SVKM'S NMIMS Deemed to be UNIVERSITY | MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING

MC Case Study

Mobile Operating Systems

Submitted by

Sadhiman Das B020
Akash Dubey B026
Manan Jain B037
Vivek Joshi B040
Astha Kacker B041

Under the Guidance of
Prof. Artika Singh

Mukesh Patel School of Technology Management & Engineering, Mumbai

APRIL 2021

TABLE OF CONTENTS

Introduction

Mobile Operating System is software that allows smartphones, tablet PCs and other devices to run applications and programs. It manages the hardware and software resources of a mobile device similar to a computer OS. Some OS platforms cover the entire range of the software stack while others may only include the lower levels (typically the kernel and middleware layers) and rely on additional software platforms to provide a user interface framework.

Android OS:

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. It offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 whereas the first commercial version, Android 1.0, was released in September 2008.

Android is a powerful operating system and it supports a large number of applications on Smartphones. These applications are more comfortable and advanced for users. The hardware that supports android software is based on the ARM architecture platform. Android development supports the full java programming language. Even other packages that are API and JSE are not supported.

Android is open source and Google releases the code under the Apache License. This open-source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. The OS is architected in the form of different layers of stacked software that comprises android applications, an operating system, android run-time, middleware, services and libraries.

Symbian OS:

Before the Android world ruled Smartphones, the Bugatti of operating systems for smart mobile devices was the Symbian OS. The platform was popular up until 2010 when Google's Android gave it a run for its money. Eventually, its development ceased in mid-2014.

The inception of the Symbian platform began with a system referred to as EPOC, an OS which was created in the 1980s by one company we have fond memories of; Psion. In 1998, Sony Ericson, Nokia, and Motorola came together and formed Symbian Ltd. EPOC hence became Symbian OS. The system was designed to run on ARM processors and was used to power some of the most powerful smartphones at the time.

Samsung and LG also joined the Symbian world after it was born and in 2000, Ericson R380 became the first Symbian mobile device in the world. The OS would enjoy a substantial market share until it faced several challenges that led to its gradual failure.

In 2010, when other members including LG, Samsung, and Sony Ericson, adopted other operating systems, Nokia took over the running of the Symbian Foundation and transitioned it to licensing organizations only. Not long after, the market shares began dropping and with stiff competition from iOS and Android, the Symbian death row began. The difficulty in programming due to fragmentation greatly contributed to its demise.

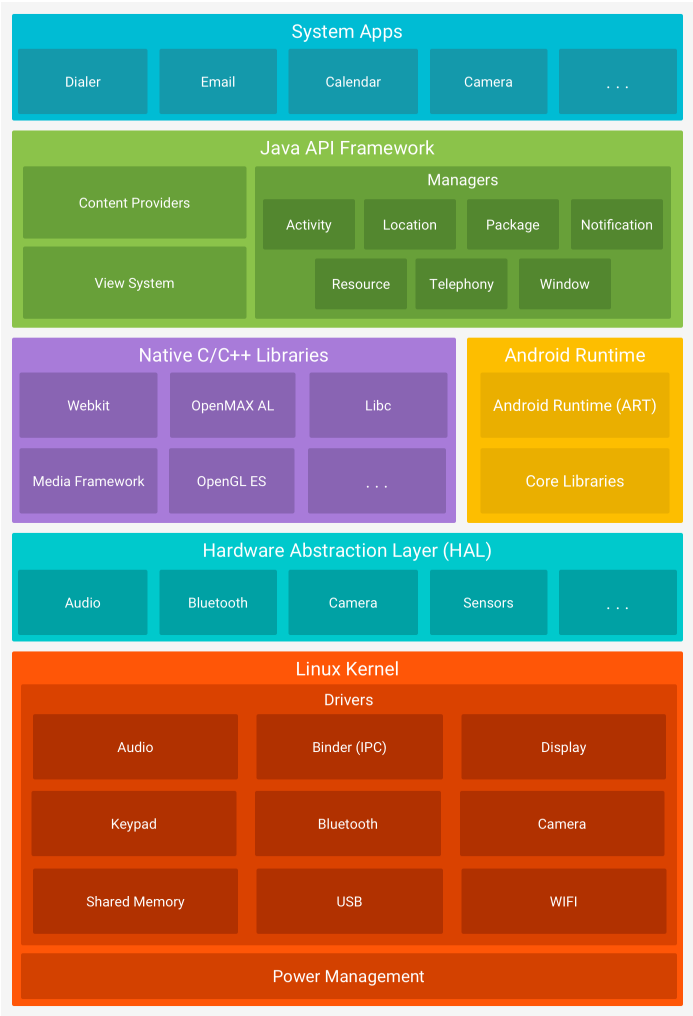
In February 2011, after Nokia's newly appointed CEO announced that the company was partnering with Microsoft to develop Nokia devices running Windows, the Symbian platform gradually dropped. Developers deserted the ecosystem rapidly.

Two months later, the company stopped sourcing Symbian codes and substantially decreased the number of collaborations it had to a mere group of specific partners in Japan. In June the same year, Accenture, through an agreement with Nokia began outsourcing the Symbian-based resources and support services.

In 2014, the support, development and the maintenance of the Symbian platform were ceased. Developers could no longer create or develop more applications for Symbian. But the applications that had already been published are still available for download.

Architecture

Android OS:



Android OS architecture is a stack of software components which is roughly divided into five sections and four main layers. It is an open source, Linux-based software stack created for a wide array of devices and form factors. The diagram above shows the major components of the Android platform. The layers are as follows:

System Apps

The operating system comes with a set of core apps for email, SMS messaging, calendars, internet browsing, contacts, and more. Apps included with the platform have no special status among the apps the user chooses to install. So a third-party app can become the user's default web browser, SMS messenger, or even the default keyboard.

This provides a sense of customization and also helps the user use the applications of their choice.

The system apps function both as apps for users and to provide key capabilities that developers can access for their own app like opening youtube videos through google search. This helps provide default applications for developers to use and implement for their applications but can also be changed according to the users choice.

Java API Framework

The entire feature-set of the Android OS is available to you through APIs written in the Java language. These APIs form the building blocks that are used to create Android apps by simplifying the reuse of core, modular system components and services, which include the following:

- A rich and extensible View System you can use to build an app's UI, including lists, grids, text boxes, buttons, and even an embeddable web browser
- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files
- A Notification Manager that enables all apps to display custom alerts in the status bar
- An Activity Manager that manages the lifecycle of apps and provides a common navigation back stack
- Content Providers that enable apps to access data from other apps, such as the Contacts app, or to share their own data

Developers have full access to the same framework APIs that Android system apps use.

Native C/C++ Libraries

Many core Android system components and services, such as ART and HAL, are built from native code that require native libraries written in C and C++. The Android platform provides Java framework APIs to expose the functionality of some of these native libraries to apps. OpenGL ES can be accessed through the Android framework's Java OpenGL API to add support for drawing and manipulating 2D and 3D graphics in applications. While developing an app that requires C or C++ code, the Android NDK can be used to access some of these native platform libraries directly from your native code.

With devices running Android version 5.0 (API level 21) or higher, each app runs in its own process and with its own instance of the Android Runtime (ART). ART is written to run multiple virtual machines on low-memory devices by executing DEX files, a bytecode format designed specially for Android that's optimized for minimal memory footprint. Build tools, such as d8, compile Java sources into DEX bytecode, which can run on the Android platform.

Some of the major features of ART include the following:

- Ahead-of-time (AOT) and just-in-time (JIT) compilation
- Optimized garbage collection (GC)
- On Android 9 (API level 28) and higher, conversion of an app package's Dalvik Executable format (DEX) files to more compact machine code.
- Better debugging support, including a dedicated sampling profiler, detailed diagnostic exceptions and crash reporting, and the ability to set watchpoints to monitor specific fields

Prior to Android version 5.0 (API level 21), Dalvik was the Android runtime. Android also includes a set of core runtime libraries that provide most of the functionality of the Java programming language, including some Java 8 language features, that the Java API framework uses.

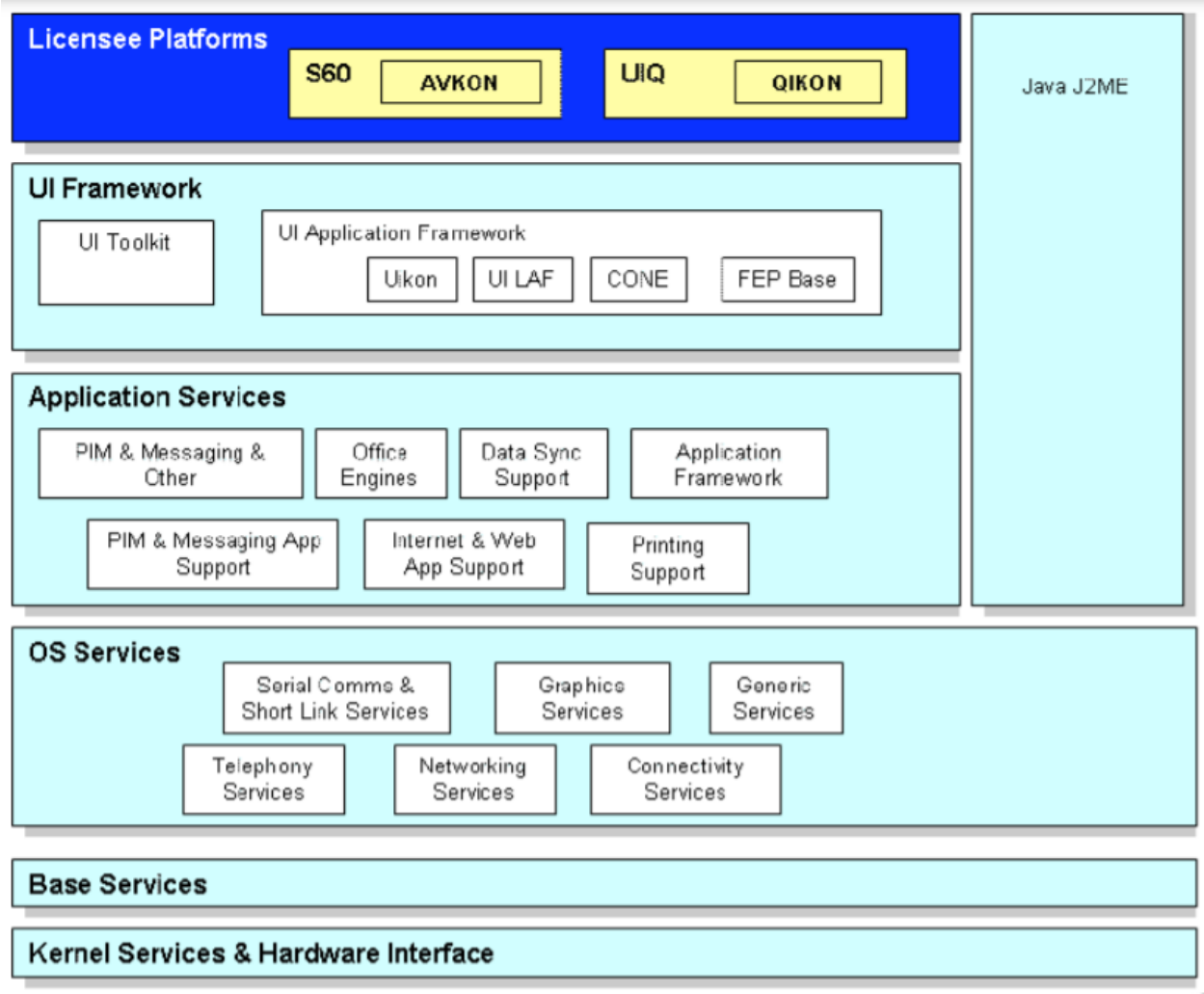
Hardware Abstraction Layer (HAL)

The hardware abstraction layer (HAL) provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework. The HAL consists of multiple library modules, each of which implements an interface for a specific type of hardware component, such as the camera or bluetooth module. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component.

The Linux Kernel

The foundation of the Android platform is the Linux kernel. The Android Runtime (ART) relies on the Linux kernel for underlying functionalities such as threading and low-level memory management. Due to the Linux kernel, Android can take advantage of key security features and allows device manufacturers to develop hardware drivers for a well-known kernel.

Symbian OS:



The Symbian operating system is composed of a set of layers. Most prominent logical layers of the operating system include -

- User Interface Framework
- Application Services
- Operating System Services
- Base Services
- Kernel Services
- Hardware Interface

The Figure shows stack of Symbian operating system layers. Symbian operating system has headless configuration i.e. minimal user interface features are supported by core operating system. Therefore, third parties have developed user interface layers on top of core Symbian operating system, depicted in Figure 1 as top most layers.

These third party libraries include

- S60 - developed by Nokia
- UIQ - developed by Sony Ericsson
- MOAP - developed by NTT DoCoMo

User applications reside typically on top of S60, UIQ or MOAP. Java MicroEdition libraries exist as separate component in the operating system.

Symbian OS layers are further divided into blocks and sub-blocks. Each block and sub-block is a collection of individual components. Thus, a layer is the highest level of abstraction, while a component is a lower level of abstraction. Components are physical realization of more logical concepts such as layers and blocks. Components consist of software code including source code, executables, libraries,and documentation.Each layer abstracts the functionality of layer below it and provides services to layer above it. The level of abstraction increases as we move up from the hardware (at the lowest level) to the user interface (at the highest level).While layers provide a basic categorization of OS services, blocks and sub-blocks correspond to specific technology domains. Each block consists of a collection of components that provides a set of related services. For example, The OS Services layer contains a Communication Services block which is further decomposed into Telephony, Short Link and Networking Services sub-blocks. Localization of MobilePlatforms 14

The following sections briefly describe Symbian OS layers and their basic functionality. All Symbian OSreleases from v7.0 to v9.3 have the same layer decomposition.

User Interface (UI) Framework Layer

The top most layer of Symbian OS provides libraries and framework for constructing a graphical user interface. This layer includes class hierarchies of user interface controls and concrete widget classes.Third party graphical user interface libraries such as S60 and UIQ have been built by extending the functionality available in user interface framework layer.

User Interface Layers on Symbian OS

User interface framework is the topmost layer of Symbian OS. It provides the framework support on which a production user interface is built. The three currently available custom user interfaces are S60,UIQ and MOAP.

- **S60** - S60 platform is developed and licensed by Nokia. It supports touch screen, keypad, 5-way navigator, soft keys. Lenovo, LG, Panasonic and Samsung have also shipped S60 enabled phones by licensing S60 from Nokia in the past [1]. In the past, S60 has been shipped in various versions such as 1st, 2nd, 3rd, and 5theditions. After S60 5th edition, Nokia has shipped S60 and Symbian as one open source package under the umbrella of Symbian Foundation and various versions of packages have been namedSymbian^1, Symbian ^2, and, more recently, Symbian ^3.

- **UIQ (User Interface Quartz)** - UIQ was developed and licensed by UIQ Technology owned by SonyEricsson. It is most commonly used on Sony Ericsson's P series of smart phones, such as the P990. Otherdevices shipped with UIQ include Sony Ericsson P990, W950 and W960i. UIQ, however, has not been made part of latest breed of Symbian operating system versions i.e. Symbian ^1 and later editions.
- **MOAP (Mobile Oriented Application Platform)** - MOAP has been developed by FOMA (Freedom of Mobile Access) consortium in Japan. It is a proprietary platform used only by NTTDoCoMo (i.e. not licensed to others).

Application Services Layer

This layer provides application support independent of user interface layer. Application services are broadly classified into three main categories:1. System level services that provide basic application framework support to all applications,2. Technology-specific services such as multimedia, telephony, mail, messaging, and browsing,3. Services that support generic types of applications such as Personal Information Management (PIM)and Alarm Server.15 Symbian Operating System Architecture

OS Services Layer

This layer acts as a middle-ware between the base services layer at a lower level and the applicationservices layer at an upper level. The services provided by this layer can be divided into four broad categories:

1. Generic operating system services such as task scheduler
2. Communications services such as telephony, short-link services, and network services
3. Multimedia services such as windows server, font server, and multimedia framework
4. Connectivity services such as services for interaction with desktop for file browsing and services for software installation.

Base Services Layer

The base services layer serves as the user side of the two-layer Symbian OS base system. It encapsulates servers, libraries and frameworks that are built on the kernel layer in order to provide upper layers basic operating system services such as file server, basic programming library, persistence model and cryptography library.

Kernel Services and Hardware Interface Layer

The lowest layer of the Symbian operating system contains the operating system kernel and includes components that interface with underlying system hardware. It includes logical and physical device drivers, scheduler and interrupt handler, timers, mutexes etc. In order to port Symbian OS to a new hardware, kernel layer is customized.

Java Micro Edition (Java ME)

Java ME has been built into the Symbian operating system as a separate component and it interacts with multiple system layers. It contains MIDP and CLDC libraries, Java Virtual Machine (JVM) and plugins for interaction with native operating system layers.

Key Features

Android:

- Support for multi-touch
- Built-in Flash support
- Small weight database for storage
- Connectivity where commonly used connectivity standards such as GSM/EDGE, Evolution-Data, Optimized EV-DO, Universal Mobile, Telecommunication System, Bluetooth, WiFi, IDEN, CDMA, LTE, and WiMAX
- File support for different types of files such as MP3, MIDI, WAV, JPEG, PNG, GIF, and BMP.
- Supports different sensor facilities such as Accelerometer Sensor, Proximity Sensor, Camera, Digital Compass, and GPS
- Intuitive UI where the ui is easily understandable by Users, with support of multiple apps
- Wireless connectivity options where internet can be shared with different devices

Symbian:

- Real-time, multiple threads executed simultaneously in a kernel
- Support for audio, recording of video, playback and streaming, and photograph conversions
- Fully data-oriented and component-based operating system
- protection mechanisms towards malware
- Support for international encoding standards with scripts as Unicode
- Support for simple and
- high possible inter-process communication where function additionally eases the porting of code written for other structures to the Symbian OS

Pros and Cons

Based on Linux, Android is the new mobile operating system, much more flexible than the latter one, even managed to run on Tablets too. Open to code and re-designed and is now the best selling smartphone platform world-wide. Successor to Symbian OS, developed by Symbian Ltd, but now, Accenture maintains it. Mobile Operating system specially designed for smartphones.

Android Pros:

- lots of apps and widgets. Very highly customizable to look and feel the way you want it to.
- The UI is fluid and quite obvious.
- The OS is basically a platform with each application sandboxed from each other. When an app crashes, the phone does not necessarily crash unless you are using a very resource intensive app like a PSX emulator. You can force close the crashed app (like the end task option in Windows Task Manager).

Android Cons:

- No proper task switching or managing. It only keeps a certain number of apps running and closes apps automatically when you are reaching full RAM usage. This may be nice in theory since you don't get the memory full message, but you have no control about which app it closes. Sometimes it's very annoying when it closes the app you actually need to be running at the time. It does cache the app to start again from where it last was, but it doesn't happen to all apps and sometimes having to go through loading processes again is annoying. Symbian allows more control on what apps get closed and what are left running.

- Its more prone to malware and spyware due to its high level of being open source. Unlike Symbian which has certificate management, Android does not, so its best to have an Antivirus on it. Malware even found their way into Android Market
- Fragmentation. There are so many different droids with differing chipsets and hardware that not all apps would work the same way on one droid as on another. For example, Falling Fred for Android works well on a Galaxy S, but on an HTC Evo, there are some lags, despite the Evo having way better hardware than the SGS.
- Has practically no advantages to Symbian when you will be using it offline apart from games. Yes the apps you will use are different and may feel better on Android, but in terms of practicality, almost anything you can do on Android you can do on Symbian if both are offline. Android only wins big if you have a data plan to make use of its full potential (or maybe wifi everywhere you go?)
- Connectivity limitations. Android cannot create or connect to adhoc networks (unless you root it). Also their PC connectivity solutions/software are absolutely awful (or if they work good, they are worse in terms of functionality and capability than Ovi Suite, let alone PC Suite).

Symbian Pros:

- Proper multitasking control. As said before, you control what gets closed and what stays open
- Proper PC integration. We all know how bad Ovi Suite is compared to PC Suite, but nothing Android has can even compare to Ovi Suite in terms of functionality.
- Symbian phones generally have better hardware components. Speaking for Nokia vs Samsung alone, the GPS antenna, AGPS computations, 2g/3g radios, wifi receivers and transmitters, audio quality, durability feel and quality, of Nokia phones are superb compared to almost any Android phone in the market today.
- Ovi Maps - still better than anything Android can throw. I cant use Google Navigation in my country, but the rest all fail in comparison. Android doesnt even natively support the use of bluetooth GPS devices, you need to buy/download a Mapping System that does.

Also, my 5800 running on Integrated GPS alone can get a more accurate fix than my SGSL on integrated GPS. It takes the same time for both to get a full fix too. If I use AGPS on the Nokia, it gets fix in 2seconds. Samsung Fail yes (bad GPS antenna), but partly Android too due to bad AGPS data calculation scripting.

Symbian Cons:

- App developers are leaving, so you have less apps and games to choose from. These is what is now starting to give Android more functionality than symbian. There's usually an app for whatever you may want to do.
- When an app crashes, the whole OS can crash, depending on the app, especially if its an integrated app.
- While many people customize Symbian (regional devs, operator devs), Symbian editing often causes problems, while Android is made to be customizable, reducing probable software errors which may ruin the phone up by devs and operators.

Challenges

Android:

1. Hardware Features- The Android OS is an open source system. Alphabet gives manufacturers the leeway to customize the operating system to their specific needs. Also, there are no regulations on the devices being released by the different manufacturers. As a result, you can find various Android devices with different hardware features running on the same Android version. Two smartphones running on Android latest ver, for example, may have different screen resolutions, camera, screen size, and other hardware structures. During android app development, developers need to account for all of this to ensure the application delivers a personalized experience to each user.
2. Lack of Uniform User Interface Design Rules- Since Google is yet to release any standard UI (user interface) design rules or process for mobile app developers, most developers don't follow any standard UI development rules or procedure. Because developers are creating custom UI interfaces in their preferred way, a lot of apps tend to function or look different across different devices. This diversity and incompatibility of the UI usually affects the user experience that the Android app directly delivers. Smart developers prefer to go for a responsive layout that'll keep the UI consistent across different devices. Moreover, developers need to test the UI of the app extensively by combining emulators and real mobile devices. Designing a UI that makes the app deliver the same user experience across varying Android devices is one of the more daunting challenges developers face.
3. API Incompatibility- A lot of developers make use of third-party APIs to enhance the functionality and interoperability of a mobile device. Unfortunately, not all third-party APIs available for Android app development are of high quality. Some APIs were created for a particular Android version and will not work on devices running on a different version of the operating system. Developers usually have to come up with ways to make a single API work on all Android versions, a task they often find to be very challenging
4. Security Flaws- As previously mentioned, Android is an open source software, and because of that, manufacturers find it easy to customize Android to their desired specifications. However, this openness and the massive market size makes Android a frequent target for security attacks. There have been several instances where the security of millions of Android mobile devices have been affected by security flaws and bugs like mRST, Stagefright, FakeID, 'Certifi-gate,' TowelRoot and Installer Hijacking. Developers need to include robust security features in their applications and utilize the latest encryption mechanisms to keep user information secure and out of the hands of hackers.

Symbian:

1. Much of the core of Symbian came from EPOC, which was developed by Psion for running 8-bit and 16-bit PDAs in the early 1990s. The same core underwent several incarnations till it became Symbian. There was a lot of clunky legacy stuff in there that had to be tweaked to handle faster processors, more RAM and storage.
2. The programming platform and the API were terribly hard compared to the modern mobile programming options such as Swift or Android Java. It was designed to be like that. Symbian C++ had its own extensions such as Cleanup Stack (a quaint way to provide garbage collection), Descriptors (strings, with some memory safety). It felt like a totally different language and was somewhat hard to master. This extended all the way up to the UI. There was no simpler, non-C++ way to create a simple screen.
3. The UI was fragmented. Symbian didn't have its own UI. The licensees like Nokia and Sony Ericsson had their own flavor of UI. Nokia had S60, SE used UIQ and there were a couple of Japanese UI flavors as well. The upshot of this was one could not write portable apps that work on both a Nokia and SE phone. It was non trivial to create a simple app.
4. Also, the Symbian OS is subject to a variety of viruses, the best known of which is Cabir. Usually these send themselves from phone to phone by Bluetooth. So far, none have taken advantage of any flaws in Symbian OS – instead, they have all asked the user whether they would like to install the software, with somewhat prominent warnings that it can't be trusted, although some rely on social engineering, often in the form of messages that come with the malware, purporting to be a utility, game or some other application for Symbian.

Applications

Android OS:

For Android the application store is Google play. Some applications are:

- Communication Application: mobile operating systems offer is the ability to connect to the internet via the smartphone's built-in modem and a wireless service provider
- Internet Application: Many mobile OSes offer a native web browser application, which allows users to search the internet and visit webpages.
- Business Application: User can use their os to grow their business virtually
- Multimedia Application: Users can view various different multimedia files like video, audio, text, etc
- Security Application: They have native GPS (global positioning system) applications that allow users to search for locations, follow step-by-step directions and, in some cases, share location with different devices
- Fun/Entertainment Application
- Gaming applications

Symbian OS:

For Symbian the application store is Nokia Ovi Store. Few applications are:

- Browser access: Symbian was the first mobile platform to make use of their own built-in WebKit based browser
- GPS Application and Software Testing.
- Messaging Application- SMS / MMS applications
- IrDa and Bluetooth-based Application for Symbian enable Mobile Devices.
- Symbian Client/Server Architecture
- Telephony and PC connectivity
- Streaming based applications
- SIP protocol based applications
- Symbian touch UserInterface

Comparative Study

Android and Symbian are two killer mobile operating systems. Symbian, once was a ruling the cell phone market. But now, Android has replaced it. Ever imagined, why the Symbian, which used to rule the cell phone market since ages, was easily outranked by the new Android mobile operating system in no time.
Successor to Symbian OS, developed by Symbian Ltd, but now, Accenture maintains it. Mobile Operating system specially designed for smartphones.
Based on Linux, Android is the new mobile operating system, much flexible than the latter one, even managed to run on Tablets too. Open to code and re-designed and is now the best selling smartphone platform world-wide.

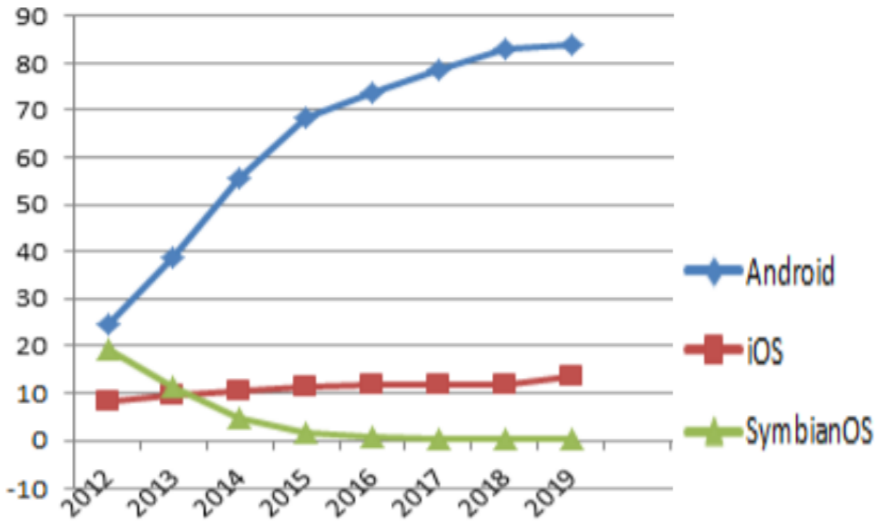
Comparison Based on Different Parameters

PARAMETERS	SYMBIAN	ANDROID
OS family	RTOS	Linux
Vendor	Symbian Ltd. And Symbian foundation	Open Handset Alliance, Google
Official Site	http://licensing.symbian.org/	www.android.com
Developed in Programming language	C,C++,ME, Python, Ruby, Flash Lite	C,C++,Java
App Store	Nokia Ove Store	Google Play
License	Proprietary	Open source
Battery Demand	Less	Highest
Security	Hard to crack	Softest to crack
Voice Assistant	Vlingo 3.2	Google now
Sideloading	Available	Available
Environment	QT, Carbide, Vistamax, Eclipse	Eclipse(Google)
CPU Architecture	ARM, x86	ARM, x86, MIPS

The following is the information about sales in different years with the following Android, iOS, and SymbianOS.

Date	Android	iOS	SymbianOS
2012	24.77	8.05	19.08
2013	38.64	9.33	11.1
2014	55.41	10.6	4.75
2015	68.34	11.37	1.76
2016	73.54	11.83	0.72
2017	78.51	11.66	0.27
2018	83.15	11.6	0.15
2019	84.02	13.33	0.05

Market Share



References

1. A. K. Maji, K. Hao, S. Sultana and S. Bagchi, "Characterizing Failures in Mobile OSes: A Case Study with Android and Symbian," *2010 IEEE 21st International Symposium on Software Reliability Engineering*, San Jose, CA, USA, 2010, pp. 249-258, doi: 10.1109/ISSRE.2010.45.
2. P. Kaur and S. Sharma, "Google Android a mobile platform: A review," *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, Chandigarh, India, 2014, pp. 1-5, doi: 10.1109/RAECS.2014.6799598.
3. Sharma, T. N., Mahender Kr Beniwal, and Arpita Sharma. "Comparative study of different mobile operating systems." *International Journal of Advancements in Research & Technology* 2.3 (2013): 1-5.
4. Malhotra, N. S. (2014). A Comparative study between the android and symbian operating systems. International Journal of Engineering and Technical Research (IJETR), 2(1), 38-43.
5. Joseph, J., & Shinto Kurian, K. (2013). Mobile OS–Comparative Study. Journal of Engineering Computers & Applied Sciences, 2(10), 10-19