

Chapter 1:

Introduction to mobile app development

Plan

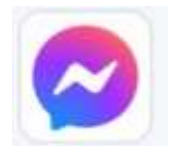
- 1. Mobile App definition.**
- 2. Mobile operating systems.**
- 3. Some statistics.**
- 4. Mobile App development.**
 - Development constraints.
 - How do mobile apps work ?
 - Development approaches.
 - Development Frameworks & Tools.

Plan

- 1. Mobile App definition.**
- 2. Mobile operating systems.**
- 3. Some statistics.**
- 4. Mobile App development.**
 - Development constraints.
 - How do mobile apps work ?
 - Development approaches.
 - Development Frameworks & Tools.

Mobile application

- **Definition:** *mobile application (app)* is an autonomous software application developed specifically for run on mobile device such as smartphones and tablets rather than desktop or laptop computers.
- Offers access to services in Web or Mobile version.
- Available on distribution platforms (Stores): *App Store, Google Play, Windows Phone Store.*
- **Examples:**



Mobile application

Challenges :

- **High performances:** running application without crashes or bugs and consuming little space in the device without affecting battery life.
- **Profitability:** the app's target audience, its monetization strategy, market competition and the overall effectiveness of its marketing efforts.
- **Scalability:** ability to handle increased workload or user demand => frequent updates.

Plan

1. **Mobile App definition.**
2. **Mobile operating systems.**
3. **Some statistics.**
4. **Mobile App development.**
 - Development constraints.
 - How do mobile apps work ?
 - Development approaches.
 - Development Frameworks & Tools.

Mobile operating systems

- Mobile OS is an OS that helps to run other application software on mobile devices.
- It focuses on the management of :
 - Wireless connectivity,
 - Internet navigation,
 - Interface types,
 - Phone calls,
 - Embedded memory,
 - Process ...

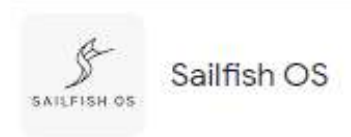
Mobile operating systems

- **iOS:** was developed by Apple for its mobile devices such as iPhone, iPad, iPod.
- **Windows Phone:** was developed by Microsoft. It's not an open-source mobile OS.
- **Android:** open-source based on Linux OS. It was developed by Android and re-purchased by Google in August 2005.
- **Symbian OS (Nokia), Blackberry OS (RIM), Bada (Samsung):** less widespread on the market.



Mobile operating systems

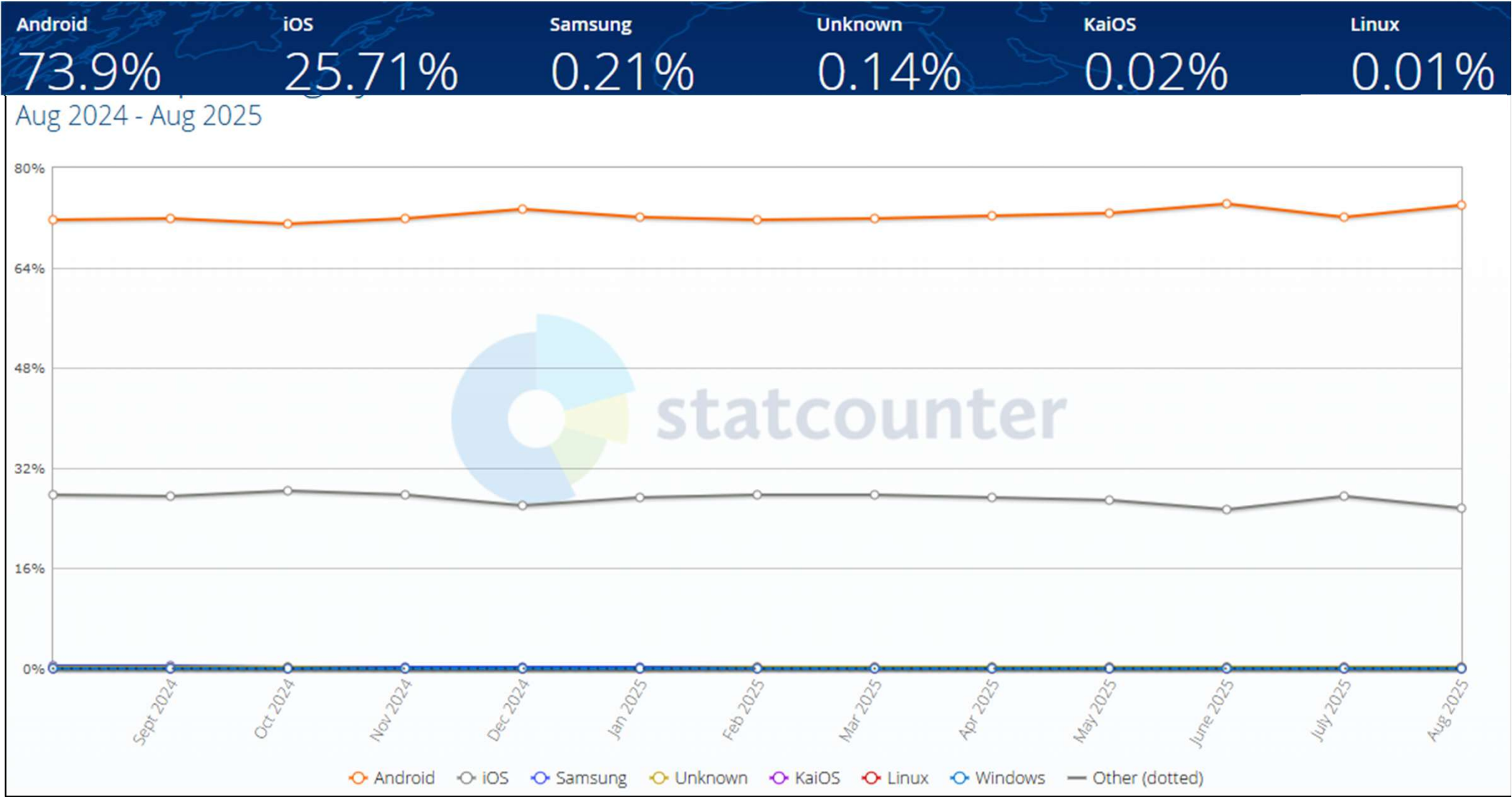
- **Existing mobile OS:**



Plan

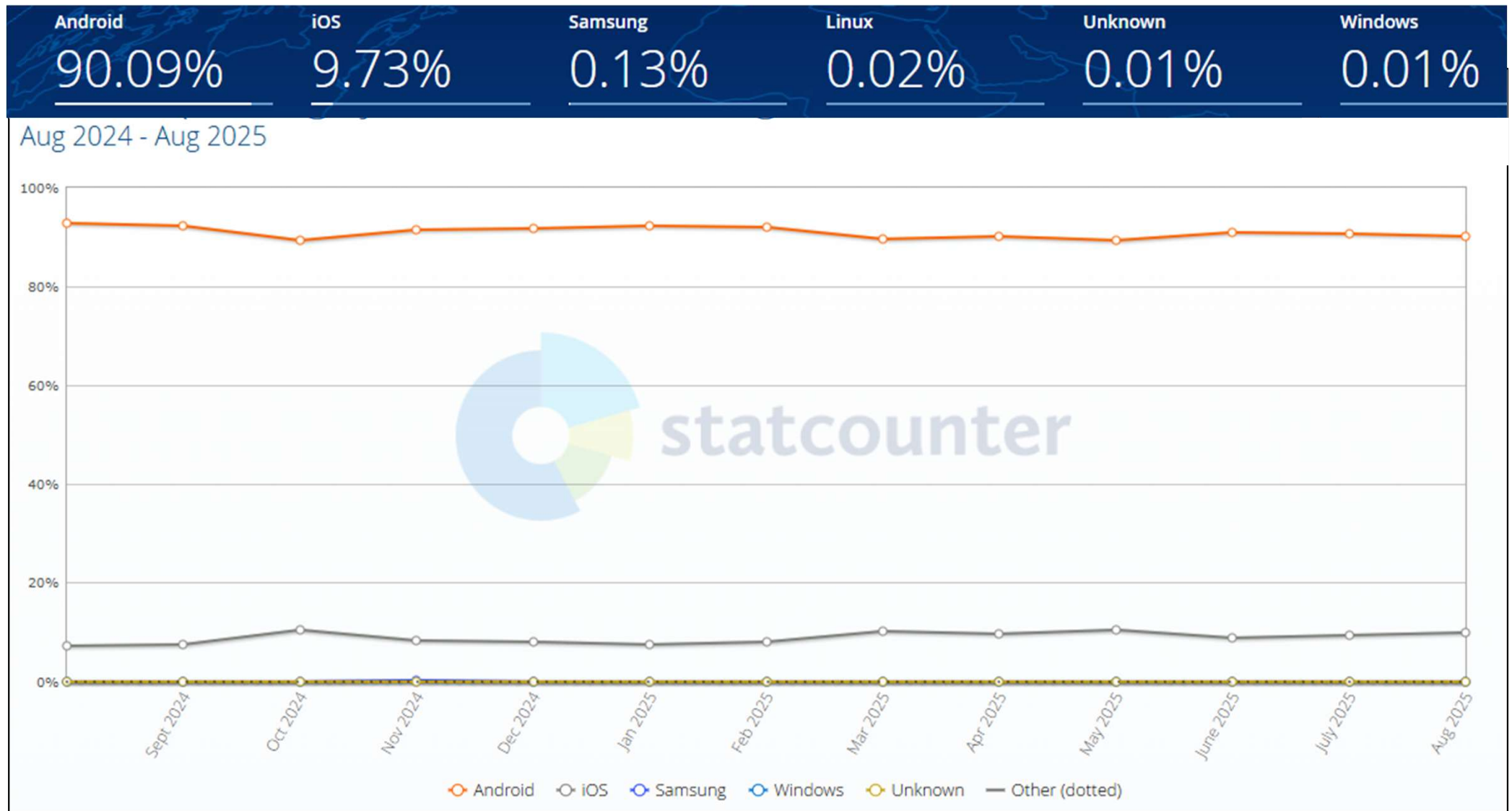
1. **Mobile App definition.**
2. **Mobile operating systems.**
3. **Some statistics.**
4. **Mobile App development.**
 - Development constraints.
 - How do mobile apps work ?
 - Development approaches.
 - Development Frameworks & Tools.

Mobile OS market share worldwide in 2025



(*) <https://gs.statcounter.com/os-market-share/mobile/worldwide>




Mobile OS Algerian market share in 2025



(*) <https://gs.statcounter.com/os-market-share/mobile/algeria>

Mobile OS market share worldwide

- **By platform :**

	Tablet 		Smartphone 		Desktop 
World	iOS 51.21%	Android 48.66%	Android 72.37%	iOS 26.98%	Windows 75.34%
France	iOS 55.9%	Android 43.75%	Android 62.12%	iOS 37.29%	Windows 74.71%

Plan

1. **Mobile App definition.**
2. **Mobile operating systems.**
3. **Some statistics.**
4. **Mobile App development.**
 - Development constraints.
 - How do mobile apps work ?
 - Development approaches.
 - Development Frameworks & Tools.

Mobile development constraints

- **Features of mobile devices:** limited memory, limited processing power, small screen size, different display resolution, wireless connectivity, limited bandwidth, heterogeneity of OS and technologies ...
- **The market concept:** reduce the Time To Market (TTM), the period from mobile app conception to it being available for purchase or use.
- **The platform choice:** knowing benefits and limitations of each platform with respect to application need and objectives.

Mobile development constraints

- Who is the target audience for the mobile application?
- What will it be used for?
- By which mobile device and on which operating system?
- Does the application need to integrate native functionalities ? (e.g. GPS, Camera, ...).

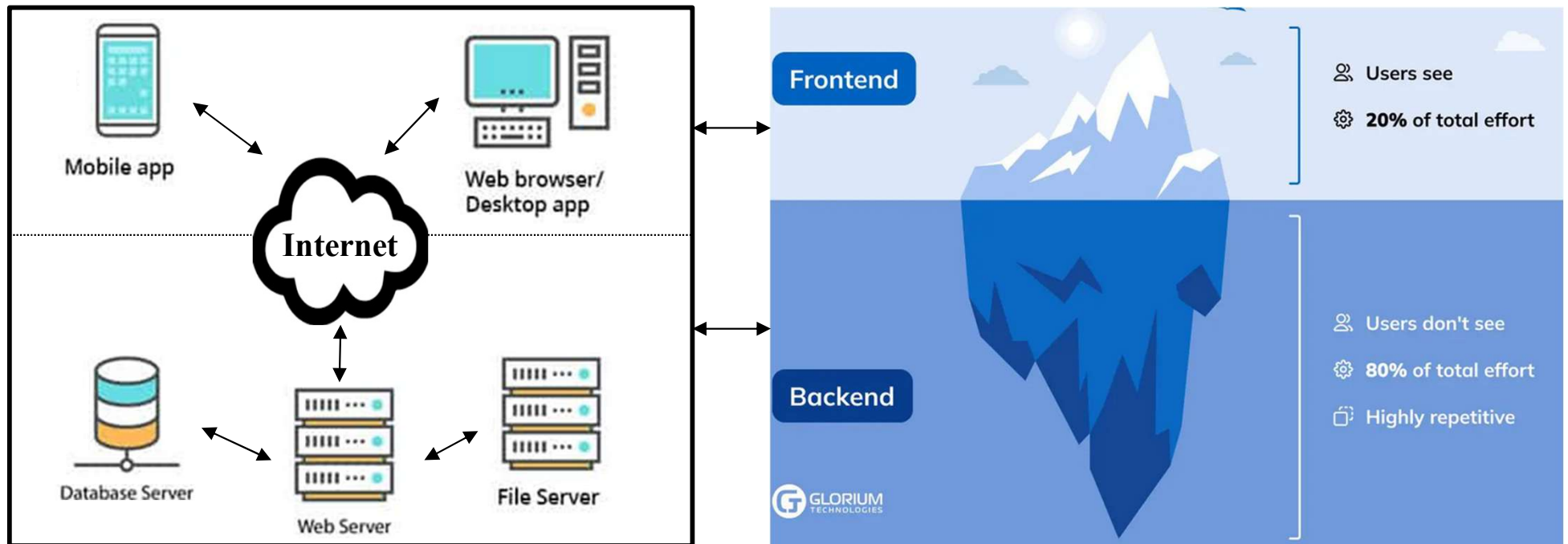
Plan

1. **Mobile App definition.**
2. **Mobile operating systems.**
3. **Some statistics.**
4. **Mobile App development.**
 - Development constraints.
 - How do mobile apps work ?
 - Development approaches.
 - Development Frameworks & Tools.

How do mobile apps work?

Mobile apps consist of two parts:

1. Frontend (client side): is where users interact with the software.



2. Backend (server side): is responsible for the application's logic and data handling. It acts as the brain of the app.

How do mobile apps work?

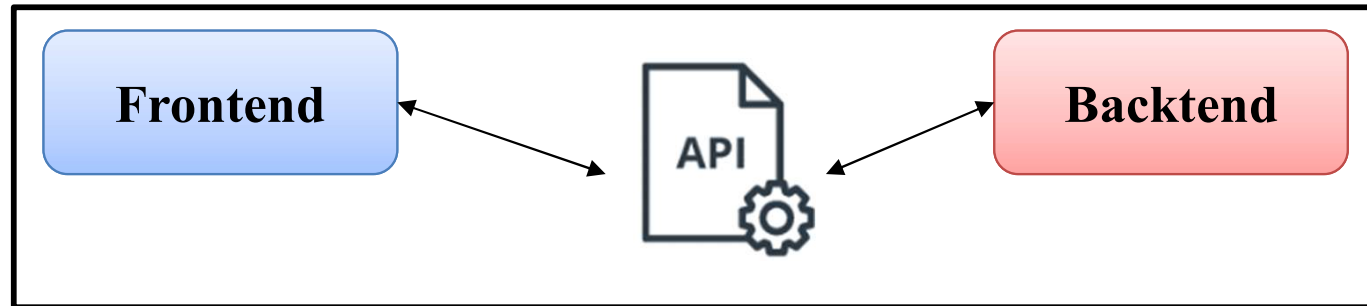
Mobile backend is an app designed for communication among machines and servers, it runs on a remote server without graphical interface.

Mobile backend takes care of:

- Data processing and storage independent of a smartphone's capabilities.
- Data sync and sharing across multiple devices.
- Content updates within the mobile app.
- Authorization and authentication that control access to data.
- Heavy processing operations.

How do mobile apps work?

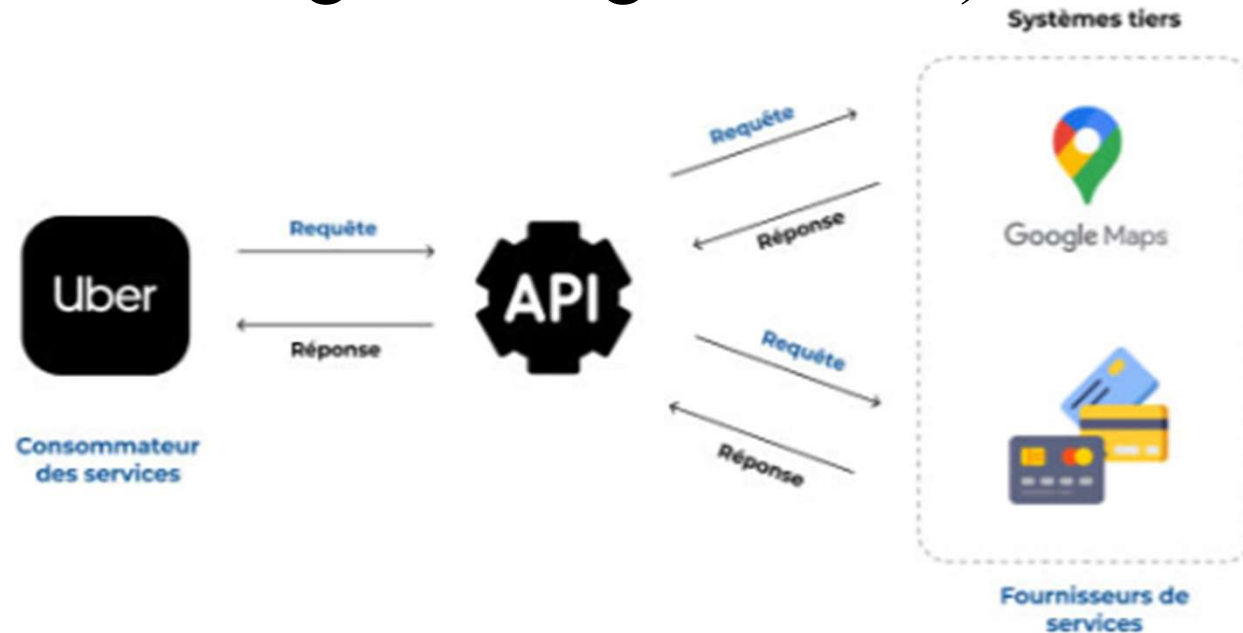
Developers access to **mobile backend** via **APIs** (Application Programming Interface).



API is an interface that refers to a set of protocols, rules and tools that allows different software applications to interact and communicate with each other.

How do mobile apps work?

Developers access to **mobile backend** via **APIs** (Application Programming Interface).



API is an interface that refers to a set of protocols, rules and tools that allows different software applications to interact and communicate with each other.

How do mobile apps work?

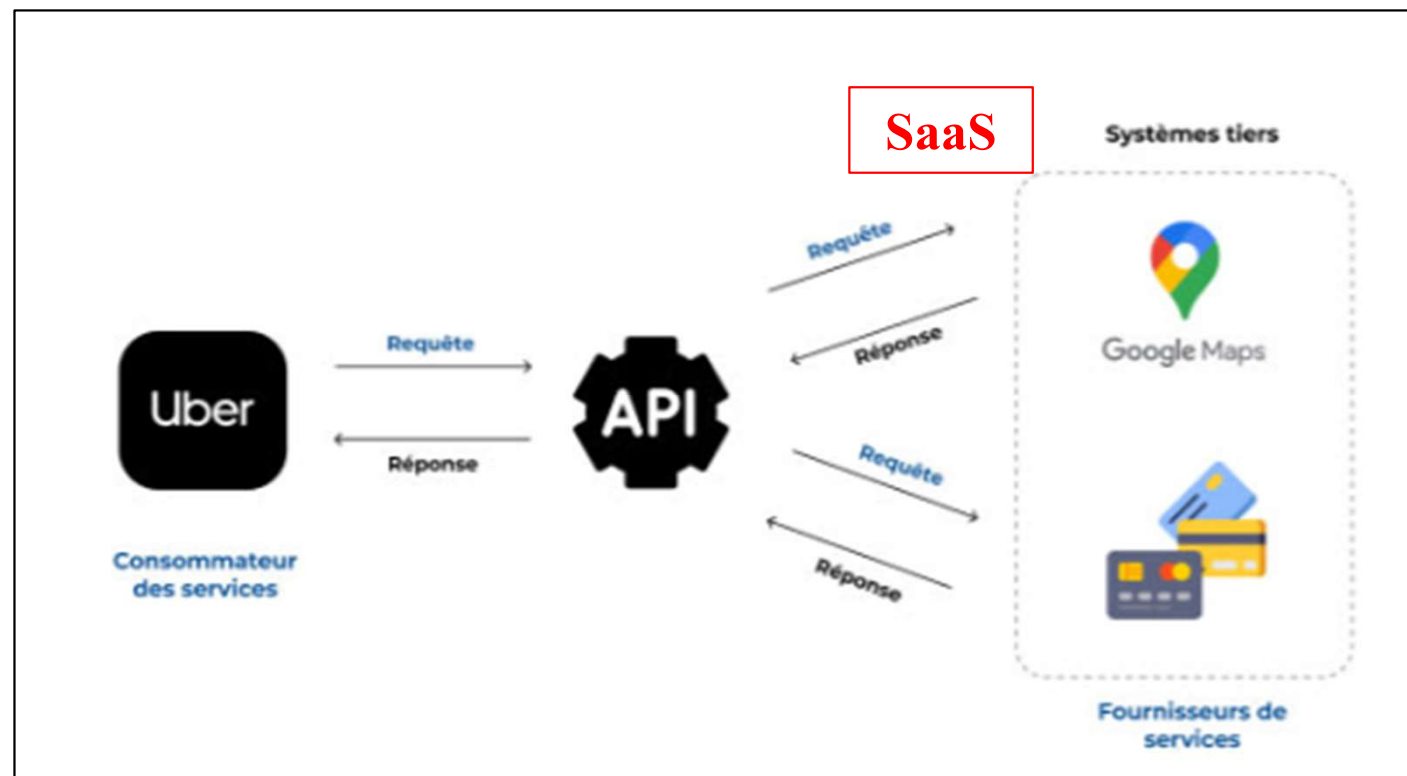
Types of backend:

Third-party SaaS

Is available solution on the market i.e. ready-made, plug-and-play solution.

You plug in your frontend, subscribe to their plan, and release your app.

Exp: AccuWeather API, Shopify



How do mobile apps work?

Types of backend:

Third-party SaaS

Is available solution on the market i.e. ready-made, plug-and-play solution.

You plug in your frontend, subscribe to their plan, and release your app.

Exp: AccuWeather API, Shopify

MBaaS Solutions

Gives a basic structure that can mold based on a specific requirements.

Allows the use of pre-developed components with a backend to offer capabilities to **store** user data, **login**/ registration features, **authentication**, and push **notifications**.

Exp: Firebase

Custom Backend

Designed to meet the unique requirements of a particular application.

Full control, flexibility, performance, security, scalability.

Zero dependencies and min API connections.

Exp: Node.js, Ruby on Rails

Plan

1. **Mobile App definition.**
2. **Mobile operating systems.**
3. **Some statistics.**
4. **Mobile App development.**
 - Development constraints.
 - How do mobile apps work ?
 - Development approaches.
 - Development Frameworks & Tools.

Mobile app development approaches

- This includes both :
 1. **The technologies used to build mobile apps.**

Native Apps dev

Cross-Platform
Apps dev

2. **The solutions that make other apps work well on mobile devices.**

Web Apps dev

Hybrid Apps
dev

Progressive
Web Apps dev

Mobile app development approaches

1. Native apps: means creating a mobile app that is tailored and dedicated to a specified platform like iOS, or Android. It requires using the native language of the mobile device **e.g:**

- *Java or Kotlin for Android,*
- *Objective-C or Swift for iOS,*
- *C# for Windows Phone.*

- Use the internal memory of the mobile device and everything the phone offers to work well.
- Apps are available on stores dedicated to the target platform.

Mobile app development approaches

1. Native apps:

Characteristics :

- 😊 Access to the OS native functionalities (contact, calendar,...) and hardware resources (GPS, camera, sensor, Bluetooth, ...).
- 😊 Provides high performance & fast execution as it's built specially for the OS (easy communication with the device).
- 😊 Offers more control and customization over the UI. Takes full advantages of the platform's design and features (Optimized UI for the platform).

Mobile app development approaches

1. Native apps:

Characteristics :

- ☹️ Uni-platform: generalization \Rightarrow duplicate efforts for each platform \Rightarrow time consuming & high cost.

Mobile app development approaches

1. Native apps:

Technology examples :

Operating system	IDE	Language	Store	Open Source
 Android	 Android Studio	Java Kotlin	 Google Play	Oui
 iOS	 Xcode	Objective-C Swift	 App Store	Non
 Windows Phone	 Visual Studio Microsoft Compact Framework	C#	 Windows Phone Store	Non

Mobile app development approaches

- 2. Cross-Platform apps:** is an approach to develop apps that are compatible with multiple mobile OS (apps can be used regardless of the OS).

From one code, it's possible to generate multiple apps for iOS and Android.

Mobile app development approaches

2. Cross-Platform apps:

Characteristics :

- 😊 No need of specialized developers.
- 😊 Faster and lower cost for development, maintenance and update (one code for multiple platforms).
- 😊 Closer performance to native apps.
- 😊 Access to device features through plugins or native code.
- 😊 Wider reach (audience).
- 😊 Reusable code.

Mobile app development approaches

2. Cross-Platform apps:

Characteristics :

- ☹ UI design quality: can't take advantage of and use all the build-in features provided by mobile devices.
- ☹ Update lag: app can't take advantage of new OS features immediately.
- ☹ The need to add additional platform-specific work to handle a particular functionality.

Mobile app development approaches

2. Cross-Platform apps:

Technology examples :

- React Native.
- Flutter.
- Ionic.
- Kotlin Multiplatform.

Mobile app development approaches

- 3. Web apps:** are websites that look and feel like apps on phones or computers. They can be accessed by the web browser of the mobile device.

There is no need for a user to download the application from an app store.

Mobile app development approaches

3. Web apps:

Characteristics :

- 😊 Cross-platform (run in web browsers).
- 😊 No installation needed => require minimum device memory.
- 😊 Easy to maintain.
- 😞 Loading depends on internet speed.
- 😞 No access to phone's features (GPS,...).
- 😞 Internet is essential.

Mobile app development approaches

3. Web apps:

Technology examples :

- HTML.
- CSS.
- JavaScript.

Mobile app development approaches

- 4. Progressive web apps (PWAs):** are web apps that aim to provide a user experience similar to that of native mobile apps. They leverage modern web technologies to deliver the same features as traditional mobile apps (exp: icon on home screen).

Characteristics :

- 😊 Run on any platform through browser (mobile and desktop devices).
- 😊 Act like a mobile app with offline availability, push notifications, access to device hardware (GPS,...).

Mobile app development approaches

4. Progressive web apps (PWAs):

Characteristics :

- ☹ Browser compatibility issues: not all PWAs features are supported by every browser.
- ☹ Limited access to device hardware.
- ☹ Performance limitation: they may struggle with handling large datasets or complex computation due to browser memory and processing limitations.
- ☹ Managing offline states and synchronization can be complex.

Mobile app development approaches

4. Progressive web apps (PWAs):

Technology examples :

- Service Workers,
- Web App Manifests,
- IndexedDB,
- Frameworks (Angular, Ionic, React), ...

Mobile app development approaches

- 5. Hybrid apps:** mix web apps and native mobile apps. They are made with web technology but are **put into** a mobile app shell (i.e. they can be download from app stores).

They can do some things that mobile apps do, like using camera or notifications.

They are a middle ground, not as perfect as mobile apps but better than just web apps.

Mobile app development approaches

5. Hybrid apps:

Characteristics :

- 😊 Quick and simple development.
- 😊 Single codebase for all platforms => Reduces maintenance costs and simplifies updates .
- 😊 Access to phone's features (GPS,...).
- 😞 Run slower and may have performance issues.
- 😞 Appearance might vary across different devices.

Mobile app development approaches

5. Hybrid apps:

Technology examples :

- Cordova (HTML, CSS, JS).
- Ionic.
- React Native.
- Flutter.

Plan

1. **Mobile App definition.**
2. **Mobile operating systems.**
3. **Some statistics.**
4. **Mobile App development.**
 - Development constraints.
 - How do mobile apps work ?
 - Development approaches.
 - **Development Frameworks & Tools.**

Development Frameworks & Tools

A **framework** is a structured set of components and tools that provide a foundation for developing software applications.

It facilitates the development process by offering pre-built functionalities and development tools, allowing developers to focus more on the specific features of their application rather than on basic underlying infrastructure.


Development Frameworks & Tools



Flutter: multi-platforms, a software development kit for mobile apps. Created by *Google* in 2013 and published in 2017. It's based on *DART* programming language.

E.g:



 **Ionic:** multi-platforms, open source created in 2013. A technology based on *AngularJS*, *Cordova* et *Node JS*.

E.g:



Development Frameworks & Tools



React Native: open source, multi-platforms for JavaScript. Created by *Facebook* in 2015. It's based on ReactJS.



Xamarin : multi-platforms. It's acquired by *Microsoft* and based on *C#* language.



Conclusion

- Together, the **frontend** and **backend** work seamlessly via APIs to provide a functional and responsive mobile app, ensuring a smooth user experience while efficiently managing data and resources on the server side.
- There are many mobile app development approaches (**native**, **hybride**, **cross-platform**, ...), each one offers advantages and disadvantages.
- The choice of a mobile app development approach depends on some constraints such as the target audience, platform, required fonctionnalités, ...

Appendix: SaaS backend

Pros	Cons
<ol style="list-style-type: none">1. Low cost of backend development/integration.2. No need to have a dedicated backend development team.3. Shorter time-to-market. Can be deployed in a few hours.	<ol style="list-style-type: none">1. Lack of flexibility in building and launching custom features.2. Monthly limits/restrictions depending on the plan.3. Limited scalability potential as the user base grows.

Appendix: MBaaS backend

Pros	Cons
<ol style="list-style-type: none">1. More flexible and customizable than SaaS backend solutions.2. Tiered cost structure based on usage. Pay only for the resources you utilize.3. Ready-to-use components and pre-programmed libraries for embedding features and capabilities within the app.	<ol style="list-style-type: none">1. Backend service outage can affect your app users.2. Costs more in case of highly-scalable apps that attract a huge amount of traffic in the long run.

Appendix: Custom backend

Pros	Cons
<ol style="list-style-type: none">1. Highly flexible and adjustable as per the needs of the project.2. Unlimited scalability and reliability.3. Ultimate security and data privacy – no dependency on third-party integrations.	<ol style="list-style-type: none">1. Costly.2. Takes a lot of time and specialized talent to develop and manage.

Appendix: Comparison between hybrid apps and PWAs

Apps dev approach Definition	Development	Deployment	Device features access
PWAs	Developed like any other web application using HTML, CSS, and JavaScript or frameworks like Angular, React, and Vue.js.	Deployed to a web server and accessed via a URL. Users "install" them by adding them to their home screen from the browser.	Full access
Hybrid	Developed using HTML, CSS, and JavaScript, often within frameworks like Cordova, Ionic or React Native. They are wrapped in a native container that allows them to be installed.	Compiled into native binaries and distributed through app stores like Google Play and Apple App Store.	Limited access