Pázmány Péter Catholic University
Faculty of Information Technology and Bionics

Basics of Mobile Application Development

Introduction, Mobile platforms

Kálmán Tornai
Room #232
tornai.kalman@itk.ppke.hu

Foreword – Examination and evaluation system

- It is mandatory to participate in the lessons
  - The maximum number of allowed absence is 3
- On each lesson a short test-paper
  - Will be rated between 0 and 10.
  - Missing one is rated as 0.
- You must hand in your homework solutions. Homework are given occasionally; the expected number of homework is 8.
  - You must submit the homework in one week after it is handed out
  - The solutions are rated, and the sum of the points is calculated at the end of the semester.
  - Missing homework is 0 points
  - Solution without issues is 25 points

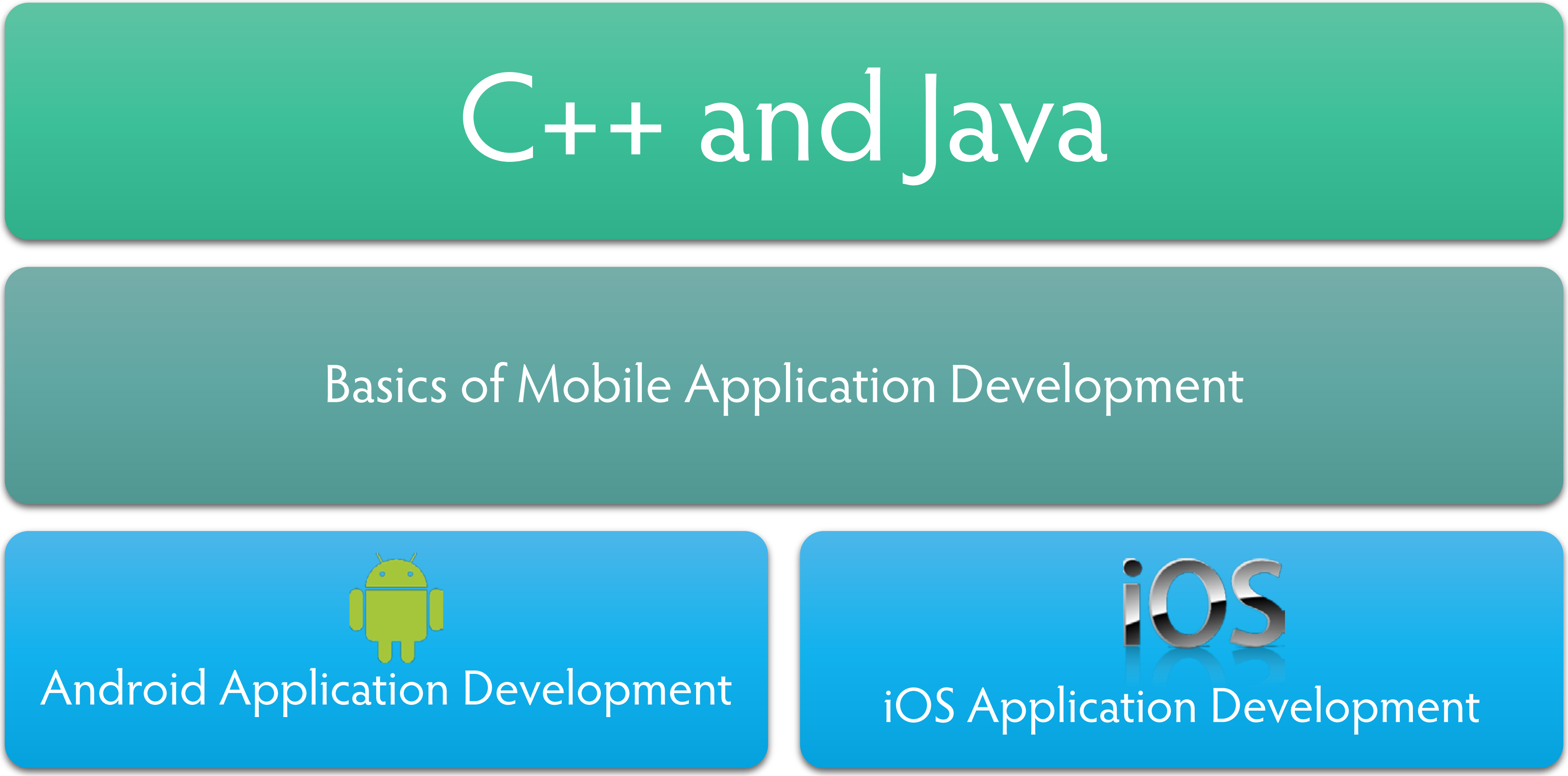
Foreword – Cont'd

- You must take a final exam, at the end of the semester.
  - 300 points
  - One substitution is permitted.
- Final grade is calculated from the sum of test points, homework points and final exam points:
  - [0% 50%): fail
  - [50% 60%): pass
  - [60% 70%): satisfactory
  - [70% 80%): good
  - [80% 100%]: excellent

Important task

- SVN
  - You must fill the following form
    - https://forms.gle/SUxPtKLa3fdZw7FQA
    - It is required in order to create the SVN repository directories, which are going to be used to submit homework!
  - Deadline:
    - 13th September, Friday 11.59 p.m.
  - Repositories will be created on next week
    - You will be notified
- Slack
  - Announcements, information, questions and answers:
    - https://join.slack.com/t/ppkeitk-ajma/shared_invite/enQtNzM0MzMyMjk3Mzc5LTljNzMxOTYzM2EzZTI3NTc0Nzg3ZTAwYTg1YTAyOTE0ZDNlNzBiMzVhY2VkMDkxMTRmMzZiYTdmMmVkYTY4NGU

Foreword – This course in the curriculum



C++ and Java

Basics of Mobile Application Development

Android Application Development

iOS Application Development

Foreword – Schedule of the semester

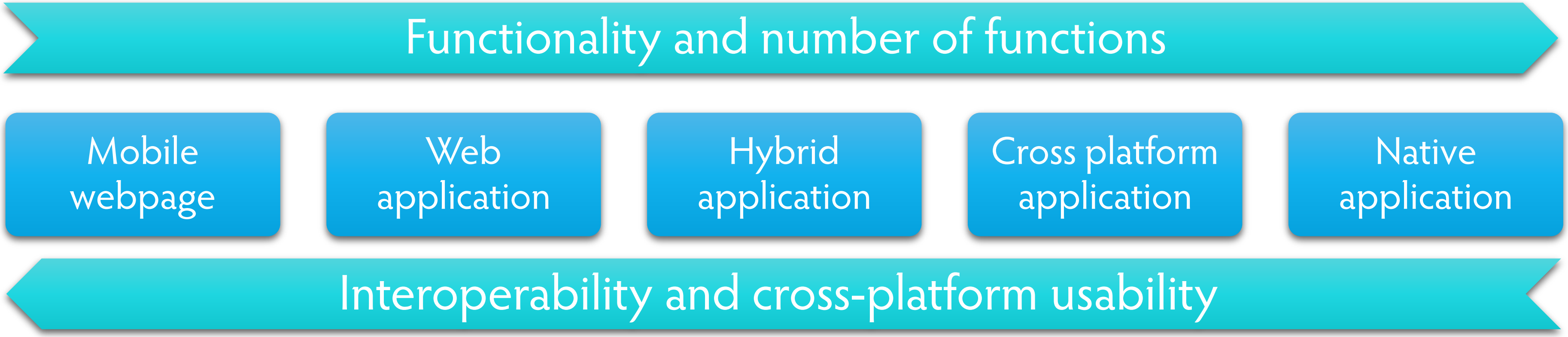
- Introduction
  - Technologies and capabilities of the mobile platforms.
  - Introduction to Android and iOS platform.
- Hardware capabilities of mobile platforms
  - GPS, Bluetooth, NFC, sensors, …
- Review of basic programming techniques and definitions
  - C++, Java, OOP, C++ and Java
- Programming patterns and methods (MVC, MVVM, Delegate, Adapter, Target-Action, TDD)
- SWIFT I & II
- Objective-C
- Kotlin I & II
- Android basics
- iOS basics
- Persistent data storage on iOS and Android, Databases
- Final Exam

What is a mobile application?

What is this all about?

Mobile applications

- What makes an application mobile?
- Further questions:
  - What is a web-based application?
  - What is a native application?
  - What is the better choice for our project?
    - Different aspects …
  - What are the differences?



Functionality and number of functions

Mobile webpage

Web application

Hybrid application

Cross platform application

Native application

Interoperability and cross-platform usability

Advantages and disadvantages

- HTML5
  - Product available instantly on multiple platforms
  - Do not have to be installed, easy to use
  - Web development knowledge can be enough
  - Updates can be spread better
  - Lower costs on development side
  - Faster development and deployment
  - Special expertise on multiple platforms not required

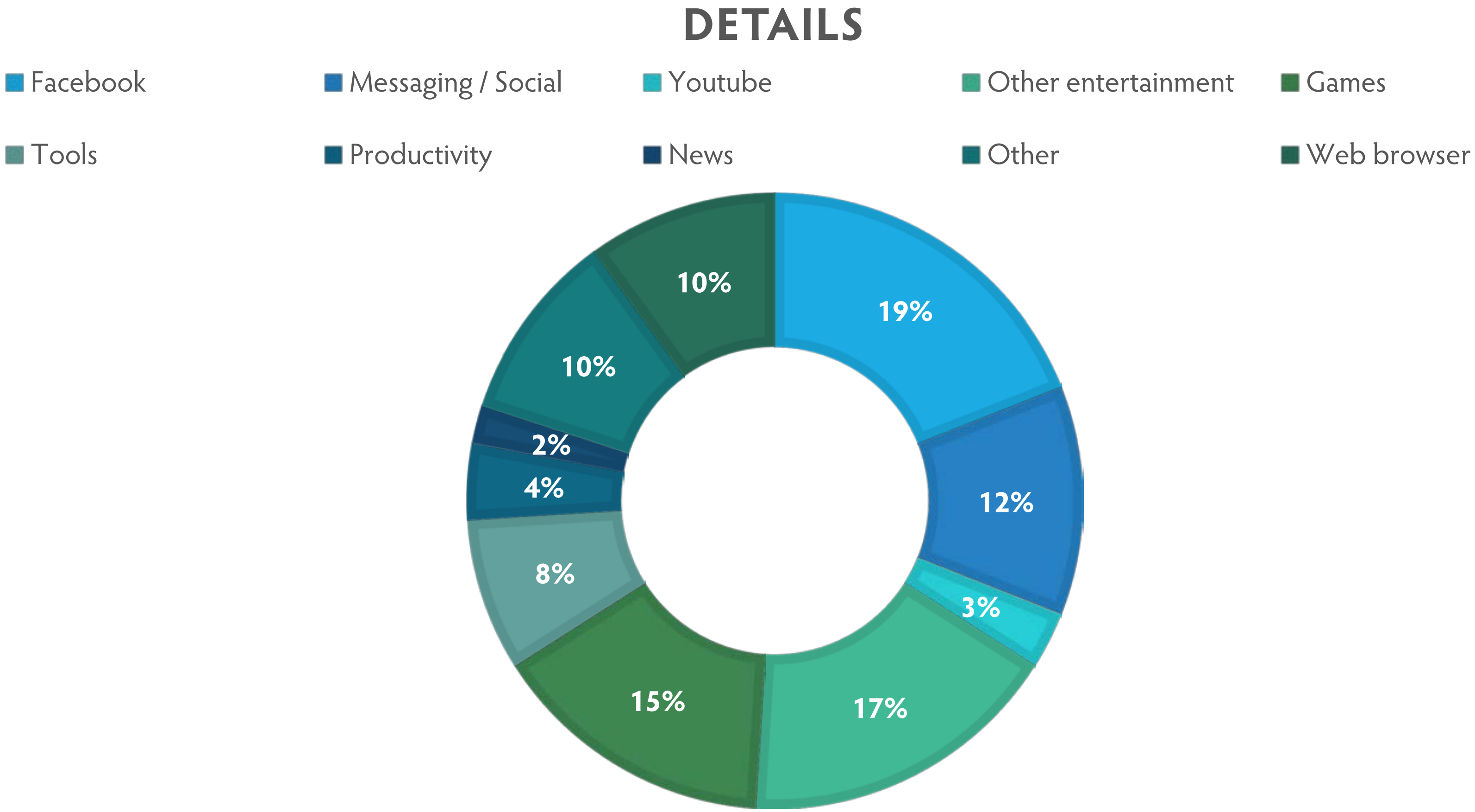
- Native
  - Device-specific capabilities can be utilized
    - Sensors
    - Graphical processor
    - Optimized code
  - Higher performance
  - Higher security level
  - Better user experience can be achieved
  - Lower costs on server/infrastructure side
  - Built-in payment options

Hybrid applications

What to use?

- Depends on the nature of the application
  - Can you imagine a 3D game using HTML5 techniques?
  - Is it really required to implement a tax calculator as a native application?
- Viewpoints
  - Is 3D graphics required?
  - Is camera or sensors of the device required?
    - Location requirement does not imply native applications
  - Is there any internet connection at the location of usage?
  - Computational complexity
    - Wolfram Alpha vs Wolfram Mathematica
  - How many social media API is planned to be used?
- This is a design issue, there may not be a perfect solution
- This course concerns only with native applications

Where are we going?



Platforms

History and overview

IBM Simon – 1992

Microsoft Pocket PC – 2002

Apple Newton – 1993

Google Pixel 3 XL – 2018

Apple iPhone Xr – 2018

In the beginnings …

- 1987/1993 Apple Newton
  - First mobile device, with applications – No mobile connectivity, thus it cannot be considered as a phone
- 1992 – IBM Simon
  - Fax, e-mail, calendar, calculator, clock, notepad, touch screen
- 1996 – Nokia 9000
  - Nokia cell phone + HP PDA
- 1999 – pdQ Smartphone
  - Palm PDA + Internet connection
- 1999 – NTT Docomo
- 1999 – Blackberry 850
- 2000 – Ericsson R380 Smartphone
  - First device which is called „smartphone"
- 2001 – Kyocera 6035
  - Palm PDA + Mobile phone (Verizon)
- 2002 – Microsoft PocketPC (later Windows Mobile)
  - Continuing the Windows CE platform

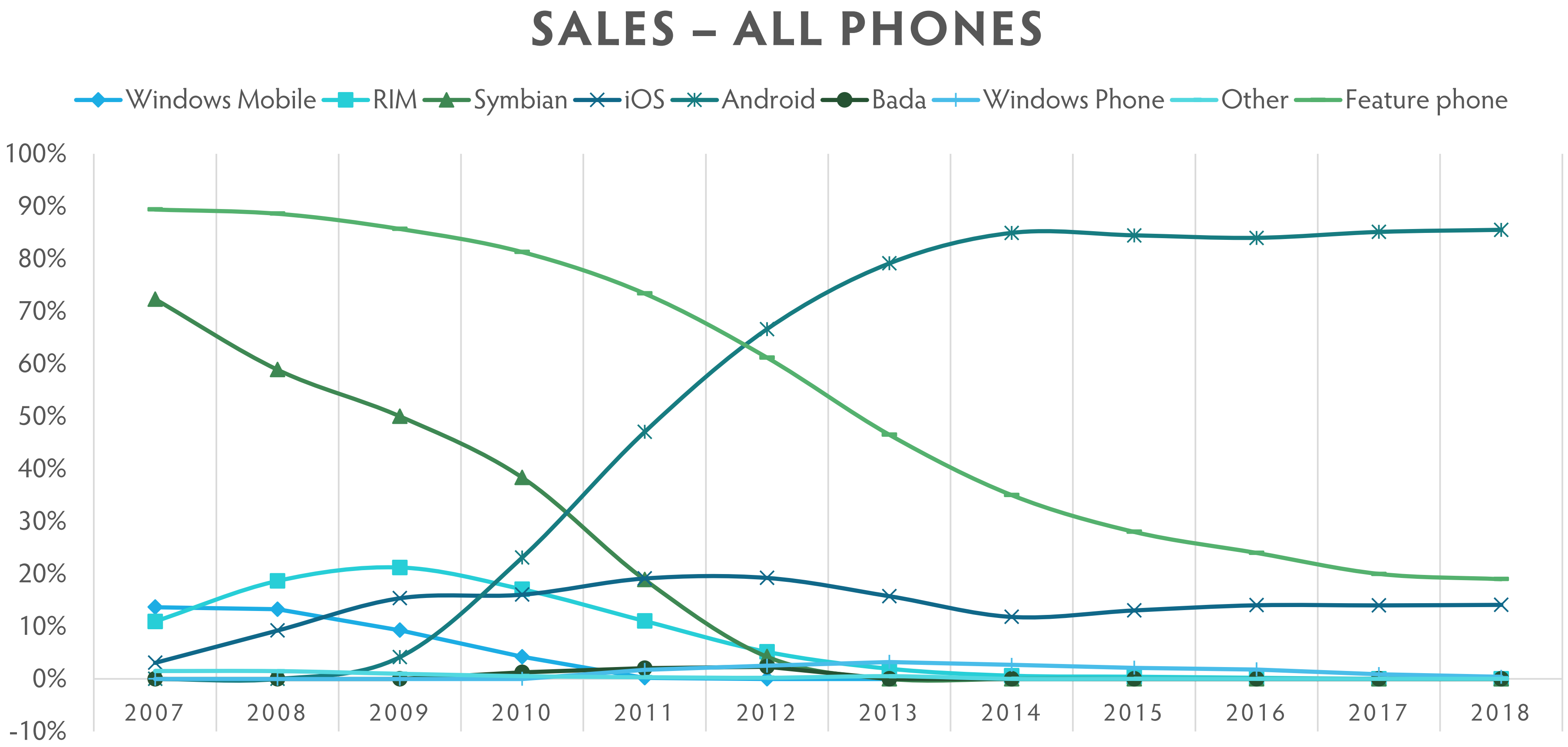
… the boom …

- 2007 – Apple iPhone
    - First multitouch phone (no stylus, keyboard, etc.)
- 2008 – Google Android
    - HTC Dream
- 2008 – Samsung Bada
    - Samsung Wave S8500
    - Migrated into the Tizen plafrom
- 2010 – Windows Phone
- 2011 – Palm Inc became part of HP
- 2011 – Nokia switches to Windows Phone from Symbian
- 2012 – Firefox OS
    - Based on HTML5
- 2012 – Samsung Tizen
- 2013 – Ubuntu Touch
- 2014 – Wearable devices boom

… nowadays …

- 2015 – Windows 10 – multiplatform
- 2015 – Firefox OS extinct …
- 2016 – Cyanogen OS is announced
- 2016 – Cyanogen OS disappears
- 2017 – Android on Nokia phones
- 2017 – Windows 10 Mobile still exists, but …
  - … zombie
- 2019 ??

… statistics


SALES – ALL PHONES
Windows Mobile
RIM
Symbian
iOS
Android
Bada
Windows Phone
Other
Feature phone
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
-10%
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018

This course

- Two main platforms will be overviewed
  - Android
  - iOS
- We are going to make simple applications on that platforms

Android

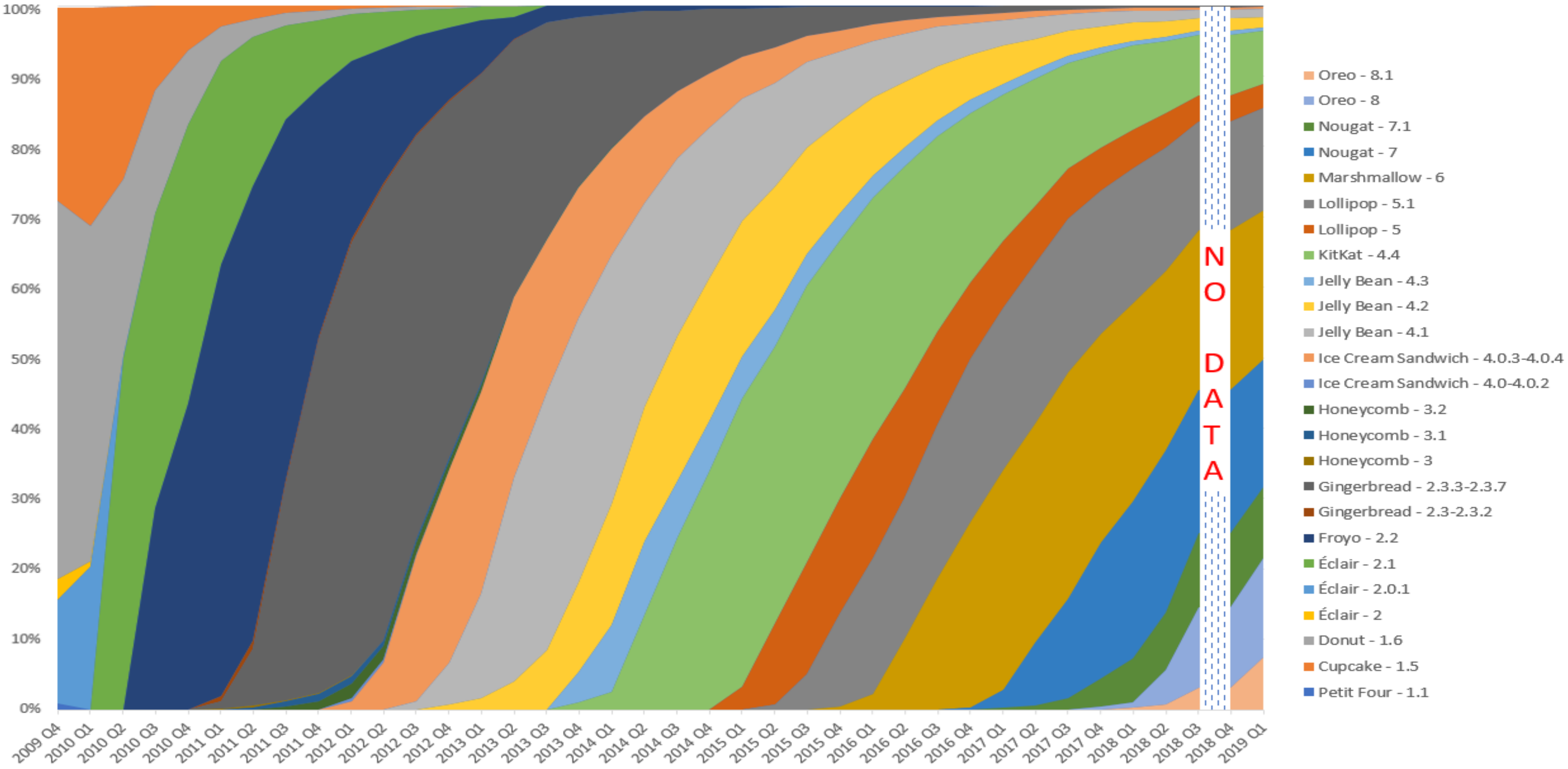
Android properties

- Supervised and developed by Google – software package
  - Linux + Android VM + Other applications
  - Many manufacturer, different hardware
    - Basically ARM and Intel processors
    - Different capabilities
    - Inaccurate implementation of specifications occur
    - Additional software are included, which are not part of the Android system
  - Development tools and emulator
    - Available for all platforms
- Main properties
  - Modular
  - Multitask, automatic memory managements, program libraries included
  - Almost arbitrary mobile communication technology is available (GSM … LTE)
  - Wi-Fi (Client and AP), Ethernet (tethering), Bluetooth, NFC
  - Sensors: GPS, Triaxial accelerometer / magnetometer, thermometer, light sensor
  - Camera support, recording and playback, even stereo
  - HDMI support, accelerated 2D and 3D graphics, parallel computation

Android versions

Introduced in	Version number	Name	API LEVEL
2007	ß		ß
2008	1.0		1
2009	1.1		2
2009	1.5	Cupcake	3
2009	1.6	Donut	4
2009	2.0	Eclair	5
2010	2.2	Froyo	8
2010	2.3	Gingerbread	9
2011	3.0	Honeycomb	11
2011	4.0	Ice Cream Sandwich	14
2013	4.1	Jelly Bean	16
2013	4.4	KitKat	19
2014	5.0	Lollipop	21
2015	6.0	Marshmallow	23
2016	7.X	Nougat	24
2017	8.X	Oreo	26
2018	9.X	Pie	28
2019	10	Android 10	29

Spread of different versions



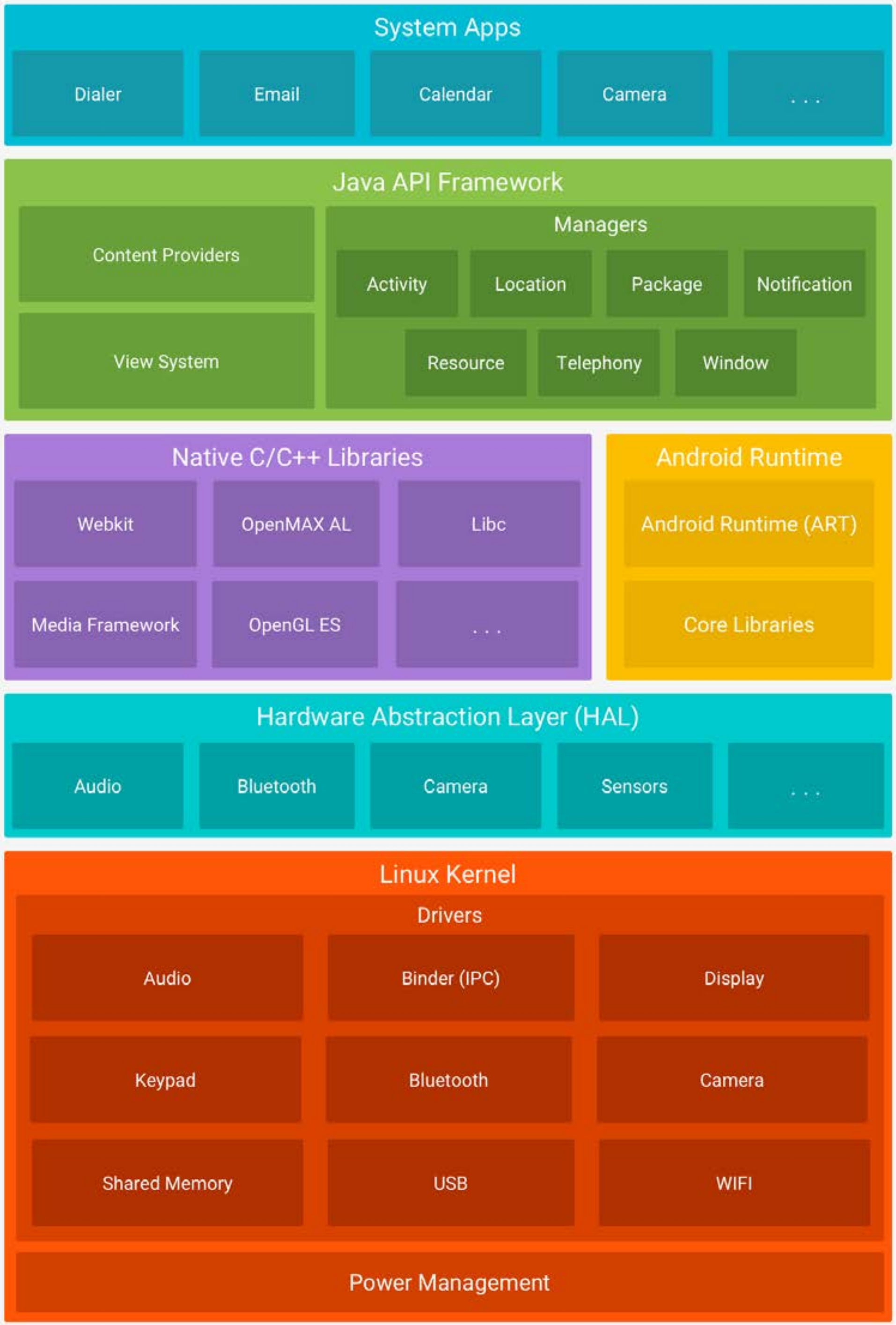
Android platform – details

- Collection of several, different software
  - Middleware
  - Core application
  - Operating system
- Kotlin – programming language
  - On Java virtual machine
  - First appeared in 2011
  - Last stable release: 1.3.50 – August 2019.
  - For Android since 2017 Google I/O
  - Kotlin is designed to be an industrial-strength object-oriented language, and a "better language" than Java, but still be fully interoperable with Java code

Android platform – details

- Java based programming language (not equivalent to Java)
  - Some of the Java packages are excluded
    - java.applet
    - java.beans
    - javax.rmi
    - javax.print
  - Not JVM, but ART (Before Lollipop it was Dalvik VM)
    - Open source
      - Less memory is required compared to other VM-s
      - Multiple VM-s can be executed parallel, without the loss of efficiency
    - Ahead-Of-Time compiling and Just-In-Time compiling
    - *.java → *.class → *.dex →*.apk
      - The executed VM *.dex files, which are generated by compiling Java byte codes

Building blocks



System Apps

Dialer | Email | Calendar | Camera | . . .

Java API Framework

Content Providers

View System

Managers

Activity | Location | Package | Notification

Resource | Telephony | Window

Native C/C++ Libraries

Webkit | OpenMAX AL | Libc

Media Framework | OpenGL ES | . . .

Android Runtime

Android Runtime (ART)

Core Libraries

Hardware Abstraction Layer (HAL)

Audio | Bluetooth | Camera | Sensors | . . .

Linux Kernel

Drivers

Audio | Binder (IPC) | Display

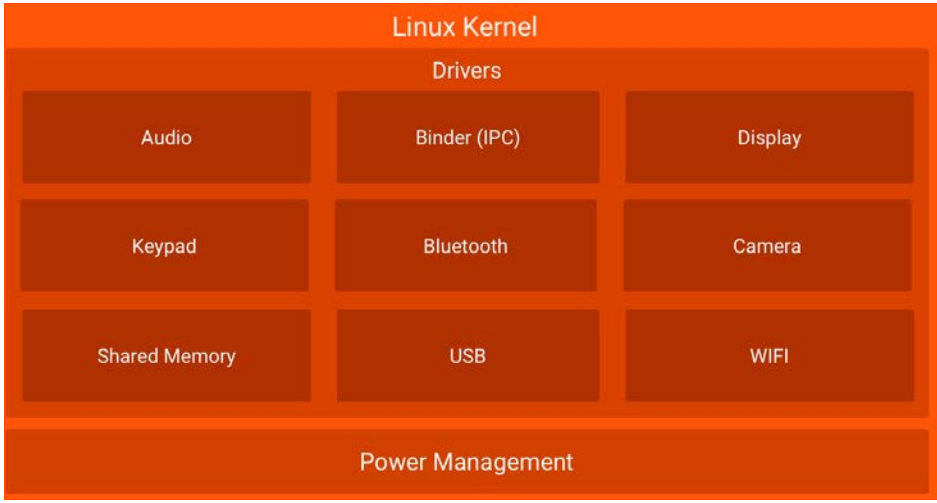
Keypad | Bluetooth | Camera

Shared Memory | USB | WIFI

Power Management

Details

- Kernel
  - Most important component
  - Abstraction level between hardware and software
  - Memory, energy, network management
  - Driver models
  - Linux kernel version:
    - 4.4, 4.9 or 4.14
- Runtime
  - Each android application is a separate VM process in the system
  - ART
    - AOT, JIT, GC, …
  - Core Libraries
    - Remaining Java packages



Linux Kernel

Drivers

Audio

Binder (IPC)

Display

Keypad

Bluetooth

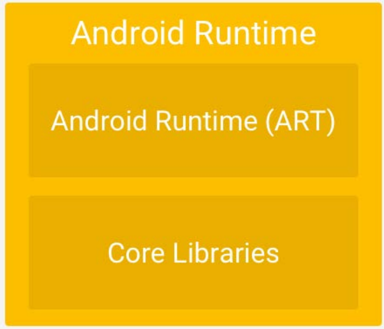
Camera

Shared Memory

USB

WIFI

Power Management



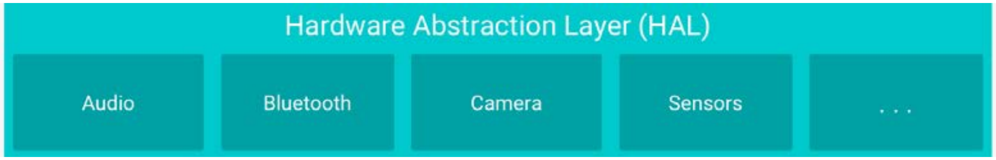
Android Runtime

Android Runtime (ART)

Core Libraries

Details

- HAL
  - A layer over hardware components to hide the differences
  - Code depends on actual device and architecture
    - Drivers



Hardware Abstraction Layer (HAL)

Audio

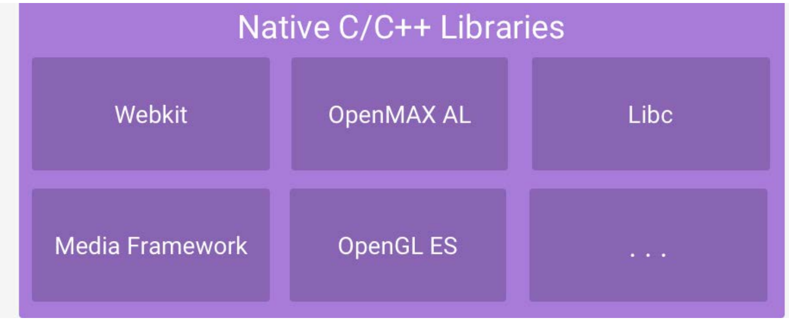
Bluetooth

Camera

Sensors

. . .

- Native C/C++ libraries
  - Developers can access it through the app framework
  - Media Framework:
    - Image/Music/Video can be managed, Media Codecs
  - Webkit
  - OpenGL, OpenAL and OpenMax
    - Graphical library, Sound library, etc.
  - SQLite
    - Relational databases can be used
  - NDK application components



Native C/C++ Libraries

Webkit

OpenMAX AL

Libc

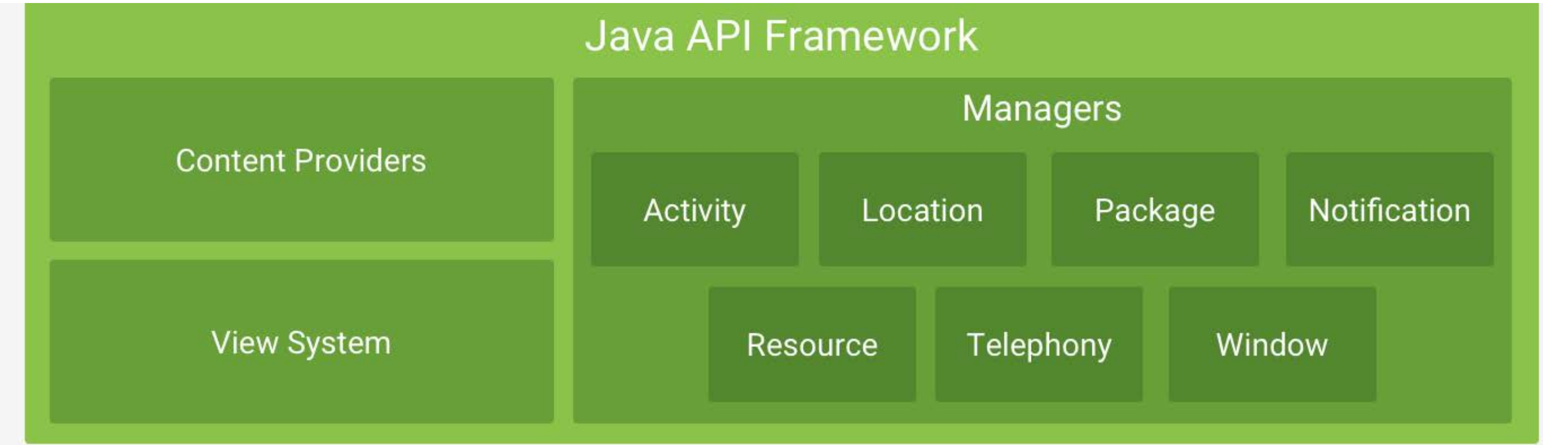
Media Framework

OpenGL ES

. . .

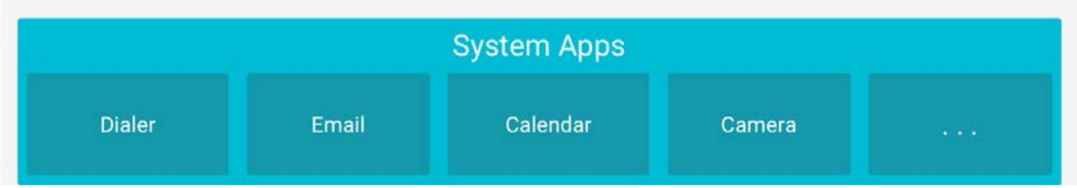
Details

- API Framework
  - Open platform for developers
  - Makes possible to develop fast and efficient
  - High level of freedom: we can use the same resources and components as System applications do
  - In addition these components can be substituted
  - Contains the resource manager, and framework responsible for view system
  - It is possible to modify the system components, designs
    - Manufactures often utilize this possibility

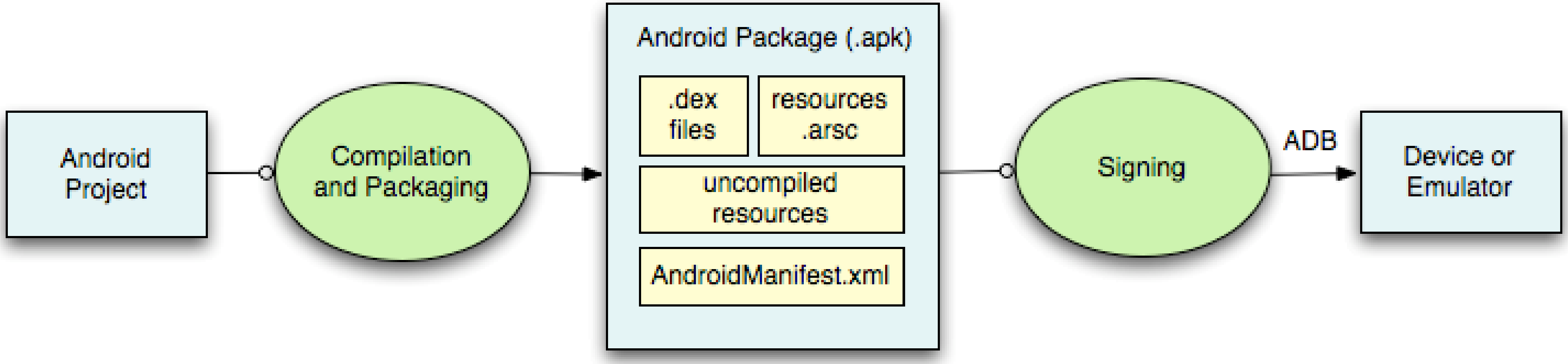


Details

- System Applications
  - E-mail client
  - Sms client
  - Phone Book
  - Calendar



- Android project compilation process



Android development tools

- Android Studio
  - Java SDK
- Android SDK
  - Compiler and program libraries
  - Emulator
    - This is and emulator, where executes the entire Android operating system over you OS on your device
- Android NDK
  - For native (C/C++) libraries
  - Currently you do not need it
- As a result we can develop Android software on any of the main desktop platforms

iOS

iOS properties

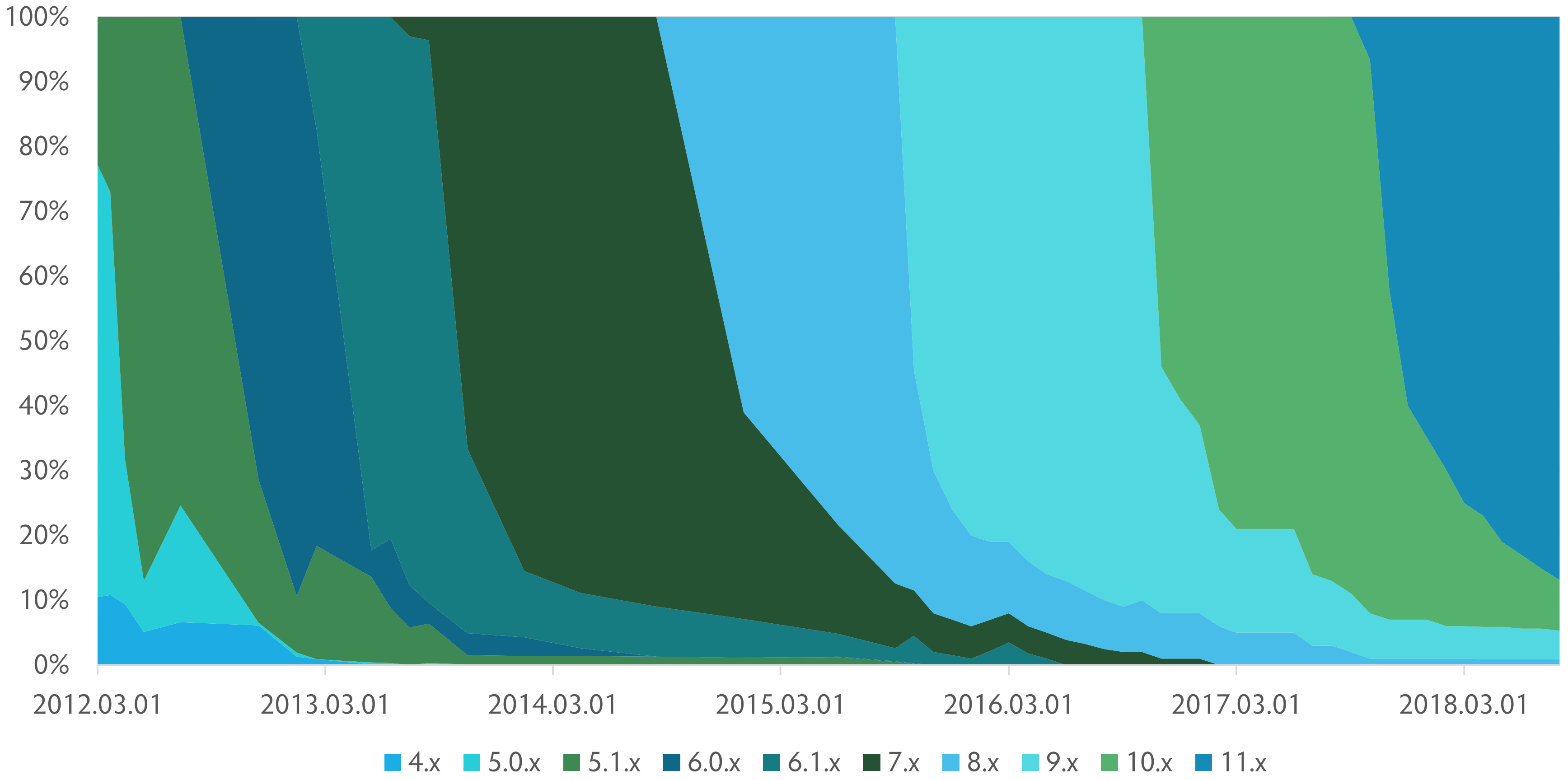
- System and applications developed by Apple
  - Based on Darwin, originated from Max OS X
  - Single manufacture, a few, well-specified hardware
    - ARM processors
    - Transitions between versions are supported
      - For example: new screen resolutions are multiplications of previous ones
      - It can result weird properties, such as incompatibilities with other standards
  - Developments tools and simulator
    - Only for Apple systems
- Main properties
  - Multitask, several built-in program libraries
  - Support for mobile communication systems (GSM … LTE)
  - WiFi, Bluetooth, NFC
  - Sensors: GPS, Triaxial accelerometer / magnetometer
  - Camera support, media recording and playback
  - HDMI support, accelerated 2D and 3D graphics

iOS versions

Introduction	Version	Name	Required Xcode version
2007	1.0	iPhone OS	
2008	2.0	iPhone OS 2.0 Final	
2009	3.0	iPhone OS 3.0 Final	
2010	3.2	iPhone OS 3.2 Final	
2010	4.0	iOS 4.0 Final	
2010	4.3	iOS 4.3 Final	Xcode 4
2010	5.0	iOS 5.0 Final	Xcode 4.2
2012	6.0	iOS 6.0 Final	Xcode 4.5
2013	6.1	iOS 6.1 Final	Xcode 4.6
2013	7.0	iOS 7.0 Final	Xcode 5
2014	7.1	iOS 7.1 Final	Xcode 5.1
2014	8.0	iOS 8.0 Final	Xcode 6
2015	8.4	iOS 8.4 Final	Xcode 6.4
2015	9.0	iOS 9.0 Final	Xcode 7.0
2016	9.3.5	iOS 9.3.5 Final	Xcode 7.3
2017	10.3.3	iOS 10.3.3 Final	Xcode 8.3
2018	11.4.1	iOS 10.4.1 Final	Xcode 9.x
2019	12.4.1	iOS 12.4.1 Final	Xcode 10.x

Pázmány Péter Catholic University
Faculty of Information Technology and Bionics

Spread of different iOS versions



iOS devices

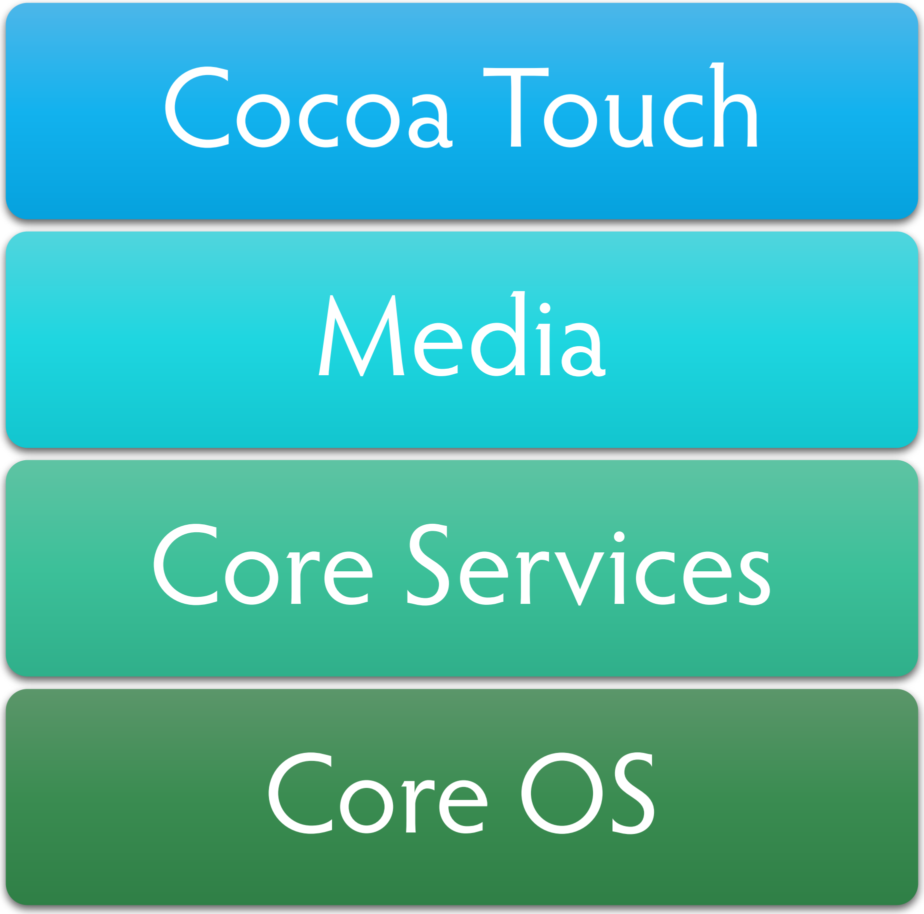
- Low number of different devices
  - Several models, different parameters for each models
  - The maximal available iOS version can be read from the table
  - It implies the supported devices of an applications

iOS	iPhone									iPod Touch		
	5	5C	5S	6/Plus	6S/ SE	7/Plus	8/Plus	X/XS	XR	5th	6th	7th
9.3										X		
10	X	X										
12 limited			X	X								
12 full					X	X	X	X	X		X	X

- iOS 12 on
  - iPad Pro *, 1st, 2nd, 3rd
  - iPad Mini 2nd, 3rd, 4th, Mini (2019)
  - iPad Air & Air 2 & Air (2019)
  - iPad 2017 & 2018

iOS building blocks

- Different software and software levels
    - Based on OS X (Darwin)
- Development on SWIFT language
    - Brand new language, not extension for Objective-C
    - However its logic can be understand better based on Objective-C
- Development on Objective-C language
    - Based on C
        - Thin layer on the C
    - Fully object oriented
    - Smalltalk messaging model



Details

- Core OS
  - Mac OS X Kernel
    - XNU
  - TCP/IP
  - Sockets
  - Energy managements
  - File systems
  - Security

- Core Services
  - Network management
  - Collections
  - Preference storage
  - URL handling
  - File accessing
  - Contacts
  - Embedded SQL database manager
  - Core Location
    - To determine the location of the devices (GPS, Cell, etc.)
  - Thread managements
  - CoreMotion
    - To retrieve the accelerometer data

Details

- Media
  - Core Audio
  - Open AL
    - Open Audio Library
  - Sound mixer
  - Sound recorder
  - Video playback
  - Supporting media formats
  - 2D acceleration
  - Core Animation
  - OpenGL ES

- Cocoa Touch
  - Multi-touch and gestures
  - Camera support
  - Localization (multilingual apps)
  - Views and view hierarchies
  - Accelerometer support
  - Web view
  - Map kit
  - Handling notifications
  - Core Motion
    - Provide and process motion related information

iOS development tools

- XCode
  - With iOS SDK
  - An OS X required as operating system (and an Apple computer)
  - Newer iOS requires newer XCode, and newer OS X
    - And newer hardware
- SDK includes a simulator
  - This is not emulator, it is a simulator
  - A simulator calculates the corresponding output for a specific input
    - Thus the iOS systems is not running on the test-environment.

Mobile platform technologies

Next week