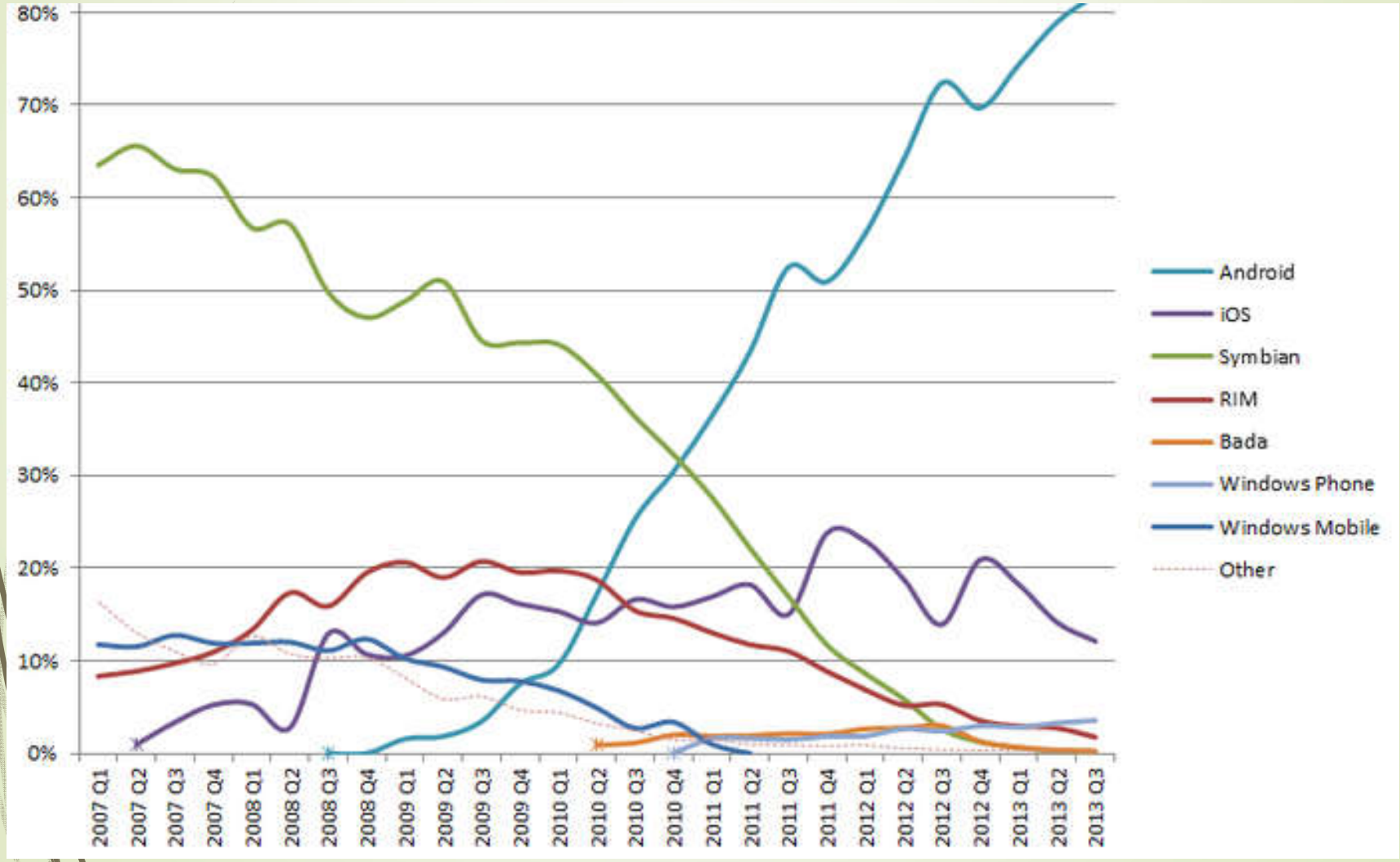
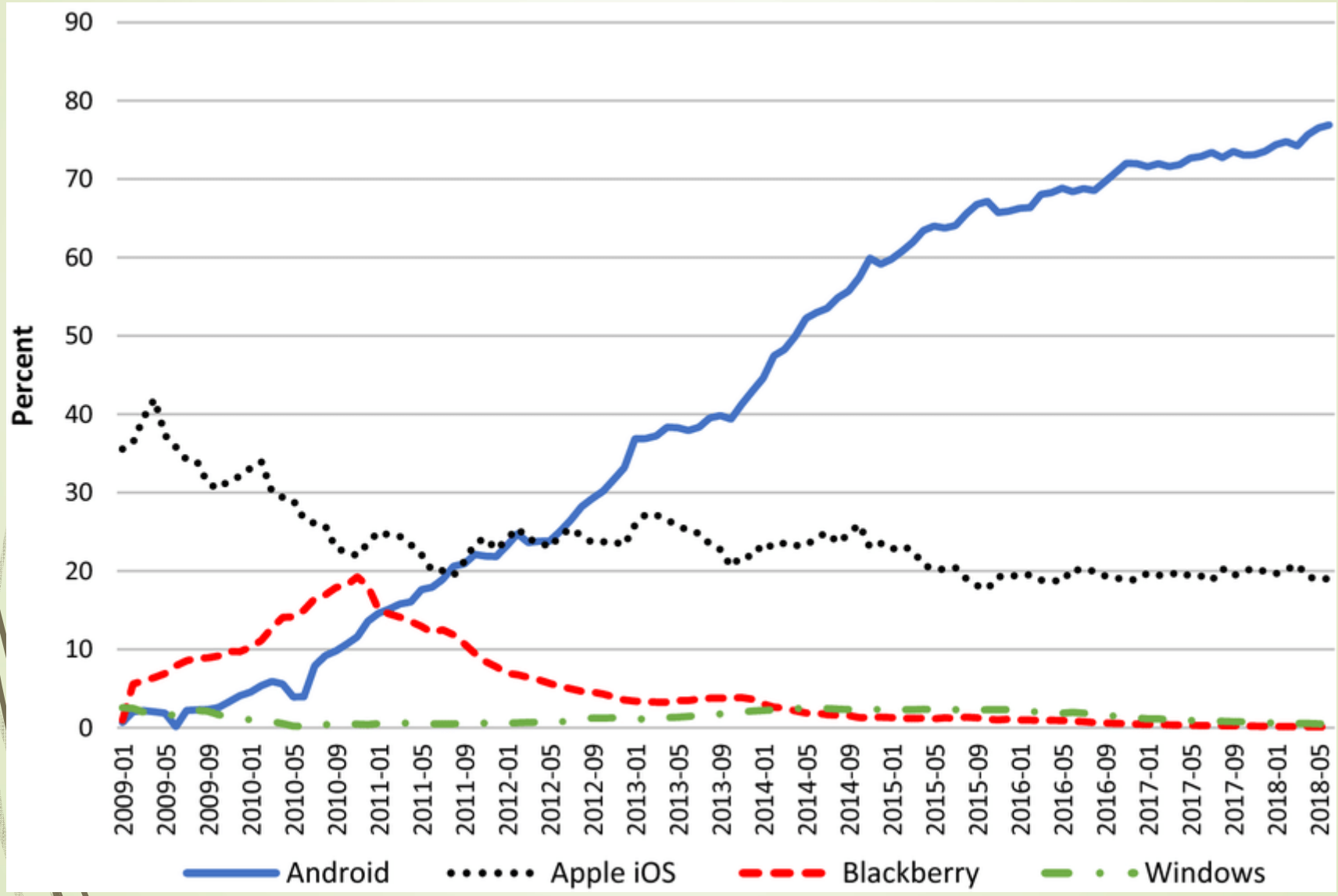
X. Mobile Operating Systems

Prof. Tsvetozar Georgiev
University of Ruse

Popularity of mobile operating systems 2007-2013



Popularity of mobile operating systems 2009-2018



1. Symbian

1.1. Characteristics

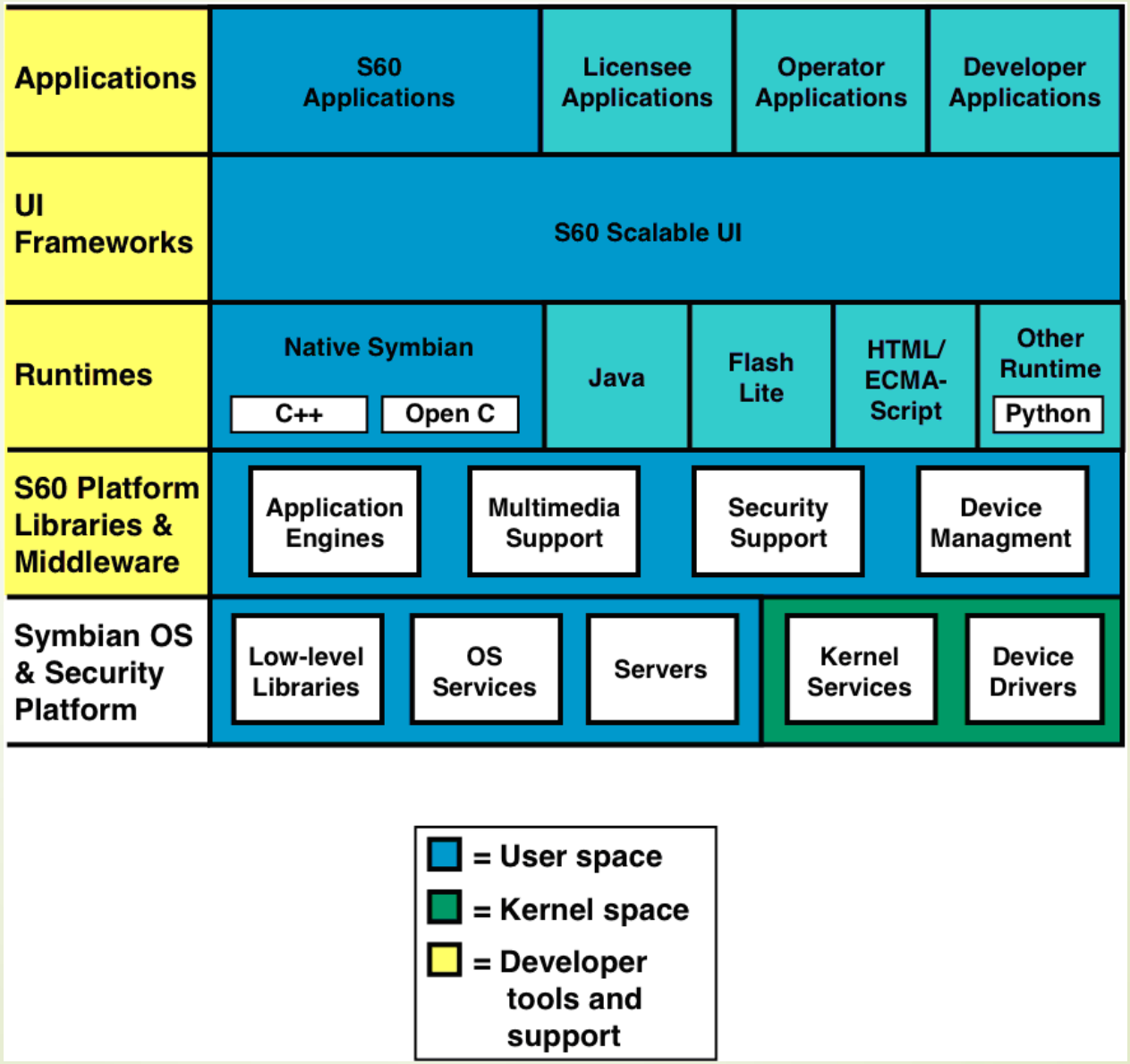
- Symbian is the first operating system for mobile devices. Symbian is the most popular mobile OS until the end of 2010. It is partially open source, designed for mobile devices.
- Symbian architecture is based on a single microkernel, which only deals with memory protection and message brokering between active processes.
- The rest of the functionality is implemented using server processes, which are used asynchronously by sending messages to them, rather than traditionally through system calls.

1. Symbian

1.1. Characteristics

- Symbian OS is written in C++, but does not use its standard libraries, but implements its own, which are designed from the beginning to use less memory (and, accordingly, less electrical energy needed to power the device running under operating system).
- These libraries are somewhat lower-level than the language's standard ones, which makes them more difficult to use and slows down the programming of applications for the operating system.

S60 software stack



1. Symbian

1.2. Symbian software stack 60

- The software layers of the platform allow supporting the different needs of Nokia, the network operator and the developer.
- The topmost layer is **Applications, in which** S60 licensed, operator or user programs run.
- Below this layer is the **S60 Scalable UI layer**. It is developed and licensed by Nokia. Contains the structures of the user interface (UI frameworks).

1. Symbian

1.2. Symbian software stack 60

- The visual components of these structures adapt to different screen sizes or when the orientation of the device changes from a vertical to a horizontal position. SVG-T (Scalabe Vector Graphics – Tiny) and relative positioning
- These UI components also adapt device localization settings and policies, as well as which direction to display text and how to display the calendar and time. A toolkit of ready-to-use components is also provided: lists, editable forms etc.

1. Symbian

1.2. Symbian software stack 60

- The next layer is **Runtimes**. As its name suggests, it contains executable libraries that support various programming languages, interactive content rendering, and web rendering.
- For example, this layer contains the executable libraries for the Symbian C++ and Open C/C++ programming languages. It also contains the Java ME executable. Web support for the S60 browser as well as Web Runtime (WRT) add-ons is provided via the WebKit rendering engine. The layer also supports other executables, such as Falsh Lite and Phyton.

1. Symbian

1.2. Symbian software stack 60

- **The Platform** layer **Libraries and Middleware** contains the structures used to implement specific services for the higher-level layers, or provides APIs for developers to work with the low-level operating system.
- also contains various application engines that manage personal information data, messaging and data synchronization.

1. Symbian

1.2. Symbian software stack 60

- Multimedia content processing (e.g. video decoding) is performed in this layer.
- Interfaces for multimedia applications for camera operation and sound recording are also provided.
- Security framework manages the security certificates and keys used in secure data sessions.

1. Symbian

1.2. Symbian software stack 60

- The layer **Symbian OS and Security Platform** is at the lowest level of the layers.
- Symbian OS provides core system services for the S60 platform.
- The operating system is based on a microkernel, with the microkernel residing in the kernel space while the rest of the operating system and software stack reside in higher layers. This provides security and memory protection for important low-level services.

1. Symbian

1.2. Symbian software stack 60

- In addition, the operating system places each application in its own address space, which isolates it from the operating system and from other applications, ensuring security and reliability.
- layer enables businesses to lock or wipe lost mobile devices.

2. Google Android

2.1. Characteristics

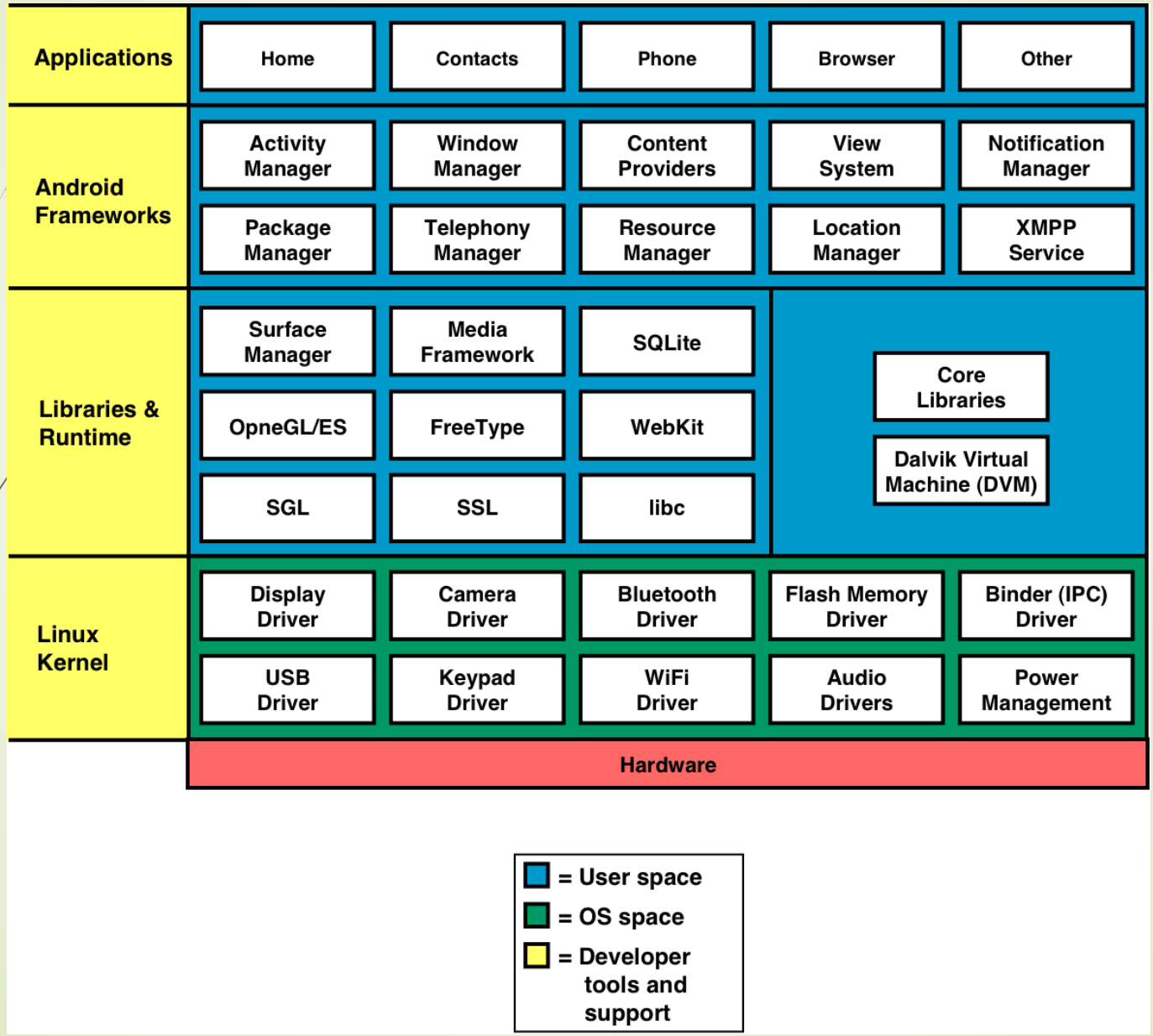
- Android is a mobile operating system running on the Linux kernel.
- It allows developers to write code in the Java language using Google-created Java libraries.
- The applications are written in Java, but Android is not Java ME and does not support such applications.
- The development of Android is taken care of by a large number of software developers who create so-called apps - small applications that expand the functionality of the system.

2. Google Android

2.1. Characteristics

- Applications can be downloaded from various sites on the Internet or from large online stores such as for example Google Play - Google's store.
- Google Android has its own integrated application development environment - the Android SDK, which includes a mobile emulator for mobile devices, debugging tools, profiling, as well as a plug-in to the popular Eclipse environment for developing Java-based applications.
- The latest version the operating system is Android 13 as of August 15, 2022.

Android software stack



2. Google Android

2.2. Android

- The Android software stack consists of four layers.
- **Applications** layer contains all Android and third-party applications, and multiple applications can be run at the same time.
- Below this layer is the **Android Frameworks layer**. This layer consists of Java classes that provide functions for applications such as window management, displaying content in a window, passing messages between applications, and intercepting phone calls.

2. Google Android

2.2. Android

- Because the interface is built with the Java language and sorc code is available, the user can modify these classes to extend their capabilities or change their behavior.
- Some of the lowest layers in the stack represent interfaces written in C++.

2. Google Android

2.2. Android

- The next layer is **Libraries & Runtime**. The libraries provide support for 2D and 3D graphics and decoding of multimedia content.
- This layer holds engines supporting application functions such as SQLite for working with databases and WebKit for rendering web content.

2. Google Android

2.2. Android

- Like Java ME, hardware independence is provided by using a bytecode interpreter that runs Android applications.
- But it doesn't use Sun 's JVM (Java Virtual Machine), it uses its own DVM (Dalvik Virtual Machine).
- The advantage of using a different bytecode interpreter is that the DVM is designed so that multiple instances of it can be run, each in its own protected memory space, and each can run an application.

2. Google Android

2.2. Android

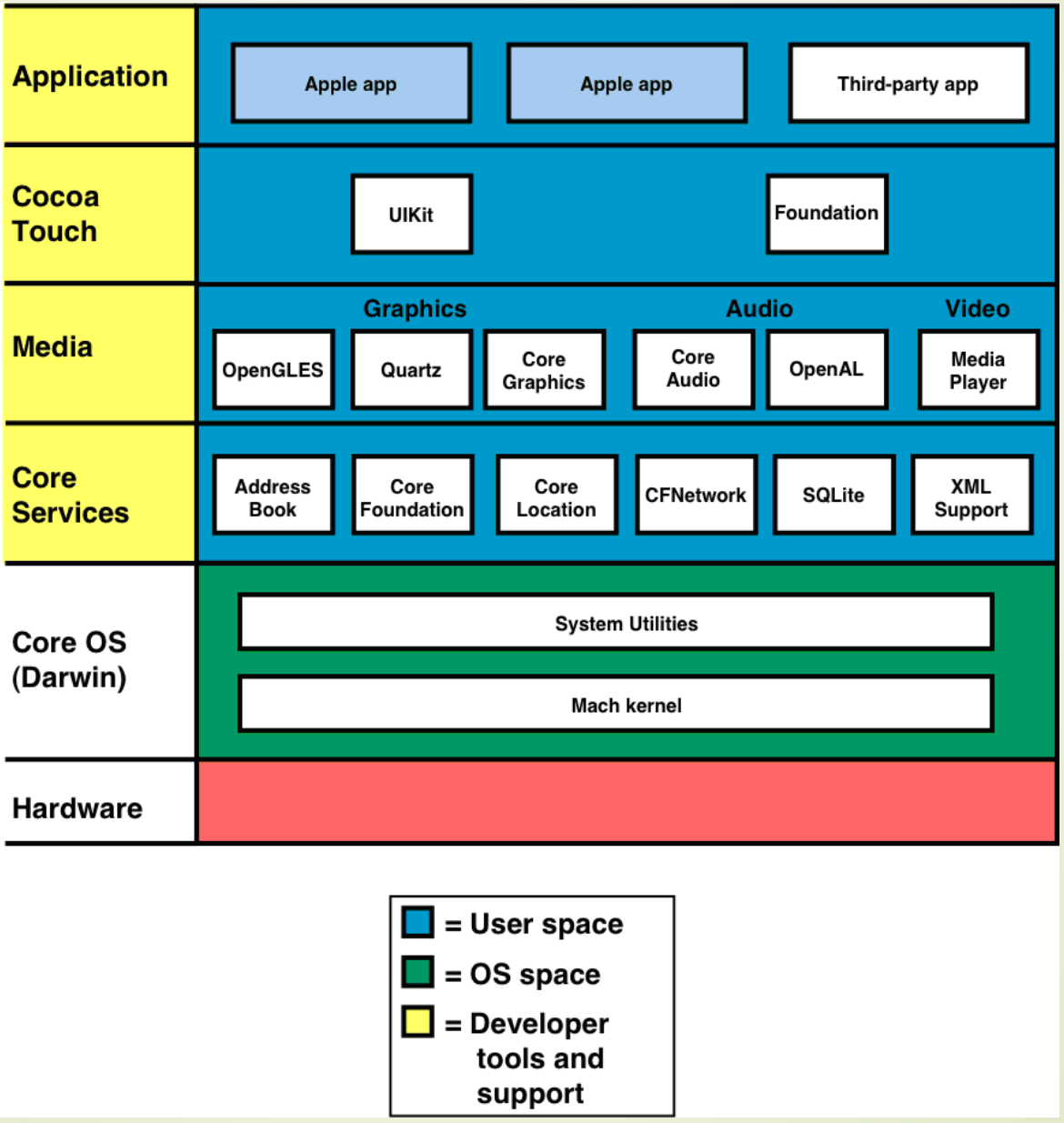
- The lowest layer of the software stack is **the Linux Kernel**. It provides multitasking and system services such as threads, network services, and process management.
- This layer contains all the low-level drivers and manages the power consumption of the device.

3. Apple iOS

3.1. Characteristics

- iOS is a mobile operating system of Apple Inc.
- Developed for the iPhone, it is also used in Apple's iPod Touch, iPad and Apple TV mobile devices.
- Apple does not allow iOS to work with third-party hardware.
- iOS is the second most popular mobile operating system in the world after Android.
- The current version of the operating system is iOS 16, released on September 12, 2022.

iPhone OS software stack



3. Apple iOS

3.2. Software stack

- Like other mobile platforms, the top layer of the iPhone OS stack houses the **Application layer**.
- Applications developed by Apple can run simultaneously, but only one external application can work together with them.
- The platform does not have any Java support, so Java ME applications cannot run.

3. Apple iOS

3.2. Software stack

- Below the first layer is the **Cocoa Touch layer**.
- It contains structures that manage the user interface (to intercept actions, to manage windows, to display graphics in those windows).
- Cocoa Touch is a subversion of the Apple Cocoa framework that is object-oriented and written in Objective-C.
- Cocoa provides many classes or components through which the user can create a fully functional application.

3. Apple iOS

3.2. Software stack

- But Cocoa Touch structures are limited to work only with this platform.
- They are well balanced between abstractly representing low-level hardware and allowing the developer to use mobile-specific features.
- For example, Cocoa Touch components handle most of the screen writing and media execution, while other APIs allow access to the acceleration sensor and the camera.

3. Apple iOS

3.2. Software stack

- At a lower level in the software stack is the **Media layer**.
- This layer controls the visualization of graphics, the generation of sound, and the playback of sound and video files.
- While Cocoa Touch provides a high level for generating animations and graphics, structures from the Media layer provide finer control over content playback.
- Three-dimensional objects are rendered using the OpenGL ES framework, which conforms to the OpenGL ES 1.1 specification.

3. Apple iOS

3.2. Software stack

- This framework uses hardware accelerators to provide full-screen animations at high frame rates.
- This layer also uses Quartz, a vector-based graphics engine for drawing 2D graphics and applying graphical effects. Quartz is identical to that used in Mac OS X.
- The Core Graphics structure supports complex animation and visual effects, and the main work in this case is taken over by the hardware. The structures for recording and playing back sound and video files are also located in this layer.

3. Apple iOS

3.2. Software stack

- **Core Services** layer provides system services for the higher-level layers. It contains structures and machines supporting an address book, a SQL database (SQLite), positioning services (using GPS coordinates), and communication services.
- The security framework manages the digital certificates, keys, and access policies that protect application data.

3. Apple iOS

3.2. Software stack

- **Core** layer provides the basic services of the operating system.
- consists of kernel, drivers and OS interfaces. The kernel is based on Mach and manages the low-level functions – virtual memory, threads, sockets, math calculations, file system access, etc.
- Only a few higher-level structures have access to the kernel and drivers. If necessary, an application can indirectly access these services by using C-based interfaces from the LibSystem library.

4. RIM Blackberry OS

4.1. Characteristics

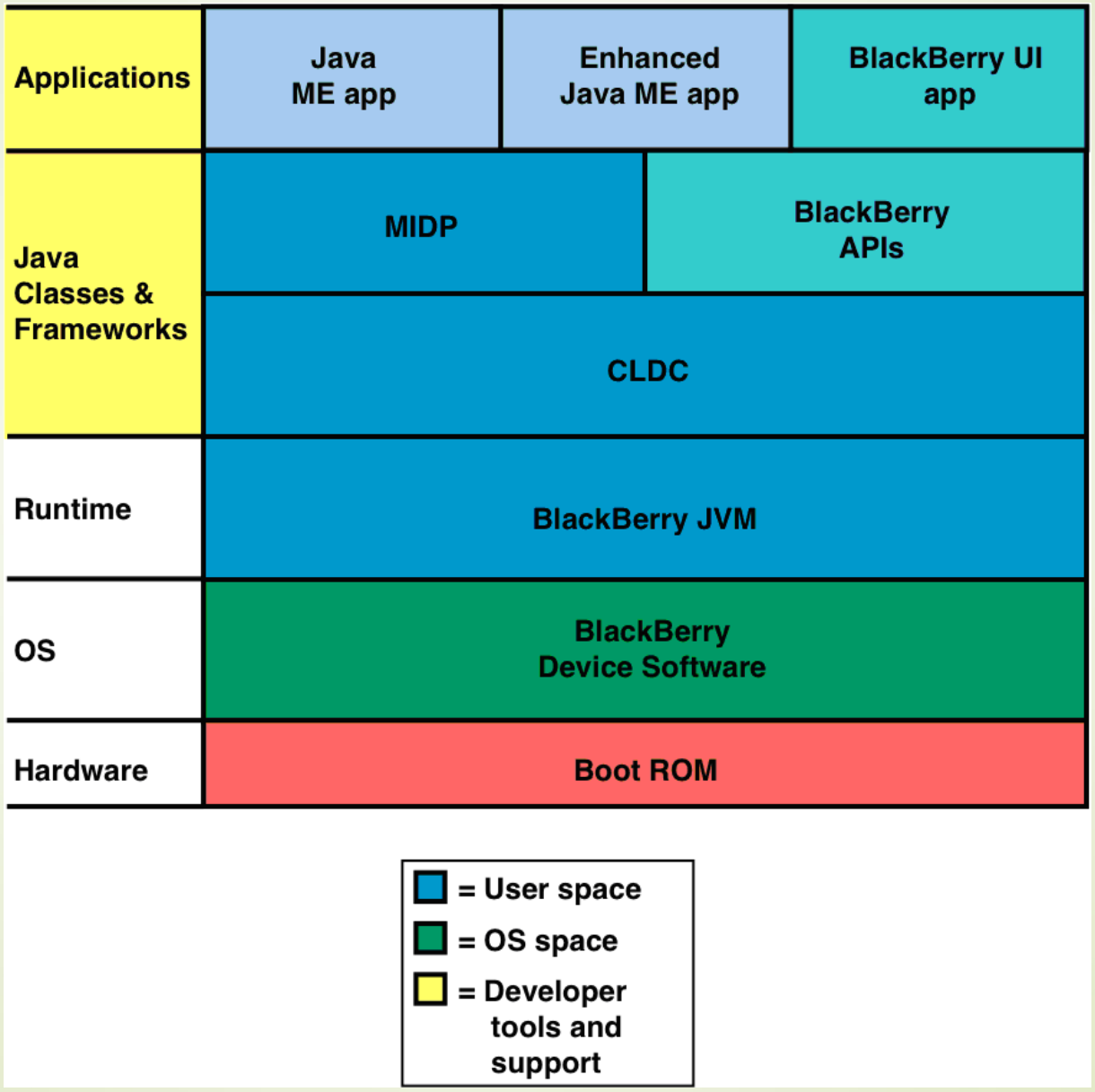
- It is a specialized operating environment created by a specific mobile phone manufacturer for use only in their branded devices (like Apple).
- In this case, it is about RIM (Research-In-Motion), one of the companies that is said to be the originator of the "business phone" class, thanks to their popular BlackBerry brand.

4. RIM Blackberry OS

4.1. Characteristics

- Most BlackBerry devices are known for their ability to send and receive instant messages and e-mail at the push of a button, while maintaining a high level of security for the message through an encryption device.
- The latest version of the operating system is BlackBerry 10 (since 2013).

BlackBerry OS software stack



4. RIM Blackberry OS

4.2. Software stack

- At the top level is the **Application s layer**.
- It runs Java ME applications (MIDlets) and BlackBerry UI applications.
- possible to take existing Java ME code and add BlackBerry- specific classes to produce a hybrid Java ME application.
- For example, a call can be made to the BlackBerry API to select an audio output device (speakers or headphones), and then use a standard media player class to play the audio content.

4. RIM Blackberry OS

4.2. Software stack

- The next lower layer is **Java Classes & Frameworks**.
- This layer resembles the Java ME platform. Contains the usual MIDP MIDlet classes that manage the user interface and application lifecycle.
- They are based on Connected Limited Device Configuration (CLDC) classes that provide access to lower-level resources.

4. RIM Blackberry OS

4.2. Software stack

- This layer also supports useful Java Specification Request (JSR) API packages such as JSR-75 (personal information management and file handling services), JSR-135 (interception and playback of multimedia), JSR-82 (Bluetooth support), JSR-120 (wireless messaging), and JSR-179 (positioning services), etc.
- All of these classes, as well as those of an application, are loaded and executed by **the BlackBerry JVM**.

4. RIM Blackberry OS

4.2. Software stack

- The BlackBerry API extensions found in this layer extend the capabilities of the platform in several ways.
- of all they provide UI API for user menu, addons and screens.
- Second, the Application class allows the application to remain and continue to run, unlike the MIDP Midlet class, which requires the application to terminate when it is closed.
- Other APIs manage network sessions or I/O to servers.

4. RIM Blackberry OS

4.2. Software stack

- Additionally, these APIs provide access to camera, media player, and web browser functionality.
- BlackBerry mobile application can only be created using CLDC and the BlackBerry API. Such an application has access to all the capabilities of the device - Bluetooth, acceleration converter, touch display, etc.
- It can run concurrently with other applications.

4. RIM Blackberry OS

4.2. Software stack

- also possible for the application to be launched when the device is booted and then continue to run in the background.
- Creating an app that uses BlackBerry API extensions makes it dependent on them and can only run on that platform.

4. RIM Blackberry OS

4.2. Software stack

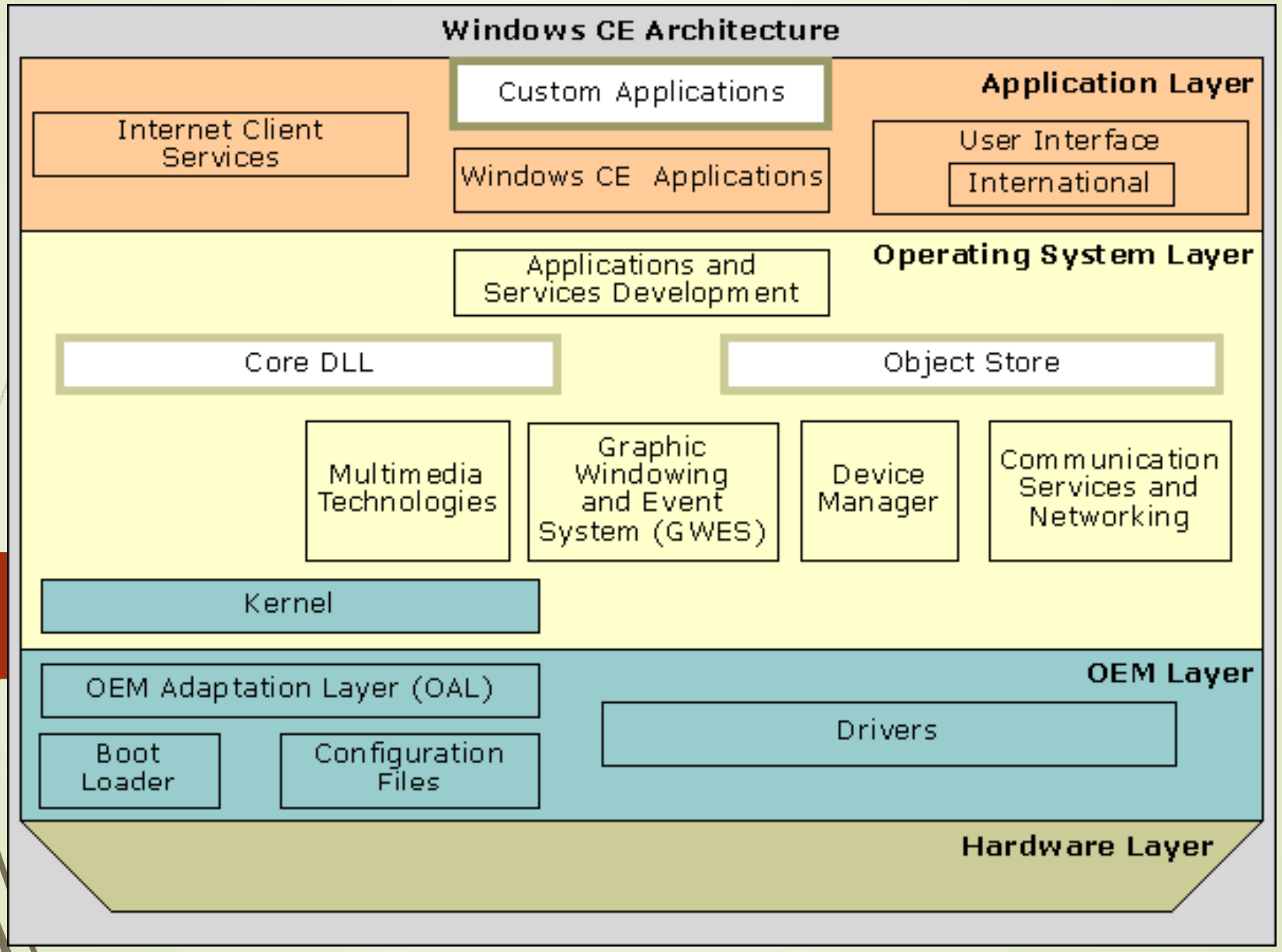
- **The BlackBerry Device Software** layer is a low-level multitasking, multithreaded operating system.
- It uses a method to track actions from a specific device using snooping threads. For example, such snooping threads manage push technology for e-mail and messaging.
- BlackBerry Device Software layer makes it possible to configure applications and turn off certain smartphone functions, as well as remotely wipe the contents of a lost device.

5. Windows Phone OS

5.1. Characteristics

- Windows Phone (abbreviated as WP) is an operating system for mobile devices (smartphones) developed by Microsoft and successor to the Windows Mobile platform.
- From version WP 7, the direction is more towards ordinary users and not so much towards enterprise customers.
- The emphasis in the functionality of the new versions of the operating system on the multimedia functionality.
- The current version is Windows Phone 10 (since 2016).

Windows CE 5.0 software stack



5. Windows Phone OS

5.2. Software stack

- **Applications** layer contains all Windows, third-party, and user applications.
- In this layer are the modules for implementing the Internet services, as well as for maintaining the user interface and for its localization.
- The layer below Applications is the actual operating system layer.
- It houses the kernel, the multimedia file handling functions, the device management manager, the communication maintenance functions, the Graphics Windows and Events Subsystem (GWES), and more.

5. Windows Phone OS

5.2. Software stack

- The layer (Object Store module) also stores the file system, registers and database.
- The core (Kernel) provides the core functionality of any Windows based device.
- This functionality includes process, thread and memory management. The kernel also provides some file management features.

5. Windows Phone OS

5.2. Software stack

- At the lowest software level and closest to the hardware of the mobile device is the **OEM layer**.
- Windows adapts to a specific hardware platform by creating a thin layer of code that sits between the kernel and the hardware platform.
- This layer is known as **OEM Adaptation Layer** (OAL). OAL isolates mobile device-specific hardware functions from the kernel.

5. Windows Phone OS

5.2. Software stack

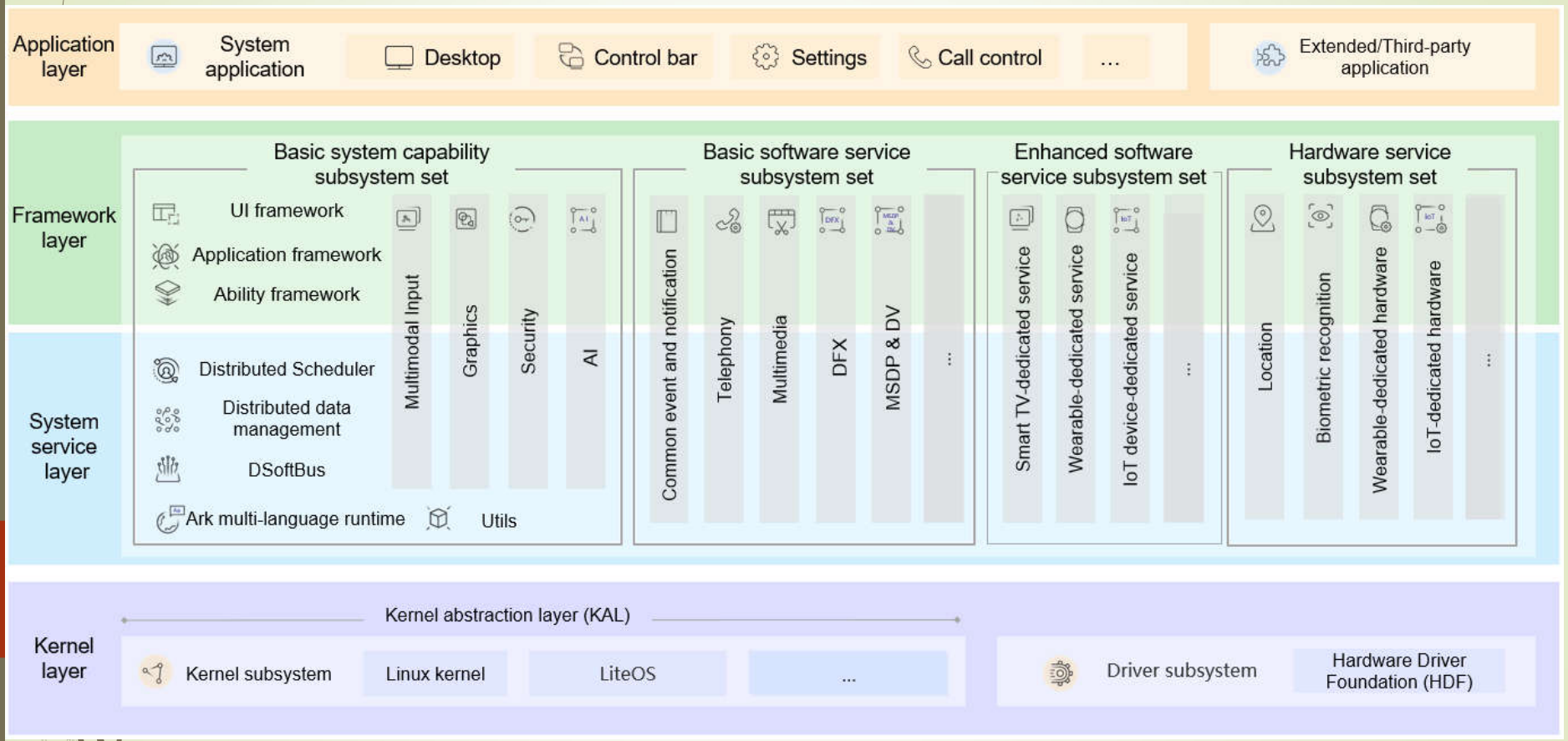
- On the other hand, the Windows kernel contains processor-specific code to work with the processor's functions.
- OAL is processor and hardware platform specific.
- The main task of OAL is to provide kernel access to the hardware. This includes managing hardware timers and interrupts, as well as managing device peripheral power consumption. This layer also houses the built-in drivers for managing the screen, keyboard, sound, battery, and more.

6. HarmonyOS

6.1. Characteristics

- HarmonyOS is an innovative, distributed operating system designed for use in the Internet of Everything (IoE) era.
- Developed by Huawei and intended for use in their mobile devices.
- Unlike traditional operating systems that run on a standalone device, HarmonyOS is built with a distributed architecture.
- It uses the same set of system capabilities to adapt to a wide range of devices (from phones and tablets to smart TVs and virtual reality helmets).
- The current version is 3 as of July 27, 2022.

HarmonyOS


Application layer
System application
Desktop
Control bar
Settings
Call control
...
Extended/Third-party application
Framework layer
Basic system capability subsystem set
UI framework
Application framework
Ability framework
Multimodal Input
Graphics
Security
AI
Basic software service subsystem set
Common event and notification
Telephony
Multimedia
DFX
MSDP & DV
...
Enhanced software service subsystem set
Smart TV-dedicated service
Wearable-dedicated service
IoT device-dedicated service
...
Hardware service subsystem set
Location
Biometric recognition
Wearable-dedicated hardware
IoT-dedicated hardware
...
System service layer
Distributed Scheduler
Distributed data management
DSoftBus
Ark multi-language runtime
Utils
Kernel layer
Kernel abstraction layer (KAL)
Kernel subsystem
Linux kernel
LiteOS
...
Driver subsystem
Hardware Driver Foundation (HDF)

6. HarmonyOS

6.2. Software stack

- HarmonyOS is designed a with a multi-layered architecture, which from top to bottom consists of Application layer, Framework layer, System service layer and Kernel layers.
- System functions expand in levels, from system to subsystem and then to function/module.
- In a multi-device deployment scenario, unnecessary subsystems, functions or modules can be excluded from the system as required.

6. HarmonyOS

6.2. Software stack

- **Application Layer**
  - This layer consists of system and third-party applications.
  - Each HarmonyOS application supports one or more Feature Abilities (FA) or Particle Abilities (PA).
  - FA provides a user interface for user interaction. PA has no user interface and provides background task processing as well as data access.
  - During user interaction, FAs may need to retrieve data in the background from PAs.
  - Applications developed on the basis of FA and PA fulfill specific business characteristics and can run on different devices.

6. HarmonyOS

6.2. Software stack

- **Framework Layer**
  - This layer provides means for developing applications for HarmonyOS: application framework and ability framework specific to multiple programming languages (Java, C, C++, JavaScript (JS) and TypeScript (TS)), Java UI, JS/TS Huawel ArkUI.
  - The Framework Layer also provides multilingual APIs for hardware and software services.
  - available for different HarmonyOS devices vary depending on component-based adaptation.

6. HarmonyOS

6.2. Software stack

- **System Service Layer.** This layer consists of the following parts:
  - **Basic system capability subsystem set**: provides distributed application execution, scheduling and migration between HarmonyOS devices. It provides the following basic capabilities: DSoftBus (a standardized way to connect multiple devices (of different types) to create a single "super device". This allows one device to control others and data can be freely shared between them.), distributed data management, distributed scheduling, multilingual Ark execution, utilities, multimodal input, schedule maintenance, security, and AI. Ark provides compilation / execution of programs written in C, C++ and JavaScript and provides libraries of core system classes. It also provides execution of Java programs compiled by Ark.

6. HarmonyOS

6.2. Software stack

- **Basic software service subsystem set**: provides HarmonyOS with common and universal software services, including common events and notifications, telephony, multimedia, Design For X (DFX), as well as the Mobile Sensing Development Platform (MSDP) and Device Virtualization (DV).
- **Enhanced software service subsystem set:** provides HarmonyOS with specific and enhanced software services, including those designed for smart TVs, wearables, IoT devices, and more.
- **Hardware service subsystem set:** Provides HarmonyOS hardware services including for location, for biometric recognition, as well as those designed for wearable and IoT devices.

6. HarmonyOS

6.2. Software stack

- **Kernel Layer**
  - **Kernel subsystem**: HarmonyOS uses a multi-core design so that appropriate OS kernels can be selected for devices with different resource constraints. The Kernel Abstraction Layer (KAL) protects the various kernel implementations and provides the upper layer with basic kernel capabilities, including process and thread management, memory management, file system management, network management, and edge management.

6. HarmonyOS

6.2. Software stack

- **Kernel Layer**
  - **Driver subsystem**: The Hardware Driver Foundation (HDF) lays the foundation for an open HarmonyOS hardware ecosystem. It allows unified access from peripheral devices and provides basic capabilities for driver development and management.

6. HarmonyOS

6.3. Application development languages

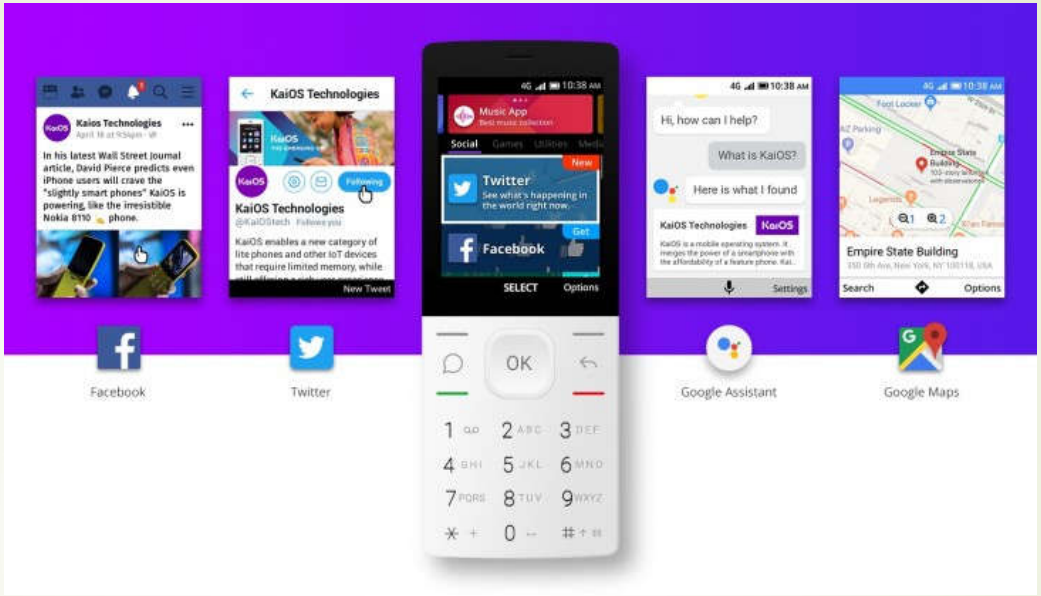
- For developing HarmonyOS applications provides APIs for multiple programming languages.
- You can choose between: Java, C/C ++, JavaScript, Cascading Style Sheets (CSS), Extensible Markup Language (XML), and HarmonyOS Markup Language (HML).

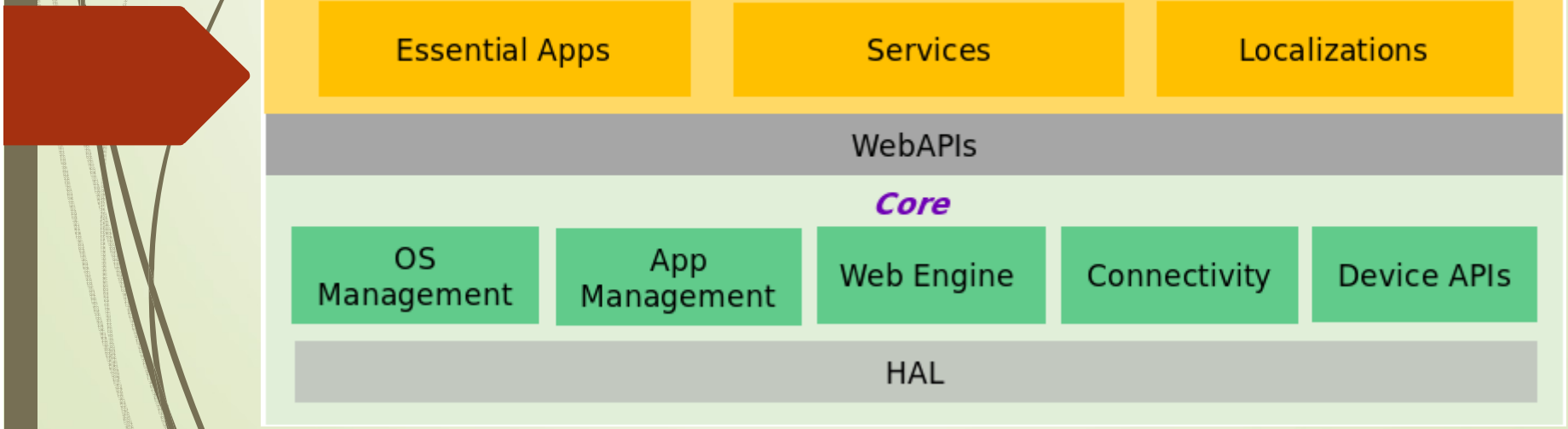
7. KaiOS

7.1. Characteristics

- KaiOS is a mobile operating system based on Linux. It was developed by KaiOS Technologies (Hong Kong) Limited.
- What makes KaiOS different is the goal of bringing "smartphone" capabilities to low-cost, resource-constrained (up to 256 MB memory), non-touchscreen cell phones. and with reduced energy consumption. These possibilities include work with HTML5 applications, support for 4G LTE, GPS, Wi-Fi, NFC, as well as dual SIM compatibility.
- The current version is 3.1 as of March 2022. There are currently over 170 million users.

KaiOS


Facebook
Twitter
Google Assistant
Google Maps
Music App
Twitter
Facebook
SELECT
Options
OK
Hi, how can I help?
What is KaiOS?
Here is what I found
KaiOS Technologies
KaiOS
Settings
Empire State Building
Search
Options


App profile
Essential Apps
Services
Localizations
WebAPIs
Core
OS Management
App Management
Web Engine
Connectivity
Device APIs
HAL

7. KaiOS

7.2. Software stack

- The KaiOS software stack is divided into two layers – App Profile and Core.
- **App Profile :** Contains a collection of built-in web applications that represents the features of the KaiOS user interface for a given device form factor.
- **Core:** Consists of Web application runtime (Gecko), hardware adaptation layer (HAL) and others supporting the module.

7. KaiOS

7.3. Technologies

- The KaiOS apps are based on web technologies – HTML, CSS and JavaScript and run by the Gecko runtime.
- Rendering is done through HTML/CSS parsing, and graphics APIs are used to render images.
- JavaScript is executed by JavaScript engine SpiderMonkey and can connect to C ++ components using XPConnect and WebIDL connections.