1 - Mobile iOS

iOS Overview

iOS is based on Darwin that is a open-source Unix OS developed by Apple.

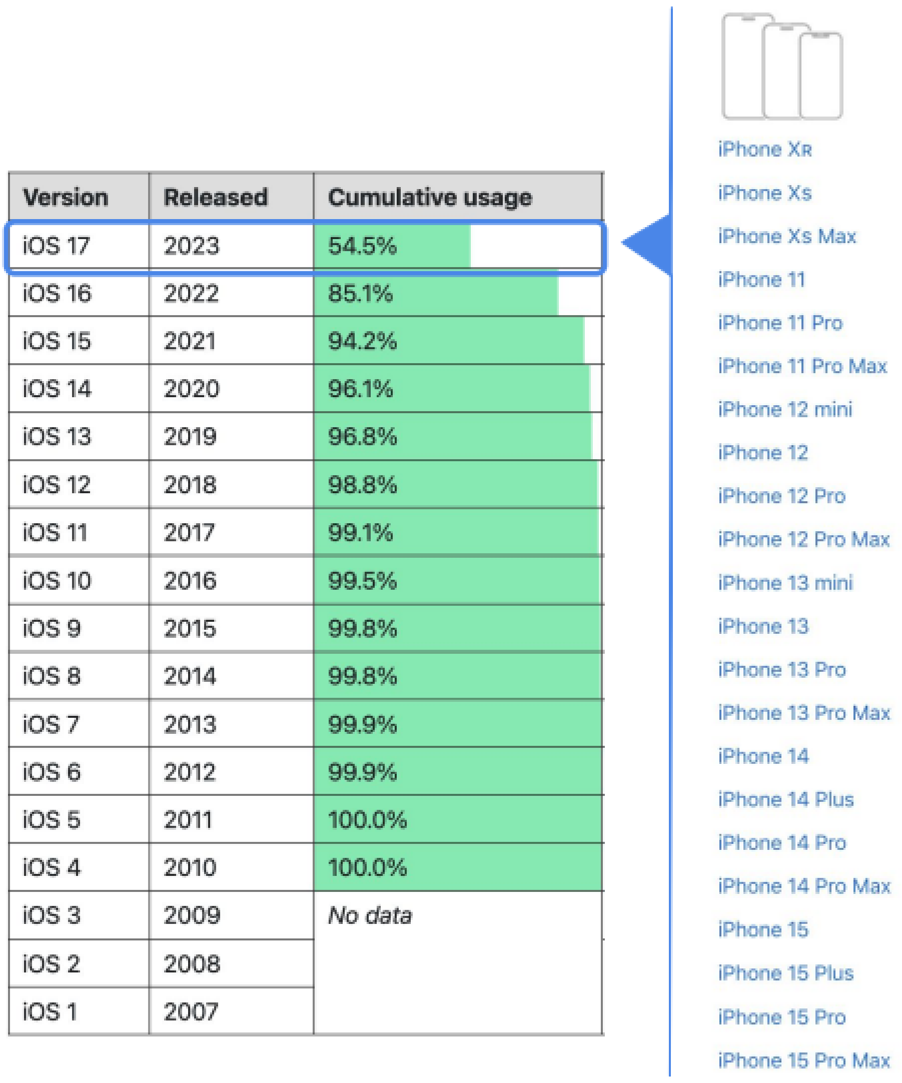
**==Some parts of Darwin are open-source such as Webkit, XNU-Kernel.
Most of them are closed-source and proprietary.==**

iOS Versioning

Apple releases a major version each year (september)

The newest versions are higly adopted:

- iOS 17: 54.5% of all devices
- android 14: 9.3%



Version	Released	Cumulative usage
iOS 17	2023	54.5%
iOS 16	2022	85.1%
iOS 15	2021	94.2%
iOS 14	2020	96.1%
iOS 13	2019	96.8%
iOS 12	2018	98.8%
iOS 11	2017	99.1%
iOS 10	2016	99.5%
iOS 9	2015	99.8%
iOS 8	2014	99.8%
iOS 7	2013	99.9%
iOS 6	2012	99.9%
iOS 5	2011	100.0%
iOS 4	2010	100.0%
iOS 3	2009	*No data*
iOS 2	2008	
iOS 1	2007	

iPhone XR
iPhone Xs
iPhone Xs Max
iPhone 11
iPhone 11 Pro
iPhone 11 Pro Max
iPhone 12 mini
iPhone 12
iPhone 12 Pro
iPhone 12 Pro Max
iPhone 13 mini
iPhone 13
iPhone 13 Pro
iPhone 13 Pro Max
iPhone 14
iPhone 14 Plus
iPhone 14 Pro
iPhone 14 Pro Max
iPhone 15
iPhone 15 Plus
iPhone 15 Pro
iPhone 15 Pro Max

In general older devices loses support, for example iPhone 8 and X cannot install iOS 17

App development

The main platform to develop iOS apps is XCode.

The distribution of apps requires paid Membership (Apple Developer Program)

- 99 USD/year
- it allows to publish apps in Apple App Store

It is also possible to distribute Custom Apps within an Organization

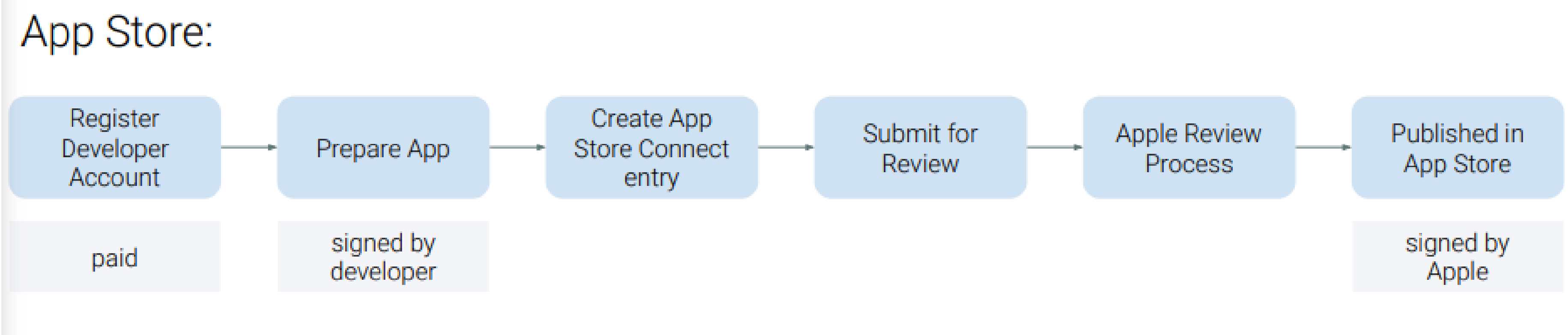
- developer Enterprise Program : 299 USD/year

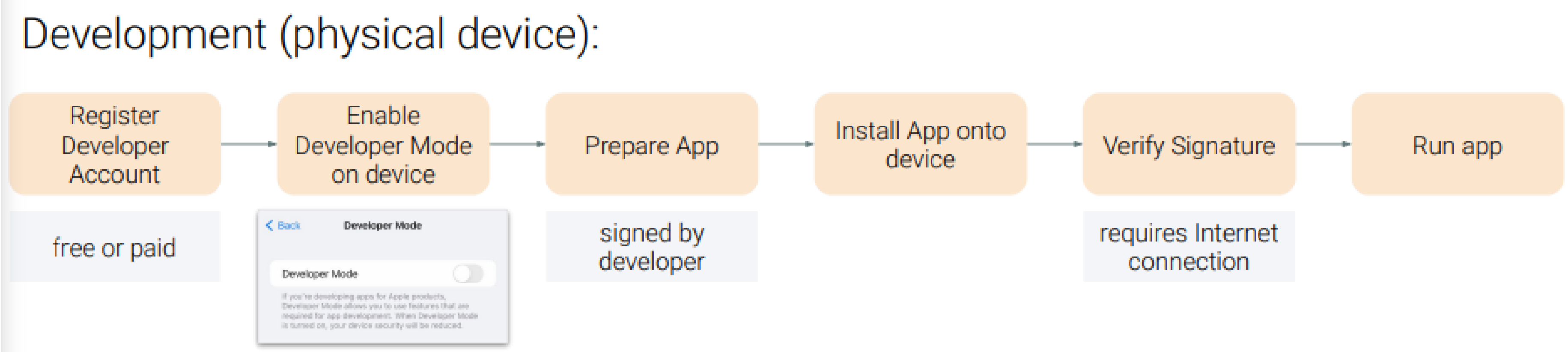
There is also a platform to test apps that is TestFlight.

Each submitted app for publishing it in the App Store is reviewed by apple.

Each app is signed using certificates issued by Apple.

Distribution process




Development (physical device):
Register Developer Account
Enable Developer Mode on device
Prepare App
Install App onto device
Verify Signature
Run app
free or paid
Back
Developer Mode
Developer Mode
If you're developing apps for Apple products, Developer Mode allows you to use features that are required for app development. When Developer Mode is turned on, your device security will be reduced.
signed by developer
requires Internet connection

- it is possible to use the installed app only for 48 hours in this case

iOS architecture

The iOS architecture is split into different parts.

1. **Core OS.** It contains the kernel, file system, security features, power management ecc
2. **Core services.** It involves networking, GPS service, gyroscope, security APIs
    1. it provides objects for primitive data types and os-services and defines object behaviour
3. **Media.** It provides graphical and multimedia services to the next layer
4. **Cocoa Touch.** It supports applications and it is the main responsible for interactions with the user.
    1. it contains the UIKit, a framework for User Interface

iOS App (.ipa package)

iOS apps are distribuited as IPA (iOS Package Archive)

- **it is simply a .zip archive. If we rename it in .zip we can extract it without problems.**

The structure is:

```
iTunesMetadata.plist
META-INF
    ↳ zip metadata
Payload
↳ MyApp.app
    ↳ Info.plist
    ↳ Frameworks
    ↳ MyApp
    ↳ further resources
```

where:

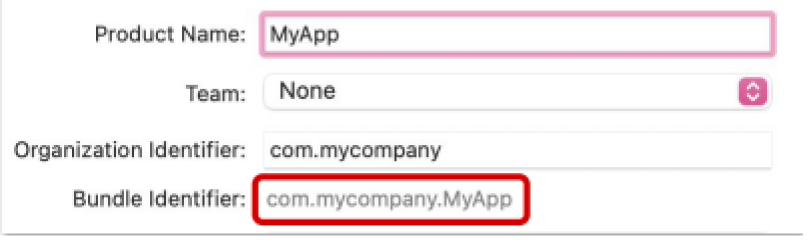
- **Info.plist** is the Information Property List and contains the configuration for the app and all information about permissions and so on
- **Frameworks**, it is a folder that contains some of the frameworks used by the app
- **MyApp** that is the app binary
- further resources contains language fiels, icons, GUI objects, videos and so on

Application identifier

Bundle identifier

It is chosen by the developers and it is usually the reverse domain (example com.mycompany.MyApp)

It used to uniquely identify an app on the device.



Product Name: MyApp

Team: None

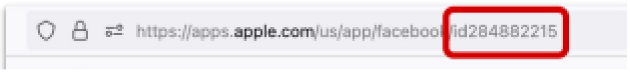
Organization Identifier: com.mycompany

Bundle Identifier: com.mycompany.MyApp

Store identifier

It is provided by Apple and is formed by numbers only.

It uniquely identifies an app on the App store



https://apps.apple.com/us/app/facebook/id284882215

iOS permissions

To access protected resources, apps declare a description (UsageDescription) in their Info.plist

On the first access on a protected resource, iOS shows a confirmation dialogue to the user that contains the UsageDescription

The OS monitors access to protected resources at runtime.

Users can revoke their choice later on.

The protected resources are:

- Location Services
- Contacts
- Photos
- Microphone
- ...

iOS App Sandboxing

All apps from tha App Store are Sandboxed with containers.

When an app is opened a process is run as unprivileged "mobile" user.

If an app wants to perform an operation with elevated/root permissions they have to declare it in their entitlements

- **entitlements are key-value pairs, digitally signed into an app**

In addition mmap and mprotect are modified to prevent access to memory that is writable and executable.

- `mmap()` viene utilizzata per mappare file o altri oggetti in memoria, consentendo loro di essere trattati come array di byte.
- `mprotect()` viene utilizzata per modificare le autorizzazioni di accesso per le pagine di memoria mappate con `mmap()`

Inter-app communication

The communication can be done from:

- **App groups (same developer)**
  - it allows multiple apps access to shared containers and interprocess communication between apps
- **Deep links (different developers)**
  - it allows an app to be opened by other apps or form the browser
  - it allows passign of information via URL
    - Custom URL Schemes such as myapp://some-resource
    - Universal links such as https: //myapp.com

Activity lifecycle

On each state change UIKit notifies the app:

- in iOS >= 13 using the UISceneDelegate
- in iOS < 13 via UIApplicationDelegate

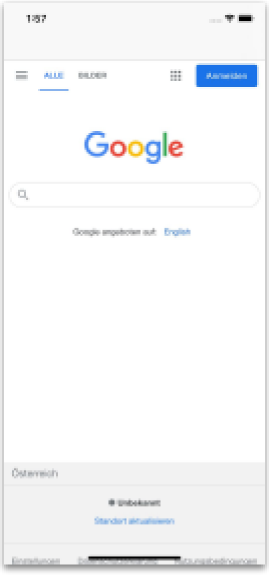
Each Scene has its own life cycle:

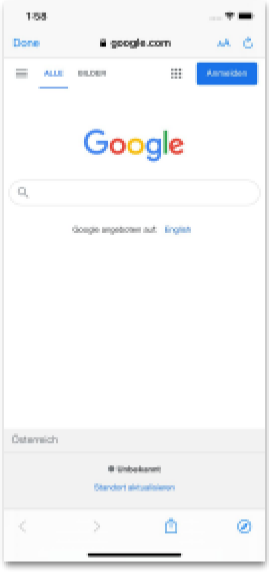
1. **Unattached** that is a newly created scene , dismissed or suspended for some time
2. **Inactive** that is when the app is launched and there are no user interactions
3. **active**
4. **backgroud** that involves system requested scenes or user dismissed scene
5. **suspended**, when the scene is in background for some time.

WebViews

WebViews are used to display Web-content (HTML, websites) directly within an app.

There are two types of WebViews:

- **WKWebView**, that interacts with web-contents and are customizable and use WebKit
- 

- **SFSafariViewController** that shows web-contents , is not customizable and useses Safari with SafariServices framework
- 

Browsers

Before iOS 17.4 browsers were forced to use WebKit, so it was a skin for safari, in order to avoid the using of JIT.

With iOS 17.4 and EU regulations apple allows other browsers engines only in EU.

iOS secure enclave

From iphone 5s an later we have a **secure sub-system called secure enclave**.

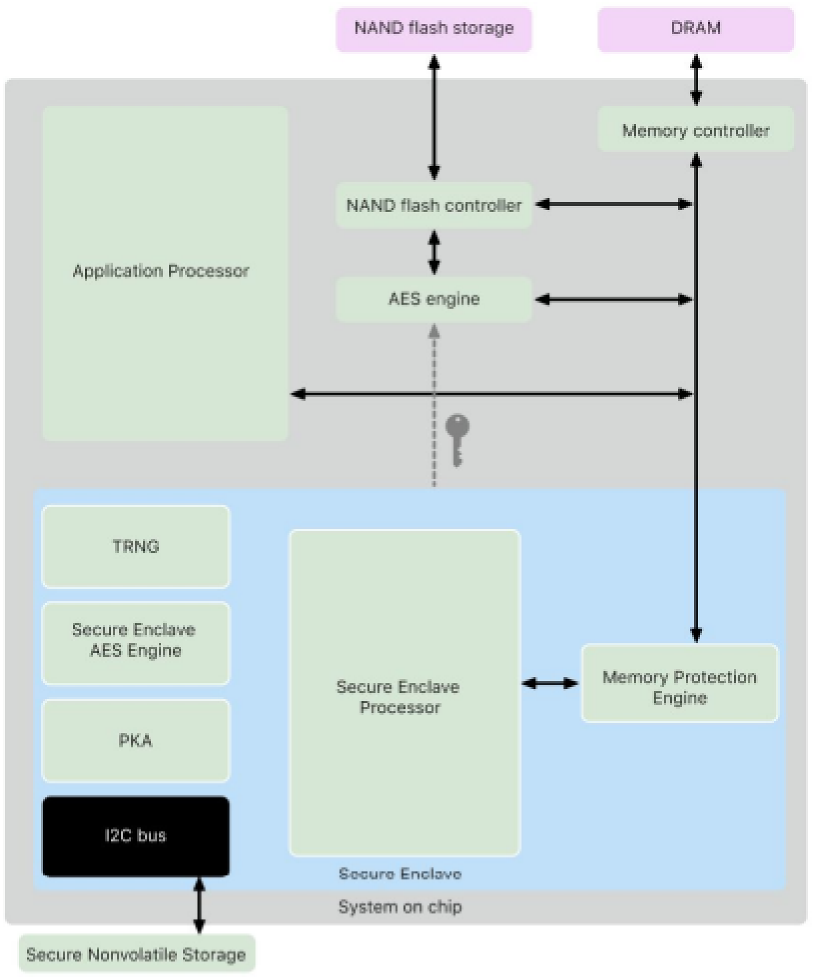
It is isolated from main processor and should keep user data safe even if Application Processor kernel is compromised.

The **Secure Enclave Processor prevents Side-channels**.

We have also **protected memory and boot ROM that establishes harware root of trust and has immutable code.**

We have an **AES engine to counter timing and power analysis**.

It is also used a **True Random Number Generator**



JailBreaking

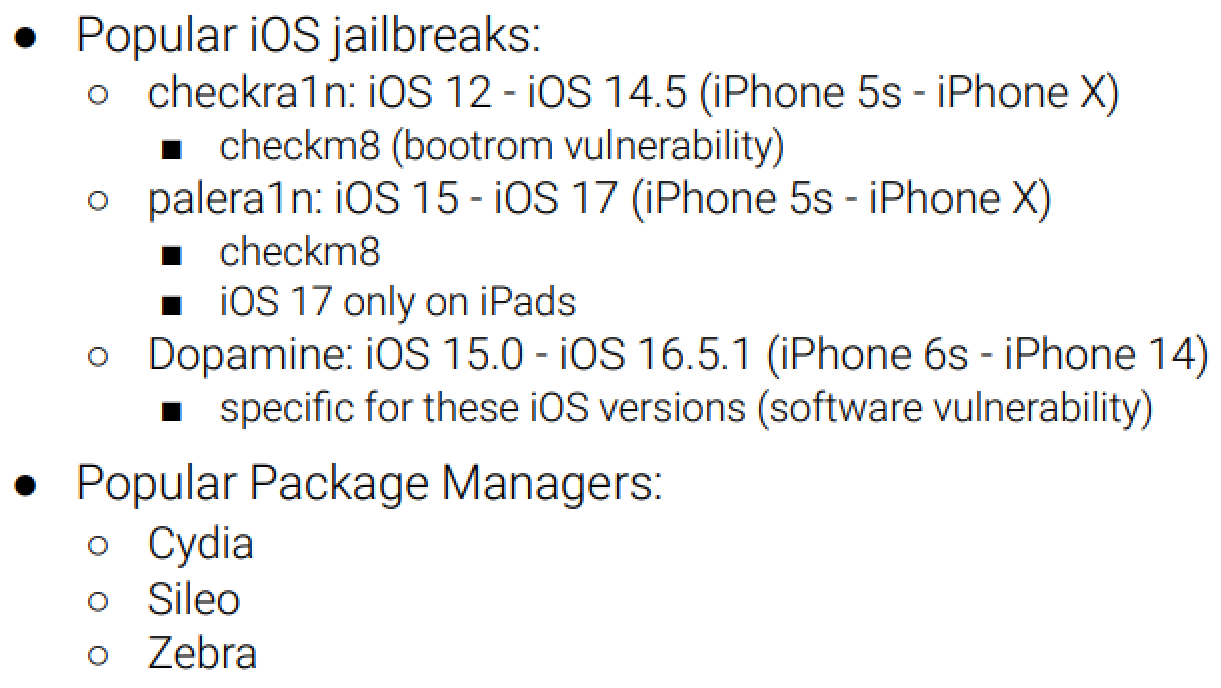
The jailbreaking is the using of exploits in a locked-down system (iphone) used to remove restrictions placed by the manufacturer (apple).

It enables:

- **root access**
- **users to install software from other sources (unsigned)**

The jail reduces attack surface and apps can only access their own data containers, run as unprivileged ans must be signed and reviewed by Apple.

Jailbreak is important for research purposes because we need to monitor applications and traffic.



- Popular iOS jailbreaks:
  - checkra1n: iOS 12 - iOS 14.5 (iPhone 5s - iPhone X)
    - checkm8 (bootrom vulnerability)
  - palera1n: iOS 15 - iOS 17 (iPhone 5s - iPhone X)
    - checkm8
    - iOS 17 only on iPads
  - Dopamine: iOS 15.0 - iOS 16.5.1 (iPhone 6s - iPhone 14)
    - specific for these iOS versions (software vulnerability)
- Popular Package Managers:
  - Cydia
  - Sileo
  - Zebra