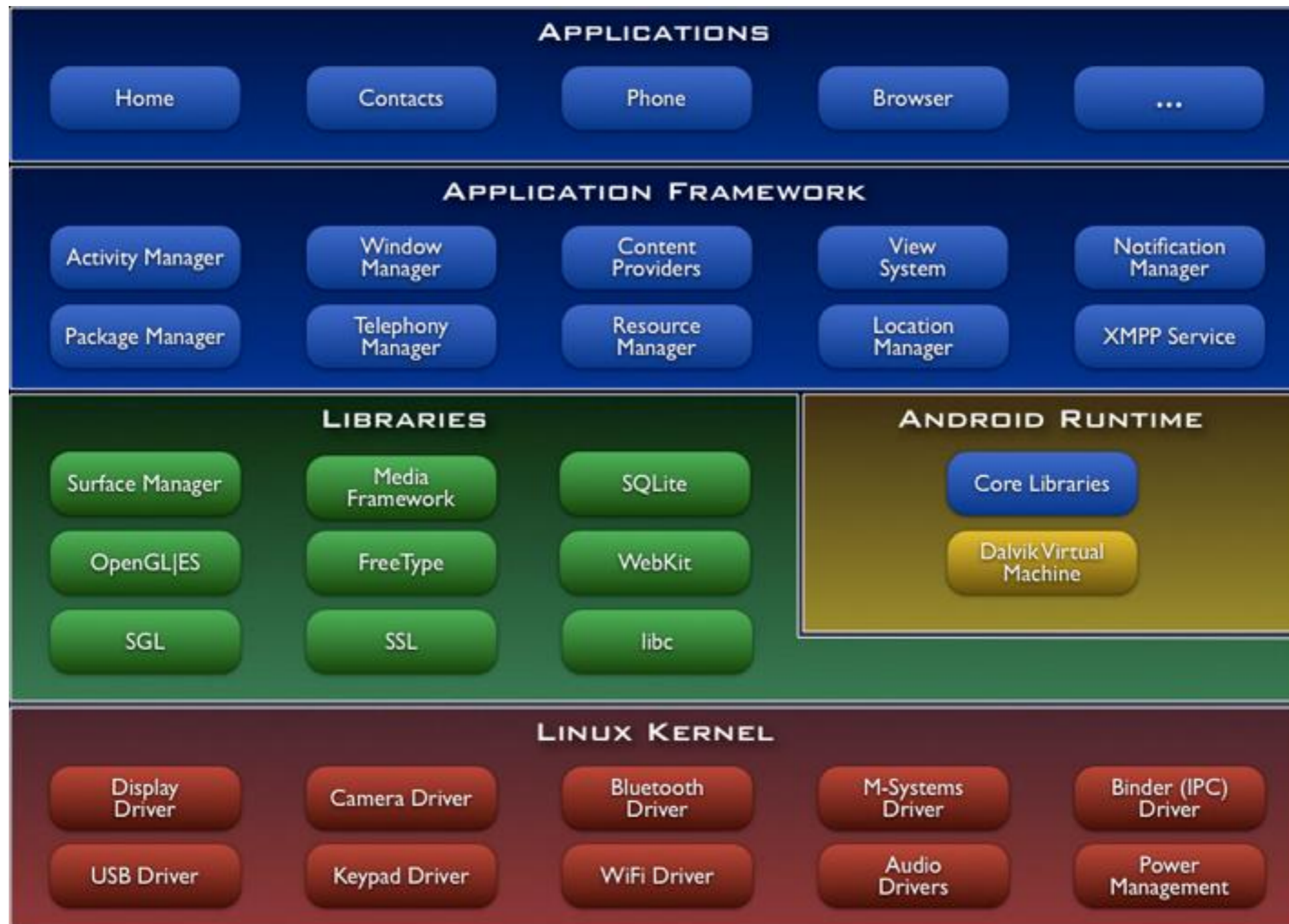- **iOS**: closed source, copyright aggressively enforced

- **Windows Phone**: closed source

ANDROID: HISTORY

- 2003: Android Inc. founded

- 2005: Android acquired by Google

- 2008: first software release (open source)

- 2008: first product (HTC Dream)

ANDROID: ARCHITECTURE

ANDROID: LIBRARIES (1/2)

- **libc**: BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices

- **SSL**: Secure Socket abstraction based on the SSL protocol

- **SGL**: 2D graphics engine

- **WebKit**: web browser engine which powers both the Android browser and embeddable web views

- **FreeType**: bitmap and vector font rendering

ANDROID: LIBRARIES (2/2)

- **OpenGL|ES**: 3D graphics engine; uses hardware acceleration where available

- **SQLite**: open-source, lightweight relational database engine

- **Media Framework**: playback and recording of MPEG4, H.264, MP3, AAC, AMR, JPG, PNG... files

- **Surface Manager**: manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications

APPLICATION FRAMEWORK (1/2)

- **View System**: provides building blocks for UI components

- **Window Manager**: creates windows, dispatch UI events to applications

- **Activity Manager**: manage the lifecycle and stacking of applications

- **Content Providers**: store and retrieve data and make it shareable between applications

- **Package Manager**: handles information on the application packages currently installed on a device

APPLICATION FRAMEWORK (2/2)

- **Resource Manager**: handles access to resources inside packages

- **Telephony Manager**: provides access to the telephony services on a device

- **Location Manager**: provides access to GPS and other location services

- **Notification Manager**: collects events happening in the background and notifies them to the user

ANDROID RUNTIME

- Android applications are **developed in Java**, albeit with a custom library (no SE or ME compliance)

- Java sources are compiled into Java bytecode, and then into a **proprietary format** (DEX)

- DEX files are platform independent and are executed into a **proprietary virtual machine** (Dalvik or ART)

PERFORMANCE (1/3)

- DEX, Dalvik and ART are designed for systems constrained in terms of memory and processor speed

- DEX packs multiple classes into a single file

- Dalvik and ART are register-based machines

  - Smaller bytecode

  - Faster execution

PERFORMANCE (2/3)

- Android 2.2: Just-In-Time (JIT) compiler added to Dalvik. It translates bytecode into machine code at run-time

- Android 4.4: ART introduced. Faster, more predictable garbage collector. **Ahead-Of-Time (AOT) compiler**: it translates bytecode into native assembly code at install time

- Android 7.0: AOT compiler complemented by JIT compiler with code profiling

- **Native Development Kit** (NDK) available to compile performance-critical code from C++ into native code
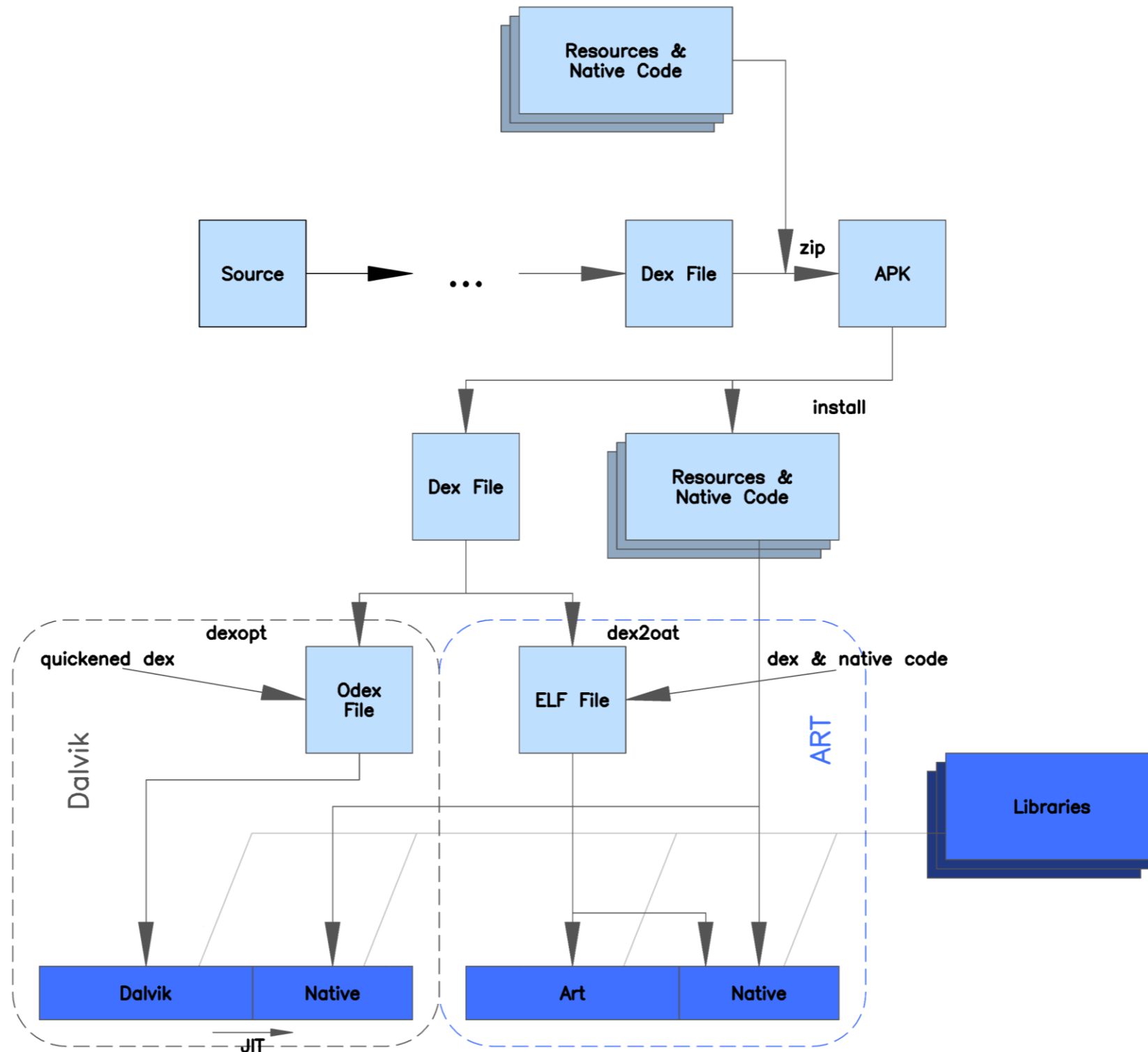
OTHER PLATFORMS

IOS: HISTORY

- 1985: NeXT founded

- 1988: Objective-C licensed

- 1989: NeXTSTEP o.s.

- 1996: Apple acquires NeXT

- 2001: Mac OS X, based on NeXTSTEP via OpenStep

- 2007: iPhone OS (later: iOS), based on OS X

- 2008: iPhone OS SDK, App Store

IOS: ARCHITECTURE
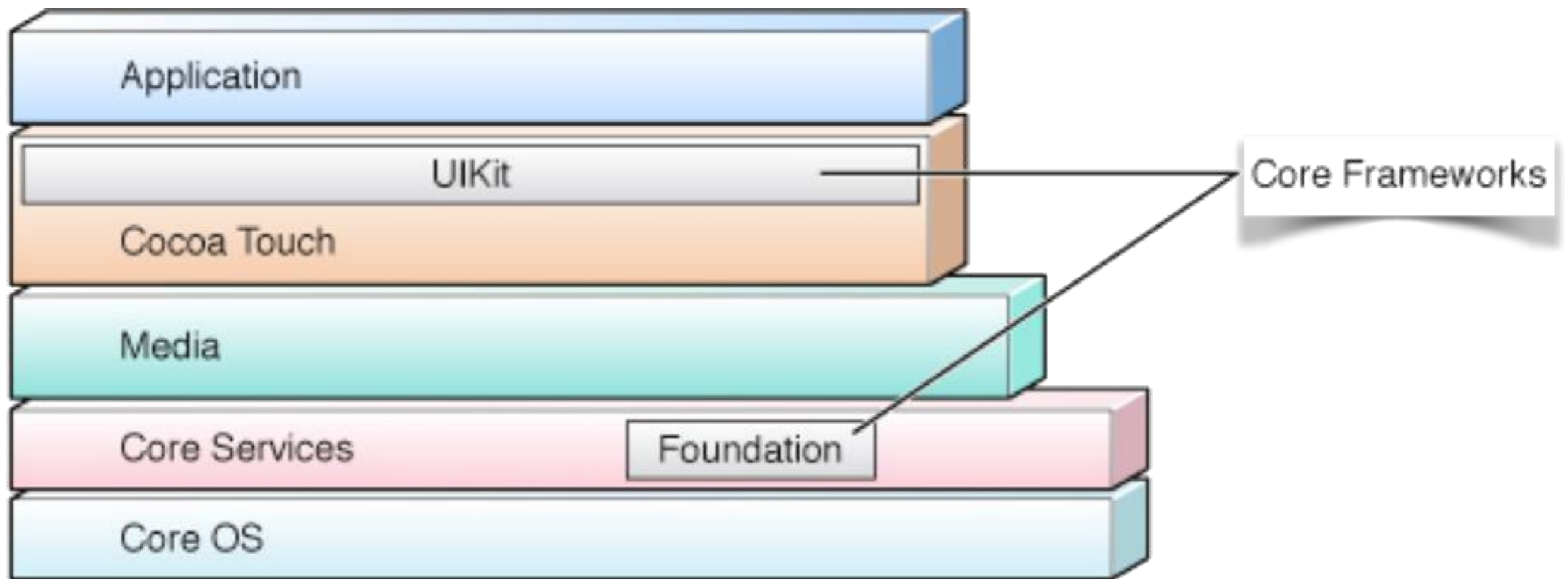
WP: HISTORY

- 1996: Windows CE 1.0, targeted at "handheld PCs"

- 1997: Windows CE 2.0.
  UI even more similar to that of Windows 95/98.
  First keyboardless devices ("palm-size PCs")

- 2000: Windows CE 3.0 / Pocket PC 2000.
  Different versions for different devices

- 2003-2010: versions proliferating for marketing reasons

- 2010: Windows Phone 7, based on Windows CE. Metro UI

- 2012: Windows Phone 8, incompatible with WP7.
  Replaces the Windows CE architecture with one
  based on Windows 8 components & the Windows NT Kernel
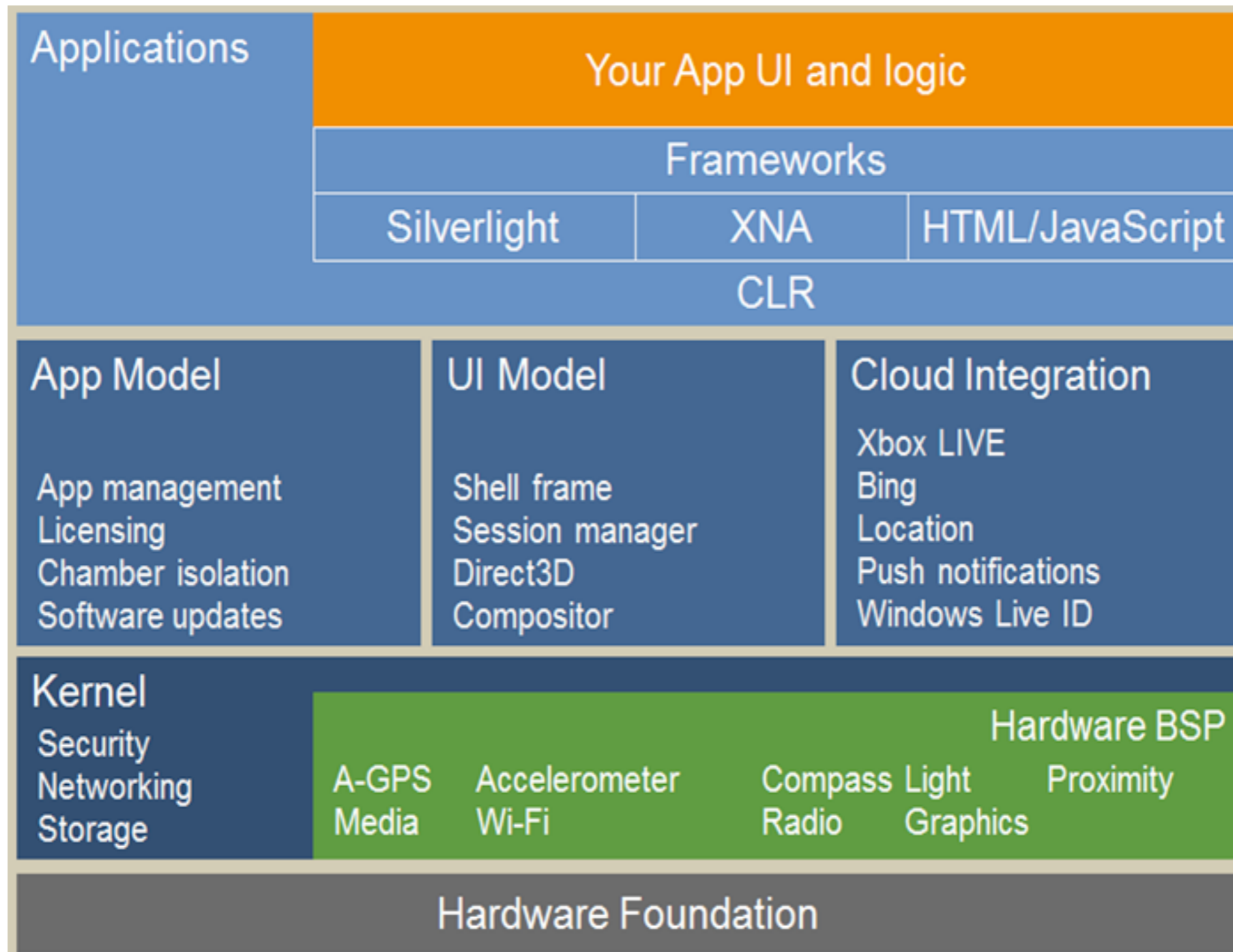
- 2015: Windows 10 Mobile
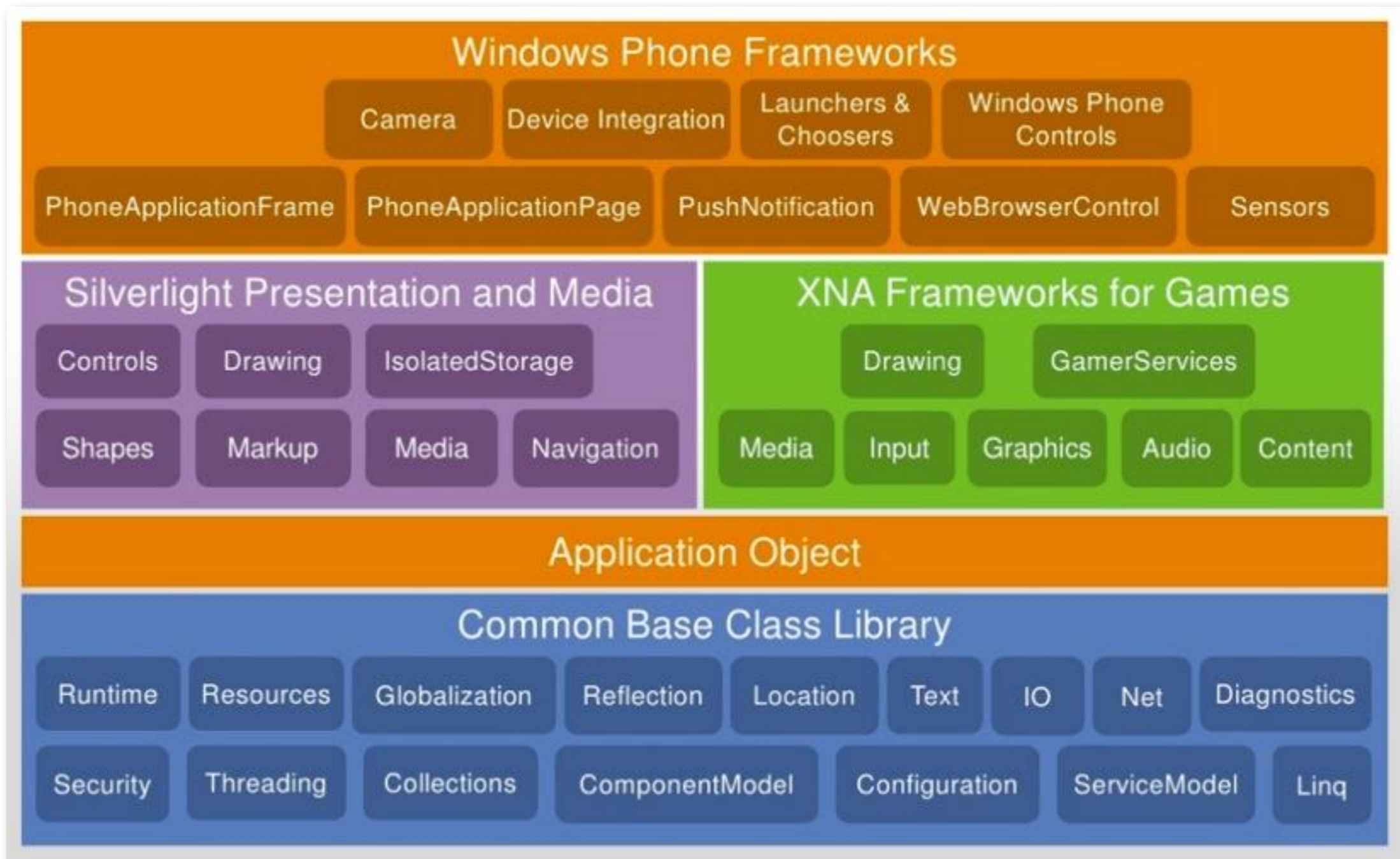
THE CONVERGENCE JOURNEY

Source: Microsoft TechNet
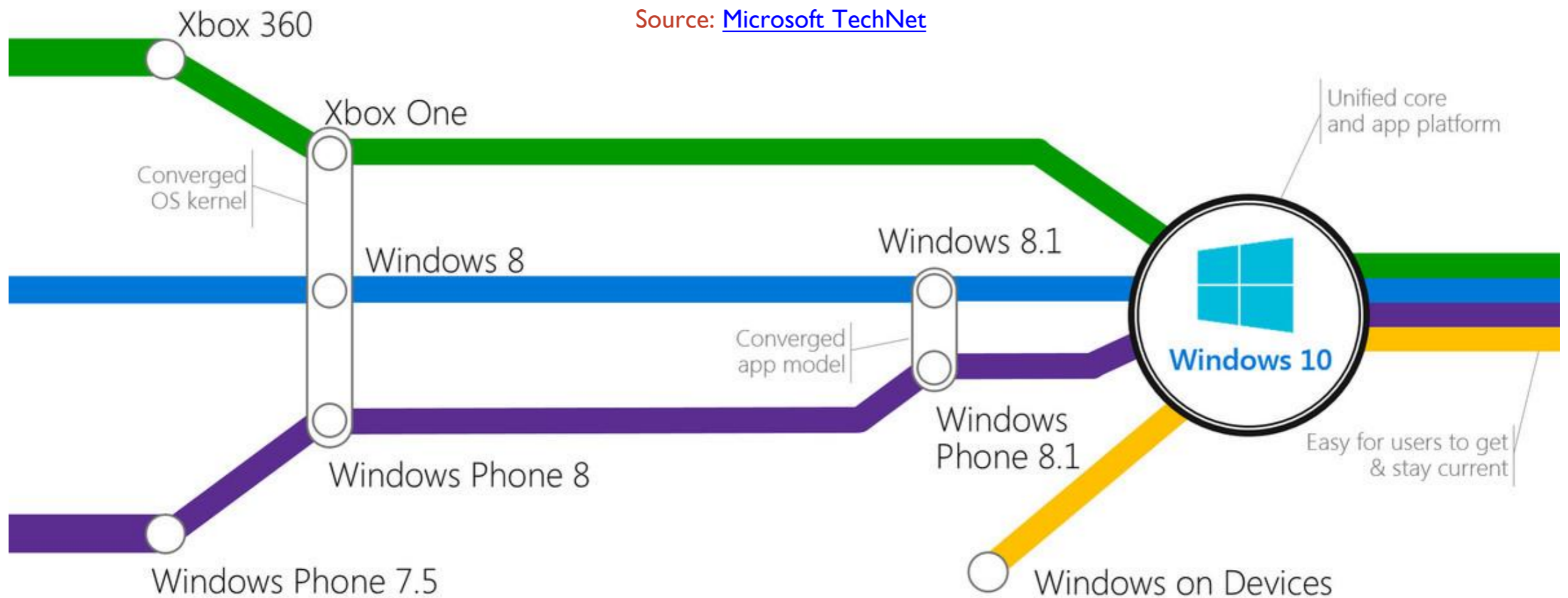
Windows 10: the Kernel and fundamental pieces of the OS (the "core") are common for all platforms. Also, one single binary works across all platforms

WINDOWS PHONE 7.X: SOFTWARE ARCHITECTURE

WINDOWS PHONE 7.X: FRAMEWORK DETAILS

DEVELOPING APPLICATIONS

- <u>Apple</u>: "creating applications that do something useful and look nice requires you to spend some time"

- <u>Microsoft</u>: "It is really easy to get started and become comfortable with the development of applications"

Who to believe?
You decide

TO LEARN MORE

- https://developer.android.com/

- https://developer.apple.com/ios/

- https://dev.windows.com/