

Text Book Of

Android Programming

For

B.C.A. Science : Semester - VI

As Per New Syllabus

Kamil Ajmal Khan

M.Sc. Computer Science

Assistant Professor, Computer application Department

Abeda Inamdar Senior College Pune

Dr. Ms. Manisha Bharambe

M.Sc (Comp. Sci.), M.Phil, Ph.D (Comp. Sci.)

Associate Professor, Computer Science Department

MES Abasaheb Garware College, Pune

Price ₹ 200.00

First Edition : February 2019**© : Authors**

The text of this publication, or any part thereof, should not be reproduced or transmitted in any form or stored in any computer storage system or device for distribution including photocopy, recording, taping or information retrieval system or reproduced on any disc, tape, perforated media or other information storage device etc., without the written permission of Authors with whom the rights are reserved. Breach of this condition is liable for legal action.

Every effort has been made to avoid errors or omissions in this publication. In spite of this, errors may have crept in. Any mistake, error or discrepancy so noted and shall be brought to our notice shall be taken care of in the next edition. It is notified that neither the publisher nor the authors or seller shall be responsible for any damage or loss of action to any one, of any kind, in any manner, therefrom.

Published By :**NIRALI PRAKASHAN**

Abhyudaya Pragati, 1312, Shivaji Nagar

Off J.M. Road, PUNE – 411005

Tel - (020) 25512336/37/39, Fax - (020) 25511379

Email : niralipune@pragationline.com**P.O.D.****STAR COPIERS PVT. LTD.**

Kumthekar Road, Sadashiv Peth

Pune - 411 030

Tel - (020) 24479201

➤ DISTRIBUTION CENTRES**PUNE**

Nirali Prakashan : 119, Budhwar Peth, Jogeshwari Mandir Lane, Pune 411002, Maharashtra
(For orders within Pune) Tel : (020) 2445 2044, 66022708, Fax : (020) 2445 1538; Mobile : 9657703145
Email : niralilocal@pragationline.com

Nirali Prakashan : S. No. 28/27, Dhayari, Near Asian College Pune 411041
(For orders outside Pune) Tel : (020) 24690204 Fax : (020) 24690316; Mobile : 9657703143
Email : bookorder@pragationline.com

MUMBAI

Nirali Prakashan : 385, S.V.P. Road, Rasdhara Co-op. Hsg. Society Ltd.,
Girgaum, Mumbai 400004, Maharashtra; Mobile : 9320129587
Tel : (022) 2385 6339 / 2386 9976, Fax : (022) 2386 9976
Email : niralimumbai@pragationline.com

➤ DISTRIBUTION BRANCHES**JALGAON**

Nirali Prakashan : 34, V. V. Golani Market, Navi Peth, Jalgaon 425001, Maharashtra,
Tel : (0257) 222 0395, Mob : 94234 91860; Email : niralijalgaon@pragationline.com

KOLHAPUR

Nirali Prakashan : New Mahadvar Road, Kedar Plaza, 1st Floor Opp. IDBI Bank, Kolhapur 416 012
Maharashtra. Mob : 9850046155; Email : niralikolhapur@pragationline.com

NAGPUR

Nirali Prakashan : Above Maratha Mandir, Shop No. 3, First Floor,
Rani Jhanshi Square, Sitabuldi, Nagpur 440012, Maharashtra
Tel : (0712) 254 7129; Email : niralinagpur@pragationline.com

DELHI

Nirali Prakashan : 4593/15, Basement, Agarwal Lane, Ansari Road, Daryaganj
Near Times of India Building, New Delhi 110002 Mob : 08505972553
Email : niralidelhi@pragationline.com

BENGALURU

Nirali Prakashan : Maitri Ground Floor, Jaya Apartments, No. 99, 6th Cross, 6th Main,
Malleswaram, Bengaluru 560003, Karnataka; Mob : 9449043034
Email : niralibangalore@pragationline.com

Other Branches : Hyderabad, Chennai

Note : Every possible effort has been made to avoid errors or omissions in this book. In spite this, errors may have crept in. Any type of error or mistake so noted, and shall be brought to our notice, shall be taken care of in the next edition. It is notified that neither the publisher, nor the author or book seller shall be responsible for any damage or loss of action to any one of any kind, in any manner, therefrom. The reader must cross check all the facts and contents with original Government notification or publications.

niralipune@pragationline.com | www.pragationline.com**Also find us on  www.facebook.com/niralibooks**

Syllabus...

- 1. Introduction to Android** (06 L)
 - 1.1 Overview
 - 1.2 History
 - 1.3 Features of Android
 - 1.4 Architecture of Android
 - Overview of Stack
 - Linux Kernel
 - Native Libraries
 - Android Runtime
 - Application Framework
 - Applications
 - 1.5 SDK Overview
 - Platforms
 - Tools – (JDK, SDK, Eclipse/Android Studio, ADT, AVD, Android Emulator)
 - Versions
 - 1.6 Creating your first Android Application
- 2. Activities, Fragments and Intents** (09 L)
 - 2.1 Introduction to Activities
 - 2.2 Activity Lifecycle
 - 2.3 Introduction to Intents
 - 2.4 Linking Activities using Intents
 - 2.5 Calling built-in applications using Intents
 - 2.6 Introduction to Fragments
 - 2.7 Adding Fragments Dynamically
 - 2.8 Lifecycle of Fragment
 - 2.9 Interaction between Fragments
- 3. Android User Interface** (10 L)
 - 3.1 Understanding the Components of a Screen
 - Views and ViewGroups
 - LinearLayout
 - AbsoluteLayout
 - TableLayout
 - RelativeLayout
 - FrameLayout
 - ScrollLayout
 - ScrollView
 - 3.2 Adapting to Display Orientation
 - Anchoring Views
 - Resizing and Repositioning
 - 3.3 Managing Changes to Screen Orientation
 - Persisting State Information during changes in Configuration
 - Detecting Orientation changes
 - Controlling the Orientation of the Activity
 - 3.4 Utilizing the Action Bar
 - Adding Action Items to the Action Bar
 - Customizing the Action Items and Application Icon

4. Designing Your User Interface with Views	(10 L)
4.1 Using Basic Views	
• TextView	
• Button, ImageButton, EditText, CheckBox,	
• ToggleButton, RadioButton, and RadioGroup Views	
• ProgressBar View	
• AutoCompleteTextView View	
4.2 Using Picker Views	
• TimePicker View	
• DatePicker View	
4.3 Using List Views to Display Long Lists	
• ListView View	
• Using the Spinner View	
4.4 Understanding Specialized Fragments	
• Using a ListFragment	
• Using a DialogFragment	
• Using a PreferenceFragment	
5. Displaying Pictures and Menus	(05 L)
5.1 Using Image Views to Display Pictures	
• Gallery and ImageView Views	
• Image Switcher	
• Grid View	
5.2 Using Menus with Views	
• Creating the Helper Methods	
• Options Menu	
• Context Menu	
6. Databases - SQLite	(06 L)
6.1 Introduction to SQLite	
6.2 SQLiteOpenHelper and SQLiteDatabase	
6.3 Creating and Using Database	
6.4 Working with Cursors	
6.5 Building and executing queries	
7. Messaging and E-mail	(06 L)
7.1 SMS Messaging	
• Sending SMS Messages Programmatically	
• Getting Feedback after Sending a Message	
• Sending SMS Messages Using Intent	
• Receiving SMS Messages	
• Caveats and Warnings	
7.2 Sending E-mail	
8. Location-Based Services and Google Map	(08 L)
8.1 Display Google Maps	
• Creating the Project	
• Obtaining the Maps API Key	
• Displaying the Map	
• Displaying the Zoom Control	
• Changing Views	
• Navigating to a Specific Location	
• Adding Markers	
• Getting the Location that was Touched	
• Geocoding and Reverse Geocoding	
8.2 Getting Location Data	
8.3 Monitoring a Location	

Contents...

1. Introduction to Android	1.1 – 1.25
2. Activities, Fragments and Intents	2.1 – 2.31
3. Android User Interface	3.1 – 3.44
4. Designing Your User Interface with Views	4.1 – 4.32
5. Displaying Pictures and Menus	5.1 – 5.24
6. Databases - SQLite	6.1 – 6.24
7. Messaging and E-mail	7.1 – 7.22
8. Location-Based Services and Google Map	8.1 – 8.20



Introduction to Android

Contents...

- 1.1 Overview
- 1.2 History
- 1.3 Features of Android
- 1.4 Architecture of Android
 - 1.4.1 Overview of Stack
 - 1.4.2 Linux Kernel
 - 1.4.3 Native Libraries
 - 1.4.4 Android Runtime
 - 1.4.5 Application Framework
 - 1.4.6 Applications
- 1.5 SDK Overview
 - 1.5.1 Android Platforms
 - 1.5.2 Tools – (JDK, SDK, Eclipse/Android Studio, ADT, AVD, Android Emulator)
 - 1.5.2.1 JDK (Java Development Kit)
 - 1.5.2.2 SDK (Software Development Kit)
 - 1.5.2.3 Android Studio
 - 1.5.2.4 ADT (Android Development Tools)
 - 1.5.2.5 AVD (Android Virtual Device)
 - 1.5.2.6 Android Emulator
 - 1.5.3 Versions of Android
- 1.6 Creating your first Android Application
 - ❖ Summary
 - ❖ Practice Questions

Objectives...

After reading this chapter you will be able,

- To learn Basic Concepts of Android Operating System
- To understand History, Versions and Architecture of Android Operating System
- To study JDK, SDK, ADT, Emulator in Android
- To study Android Studio with its Screen Elements
- To develop Android Applications using Android Studio

1.1 OVERVIEW

- Technology begets more technology, and new technologies proliferate with ever-increasing speed. Some of these new technologies will survive beyond a few years, but most will not.
- There is little worse than investing time and energy in acquiring a new skill that is obsolete on arrival or whose utility is short-lived. In this era, most of the people want to use smart devices for communication, planning and organizing their schedule for their private and professional life.

- These technologies are causing dramatic changes in the organization of information system. Android has been changed in smart device market. Android is a new generation mobile OS which runs on Linux kernel.
- Android device application developed is based on Java Programming. These codes are used to control smart device via Google-enabled java libraries. It is an important platform to develop mobile device application using software stack provided in the Google Android SDK.
- Android combines Operating System features like structured shared memory, preemptive multi-tasking, Unix User Identifiers (UIDs) and file permission with Java language and its class library. The Security platform is much better than J2ME or Blackberry Platforms.
- Programs can typically neither read nor write each other's code. The software developers at mobile development India have expertise in developing application based on Android java libraries.
- Android is an open-source operating system named Android Inc. Google has made the code for all the low-level "stuff" as well as the needed middleware to power and use an electronic device, and gave Android freely to anyone who wants to write code and build the operating system from it. There is even a full application framework included, so third-party apps can be built and installed, then made available for the user to run as they like.

1.2 HISTORY

- Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touch screen mobile devices such as smart phones and tablets.
- In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.
- Initially developed by Andy Rubin founded Android Incorporation in Palo Alto, California, United States in October, 2003, which Google bought in 2005, and Android was unveiled in 2007, with the first commercial Android device launched in September 2008.
- The version history of the Android mobile began with the public release of the Android beta on November 5, 2007. The first commercial version, Android 1.0, was released on September 23, 2008. Android is continually developed by Google and the Open Handset Alliance (OHA) and it has seen a number of updates to its base operating system since the initial release. OHA is a business alliance that consist of 47 companies like Google, HTC, LG Electronics, Samsung Electronics etc. for developing standard platforms for mobile technology.
- Versions 1.0 and 1.1 were not released under specific code names, although Android 1.1 was unofficially known as Petit Four. Android code names are confectionery-themed and have been in alphabetical order since 2009's Android 1.5 Cupcake, with the most recent major version being Android 8.1 Oreo, released in December 2017.
- The operating system has since gone through multiple major releases, with the current version being 9.0 "Pie", released in August 2018. The core Android source code is known as Android Open Source Project (AOSP), and is primarily licensed under the Apache License.

1.3 FEATURES OF ANDROID

- Various features of Android are:
 1. **Application Framework:** It enables reuse and replacements of components.
 2. **Dalvik Virtual Machine:** It is optimized for mobile device.
 3. **Integrated Browser:** It is based on open source web kit engine.
 4. **Handset layouts: Optimized Graphics:** It is powered by a custom 2D graphics Library 3D Graphics based on the OpenGL ES 1.0 Specification.
 5. **Storage:** SQLite, a lightweight relational database.
 6. **Media Support:** Android supports the following audio/video/still media formats like H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, BMP.
 7. **Connectivity:** GSM/EDGE, IDEN, CDMA, EVDO, UMTS, Bluetooth, WiFi.
 8. **Messaging:** SMS and MMS are available forms of messaging also supports Android Cloud to Device Messaging Framework (C2DM).
 9. **Multiple Language Support:** Multiple Language Available in Android.
 10. **Additional Hardware Support:** Android can use video/still cameras, touch screens, GPS, accelerometers, gyroscopes, magnetometers.
 11. **Multi-touch:** Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
 12. **Multitasking:** Multitasking of applications is available.
 13. **Voice Based Features:** Google search through voice has been available since initial release, voice actions for calling, texting, navigation etc.
 14. **Android Market (Google Play):** An online software store developed by Google for Android devices allows users to browse and download apps published by third party developers, hosted on Android Market.
 15. **Streaming Media Support:** RTP/RTSP streaming (3GPP PSS, ISMA), HTML progressive download, Adobe Flash Streaming (RTP) and HTTP Dynamic Streaming are supported by the Flash 10.1 plugin.

1.4 ARCHITECTURE OF ANDROID

- Android architecture is structured in the form of a software stack of components such as Linux Kernel, Libraries, Android Runtime Application Framework and Applications. These five components assist with the operation and functionality of the Android architecture as a whole.

1.4.1 Overview of Stack

- Android operating system is a stack of software components. Main components of Android Operating system Architecture or Software Stack are Linux Kernel, Libraries, Android Runtime, Application Framework and Applications.
- On top of the Linux kernel, there are the middleware, libraries and APIs written in C/C++ and application software running on an application framework which includes Java-compatible libraries. Development of the Linux kernel continues independently of Android's other source code projects.

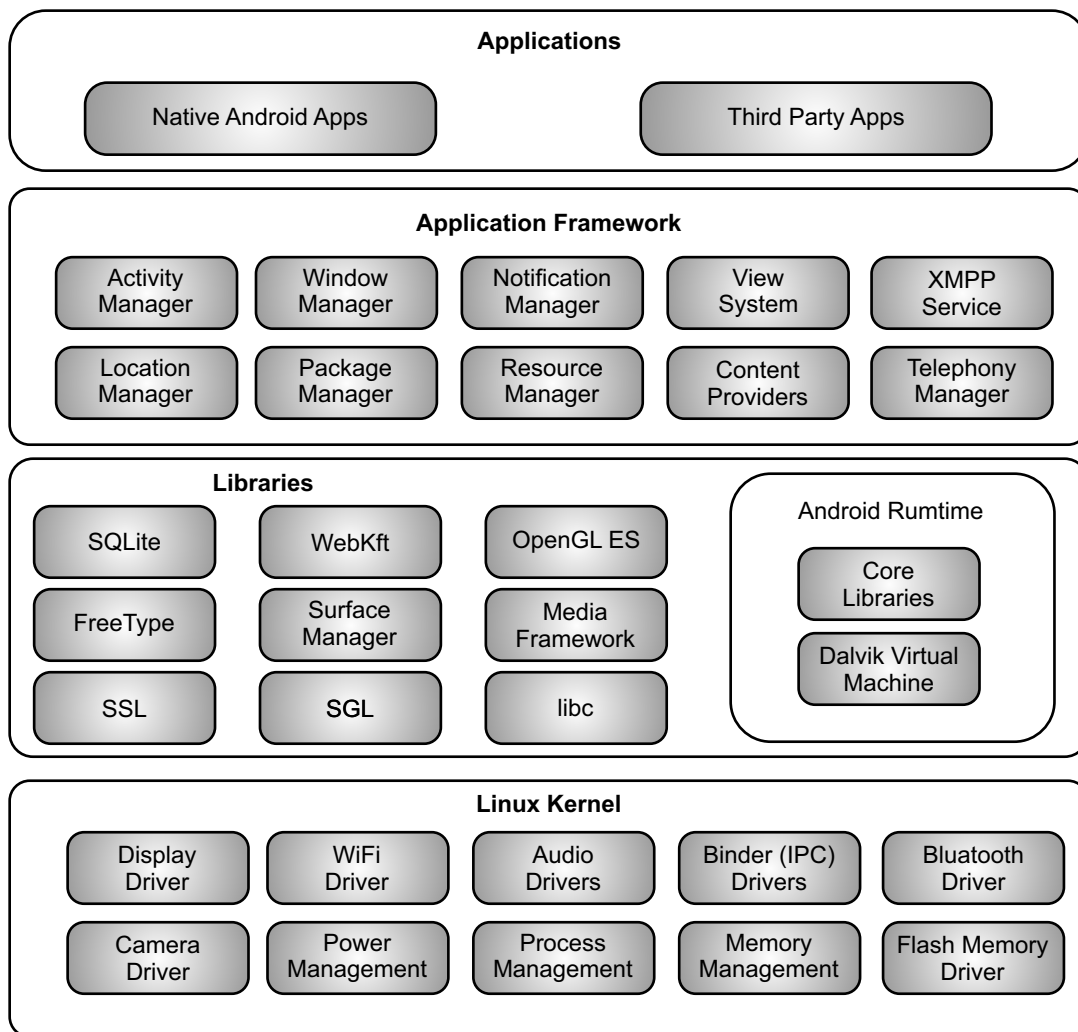


Fig. 1.1: Architecture of Android (Layers in Android OS)

- An Android OS can be referred as a software stack of different layers, where every layer is a group of several program components.
- The Android OS is roughly divided into five sections in four main layers:
 - **Layer 1: Linux Kernel:** Linux Kernel is the first layer, residing at the bottom of the architecture and considered as to be the core layer. Linux Kernel is the kernel on which Android is based. This layer contains all the low level device drivers for the various hardware components of an Android device.
 - **Layer 2: Libraries:** This layer includes all the native libraries of Android. In this layer, native libraries are developed/written in C/C++ languages to ensure smooth OS functionality when accessing multiple apps at once. These libraries contain all the code that provides the main features of an Android OS. For example, the SQLite library provides database support so that an application can use it for data storage. The WebKit library provides functionalities for web browsing.

Android Runtime: At the same layer (2nd Layer) as the libraries, the Android runtime provides a set of core libraries that enable developers to write Android apps using the Java programming language. This layer operates its own virtual environment. Dalvik Virtual Machine (DVM), for developing Android Apps.

- **Layer 3: Application Framework:** Exposes the various capabilities of the Android OS to application developers so that they can make use of them in their applications.
- **Layer 4: Applications:** At this top layer, you will find applications that ship with the Android device (such as Phone, Contacts, Browser, etc.), as well as applications that you download and install from the Android Market. Any applications that you write are located at this layer.

1.4.2 Linux Kernel

- Positioned at the bottom of the Android software stack, the Linux Kernel provides a level of abstraction between the device hardware and the upper layers of the Android software stack.
- Android devices use the Linux kernel, but every phone uses their own version of it. Linux kernel maintainers keep everything tidy and available, contributors (like Google) add or alter things to better meet their needs, and the people making the hardware contribute as well, because they need to develop hardware drivers for the parts they're using for the kernel version they are using.
- This is why it takes a while for independent Android developers and hackers to port new versions to older devices and get everything working. Drivers written to work with one version of the kernel for a phone might not work with a different version of software on the same phone.
- And that's important, because one of the kernel's main functions is to control the hardware. It's a whole lot of source code, with more options while building it than you can imagine, but in the end it's just the intermediary between the hardware and the software.
- When software needs the hardware to do anything, it sends a request to the kernel. From the brightness of the screen, to the volume level, to initiating a call through the radio, even what's drawn on the display is ultimately controlled by the kernel.
- Linux kernel is responsible for Device Drivers, Power Management, Memory Management, Device Management and Resource Access.

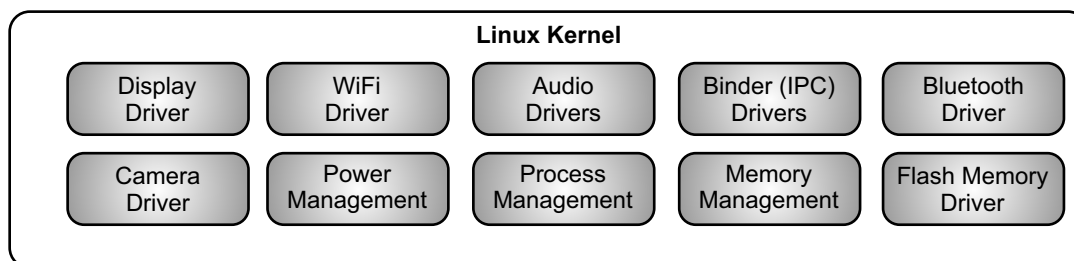


Fig. 1.2: Linux Kernel

1.4.3 Native Libraries

- This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access.
- A summary of some key core Android libraries available to the Android developer is as follows:
 1. **android.app:** Provides access to the application model and is the cornerstone of all Android applications.
 2. **android.content:** Facilitates content access, publishing and messaging between applications and application components.
 3. **android.database:** Used to access data published by content providers and includes SQLite database management classes.
 4. **android.opengl:** A Java interface to the OpenGL ES 3D graphics rendering API.

5. **android.os:** Provides applications with access to standard operating system services including messages, system services and inter-process communication.
6. **android.text:** Used to render and manipulate text on a device display.
7. **android.view:** The fundamental building blocks of application user interfaces.
8. **android.widget:** A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
9. **android.webkit:** A set of classes intended to allow web-browsing capabilities to be built into applications.

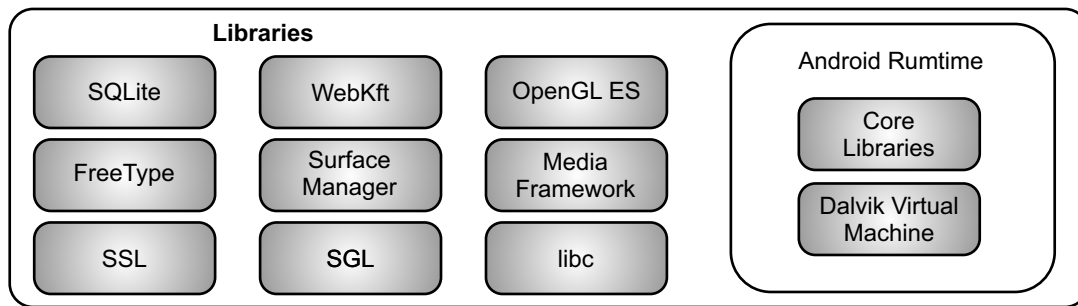


Fig. 1.3: Native Libraries of Android

1.4.4 Android Runtime

- Android Runtime (ART) is the third section of the Android architecture and situated on the second layer (Libraries). It provides a set of Dalvik Virtual Machine (DVM) and Core Libraries that enables developers to write Android applications using Java.
1. **Dalvik Virtual Machine (DVM):**
 - The Android runtime also includes the Dalvik virtual machine, which enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine (Android applications are compiled into the Dalvik executables).
 - Dalvik is a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU.
 2. **Core Libraries:**
 - These are different from Java SE and Java ME libraries. But these libraries provide most of the functionalities defined in the Java SE libraries.
 - (i) Data Structure
 - (ii) File Access
 - (iii) Network Access
 - (iv) Utilities
 - (v) Graphics

1.4.5 Application Framework

- The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.
- The application framework is used by developers for developing applications.
- The application framework is a set of services that collectively form the environment in which Android applications run and are managed. It manages User Interface (UI), application resources, and abstraction for hardware access.