Introduction Mobile Programming

First Two Weeks

Pervasive Computing

- **Pervasive computing,** also called **ubiquitous computing**, is the growing trend of embedding computational capability (generally in the form of microprocessors) into everyday objects to make them effectively communicate and perform useful tasks in a way that minimizes the end user's need to interact with computers as computers.

- Pervasive computing devices are network-connected and constantly available.

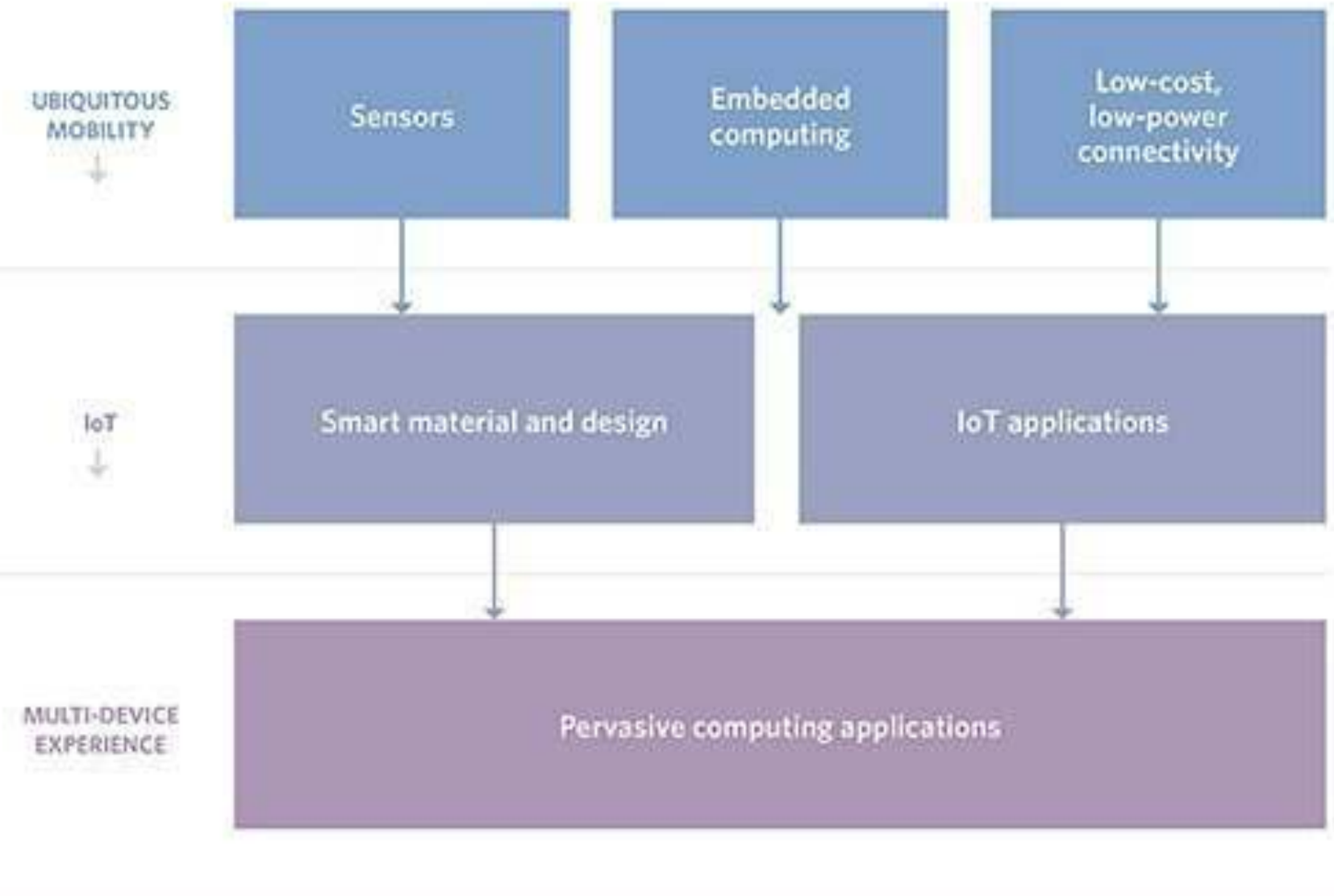
“Existing everywhere: Anywhere, Anytime, Any Device, Any Network, Any Data”

Pervasive Computing

Evolution of Pervasive Computing

- Distributed Computing
- Mobile Computing
- Pervasive Computing

Pervasive Computing



Tools to use:

iOS:

XCode

AppCode

Atom

Android:

Android Studio

Android IDE

Intellij IDEA

Examples:

Artsy

Pinterest

iOS calculator

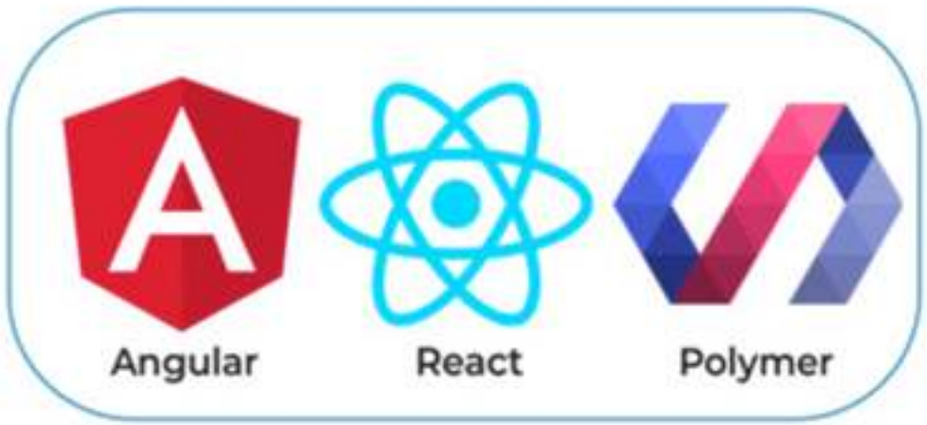
Tools to use:

Examples:

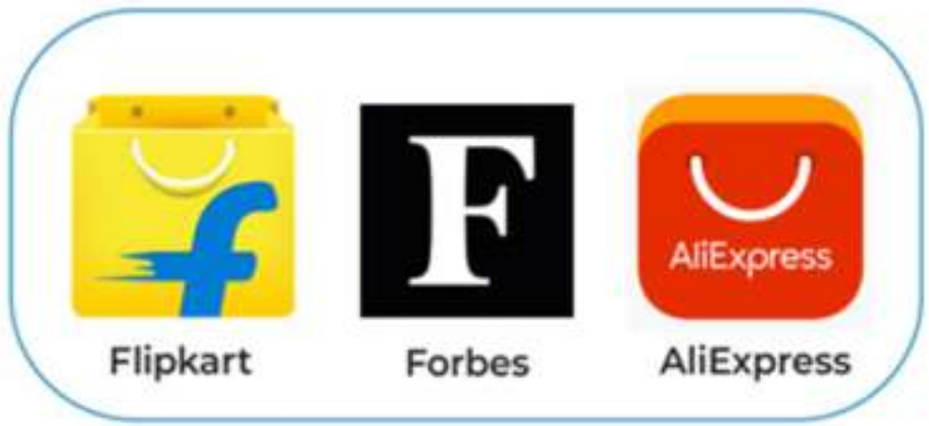
Tools to use:

Examples:

Tools:

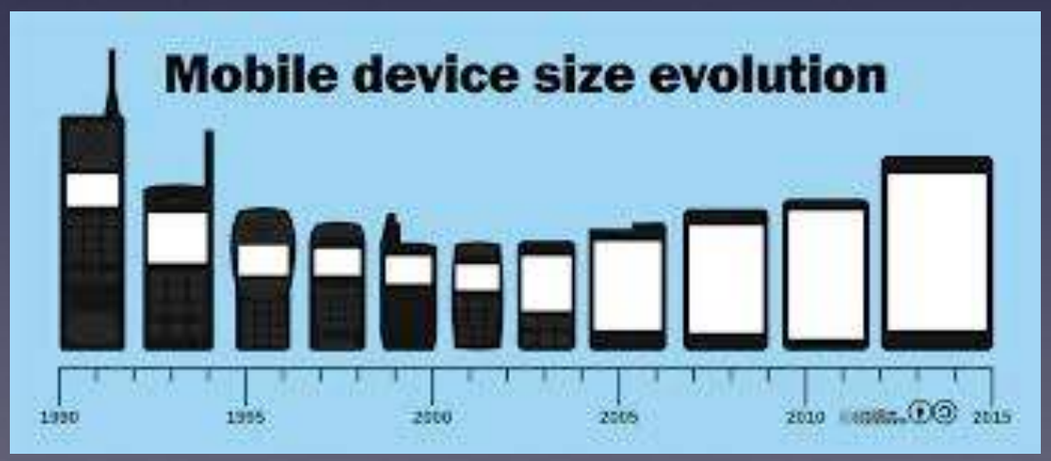


Examples:



Mobile Devices

Mobile Phones



Feature phone vs Smartphone

- Feature phones are highly economical. If you want to get a mobile phone with all the basic features without spending a lot of money, features phones are the best choice
- Not only are feature phones bulkier they are very hard and durable. If you accidentally slip your phone and it hits the ground, chances are the phone won't get damaged at all.
- Feature phones always win over smartphones in terms of battery life. There aren't any high-power consuming apps in the feature phones making the battery lasts for a really long time.
- Most people prefer feature phones over smartphones because they are easier to use with buttons and smart keys.
- Making and receiving calls is very convenient in feature phones because of the clear sound quality and louder volume.
- There won't be the extra expense of buying a phone case, screen protector, phone stand if you are using a feature phone. On the other hand, smartphones do seem to require these unnecessary expenses for their protection and convenience.
- Smartphones might make you more productive if you are someone who has the habit of frequently engaging in social media. It will keep you focused when you are working.

Feature phone vs Smartphone

- The processing speeds of the latest smartphones even beats a PC these days. With its memory and storage capacity in numerous gigabytes, sensors, display functions, it is very superior to basic phones.
- The best part about smartphones is their ability to enormously surf the internet at the fastest speed. Staying connected to friends and families through video calls, texting, voice messaging and various social media platforms have made life so much easier for people.
- All kinds of official works can be performed with the help of smartphones like sending an email, holding a conference meeting, making slideshows and PowerPoint presentations, making corrections in Google Docs. There's literally no need for laptops or computers.
- Another best part about smartphones is their multimedia functionality which allows people to click high-resolution pictures and videos, listen to lots of music online, and play high storage games.
- Using wireless connections facilities like BlueTooth, WiFi you can also connect your smartphone to other devices such as your watch, computers, or your car audio. Tracking your way using GPS tracking is another cool facility of smartphones that is not available in feature phones.
- Smartphones are very slim in shape which makes them very comfortable to carry around or put in pockets. The screen makes use of the best quality glasses making their experience more likable and enjoyable.
- The ability to download various applications online is another important function of smartphones. You can download different new apps and run them independently on smartphones.
- Unlike feature phones, you can enjoy making changes to the settings of your smartphones like changing the theme to a different color, using the fingerprint sensors, or face- recognition for device protection. There is also a function to track your smartphone if you lose it somewhere using various methods.

Operating System Tasks

- Managing Memory
- Managing Processor
- Managing Devices
- Management of File System
- Computer Security
- Controlling Performance of the System
- Job allocation
- Detection of Errors and Troubleshooting
- Coordinating other users and software applications

Mobile Operating System Tasks

- With catchy graphics, it should also be easy to use
- It should provide a good useful app store
- It should have a longer battery life
- Should be good at managing data and network usage

Operating Systems

Features	Mobile Operating System	Desktop Operating System
Definition	It is a type of operating system that allows application software to operate on mobile devices.	It is the environment in which a user handles a personal computer.
Memory Requirement	It needs minimum RAM to optimize.	It needs huge memory to operate.
Storage	It uses a flash drive to store the data.	It uses hard drives and flash drives to store data.
Boot Time	It takes less time to boot.	It takes much time to boot.
Purpose	It handles cellular and wireless connectivity and device access.	It handles the software and hardware resources of the system.
Power	It is optimized to work under minimal power needs and has a feature to prevent energy loss.	It is not readily optimized for energy loss.
Interface	It runs on touchscreen or touchpad devices.	It runs through many input devices, including a mouse, keyboard, etc.
Example	Some examples of the Mobile OS are Apple iOS, Google Android, Bada, Palm OS, Symbian OS, Windows Mobile OS, Blackberry OS, iPhone, Harmony OS, WebOS, etc.	Some examples of the desktop OS are Windows 10, MacOS, Windows Vista, etc.

Mobile Operating Systems

Year	Android (%)	iOS (%)
2023	71.65	27.71
2022	71.47	27.85
2021	71.89	27.34
2020	73.06	26.28
2019	75.47	22.71
2018	75.45	20.47
2017	72.63	19.65
2016	69.11	19.29
2015	64.2	20.2
2014	53.65	23.95
2013	39.21	24.03
2012	27.41	24.04
2011	19	22.29
2010	8.82	25.48
2009	2.41	34.01

Mobile Operating Systems

Operating System	Market Share in Q1 2023 (%)
Android	71.95
iOS	27.42
Samsung	0.34
Unknown	0.13
KaiOS	0.1
Windows	0.02
Linux	0.01
Other	0.02

iOS vs Android

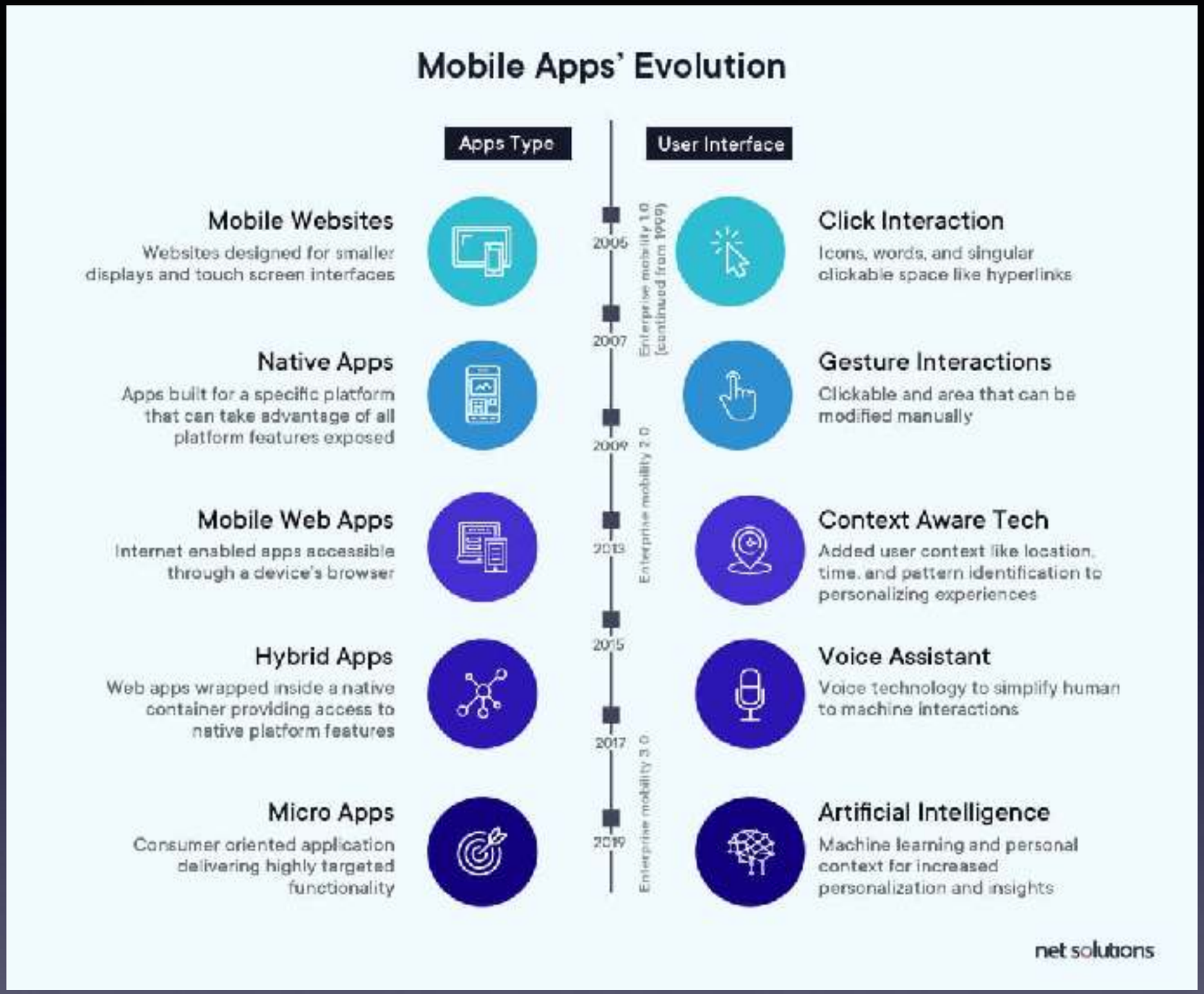
	Apple iPhone 13	Apple iPhone 13 Pro Max	Samsung Galaxy S21	Samsung Galaxy S21 Ultra
Lowest Price	$799	$1,099	$800	$1,200
Processor	A15 Bionic	A15 Bionic	Qualcomm Snapdragon 888	Qualcomm Snapdragon 888
Storage	128GB/256GB/512GB	128GB/256GB/512GB/1TB	128GB/256GB	128GB/256GB/512GB
Connectivity	LTE, 5G	LTE, 5G	LTE, 5G	LTE, 5G
Display Size	6.1 inches	6.7 inches	6.2 inches	6.8 inches
Display Type	OLED	OLED	AMOLED	AMOLED
Display Resolution	2,532-by-1,170 pixels	2,778-by,1,284 pixels	2,400-by-1,080 pixels	3,200-by-1,440 pixels
Refresh Rate	60Hz	120Hz Variable	120Hz Variable	120Hz Variable
Front-Facing Camera	12MP	12MP	10MP	40MP
Rear-Facing Cameras	12MP, 12MP	12MP, 12MP, 12MP	12MP, 12MP, 64MP	108MP, 10MP, 10MP, 12MP
OS	iOS 15	iOS 15	Android 11	Android 11
Special Features	MagSafe, Face ID	MagSafe, Face ID, Variable Refresh Rate, LiDAR scanner	Variable refresh rate	Variable refresh rate, S Pen support

iOS vs Android

Closed (iOS)	**Open (Android)**
Only available on Apple's products.	Available for free to use on any hardware.
Source code is a trade secret.	Source code is public and available for modification. Companies can launch competing platforms on altered Android code.
All devices come with an Apple experience - 36 pre-installed Apple apps with limited ability to change defaults.	Google's apps are available for download in a suite of 11 free apps by agreement. But users can freely change defaults, and hardware makers can use Android without Google and/or preload non-Google apps.
Apps are only available through Apple's proprietary store, and must go through a lengthy approval process.	Most apps are delivered through Google's Play store, but users can use alternative stores. Developers can place apps in the Play store through a simpler process.
Apple can monetize by charging a premium on hardware and through the use of Apple's mandatory app store and other default applications.	Because Google distributes Android for free, Google monetizes Android if users choose to use Google's apps and services, including the Play app store.
Apple can guarantee developers a consistent environment across all Apple hardware, so long as users update to the latest iOS version.	Android can be different based on the hardware and additional tweaks made as it makes its way from Google, to handset manufacturer, to carrier, to customer. Without proper management, changes to the code may fragment the ecosystem so that developers' apps may not work properly across all devices.

iOS vs Android

Android	iOS
Programming Language Used: Java	Programming Language Used: Objective-C Swift
Development Tools Used: Android Studio, Xamarin Studio	Development Tools Used: Xcode, AppCode
App Publishing Cost: $25 one-time fee	App Publishing Cost: $99 one-time fee
Procedure: The app gets instantly approved once submitted to the Google Play Store	Procedure: The app usually takes 4-6 weeks for approval
Pros	**Pros**
Largest smartphone market share	More valuable audiences than Android
Short app review process	Latest iOS adoption rate is high as compared to Android
Deeper market penetration	Uses more advanced and stable tools
Larger customer base	iOS apps generate more revenue
Built in beta testing and staged rollout	Development cost is less
Cons	**Cons**
2-3x higher development costs	Long app review process (usually takes 7-8 days on average)
Less revenue as compared to iOS apps	Smaller customer base
Android users are less likely to make online purchase	iOS apps are less used for ad purposes
High device fragmentation	Less global smartphone market share than Android
Android's latest OS adoption rate is less as compared to iOS	Fewer app downloads than Android



Mobile Application Evolution

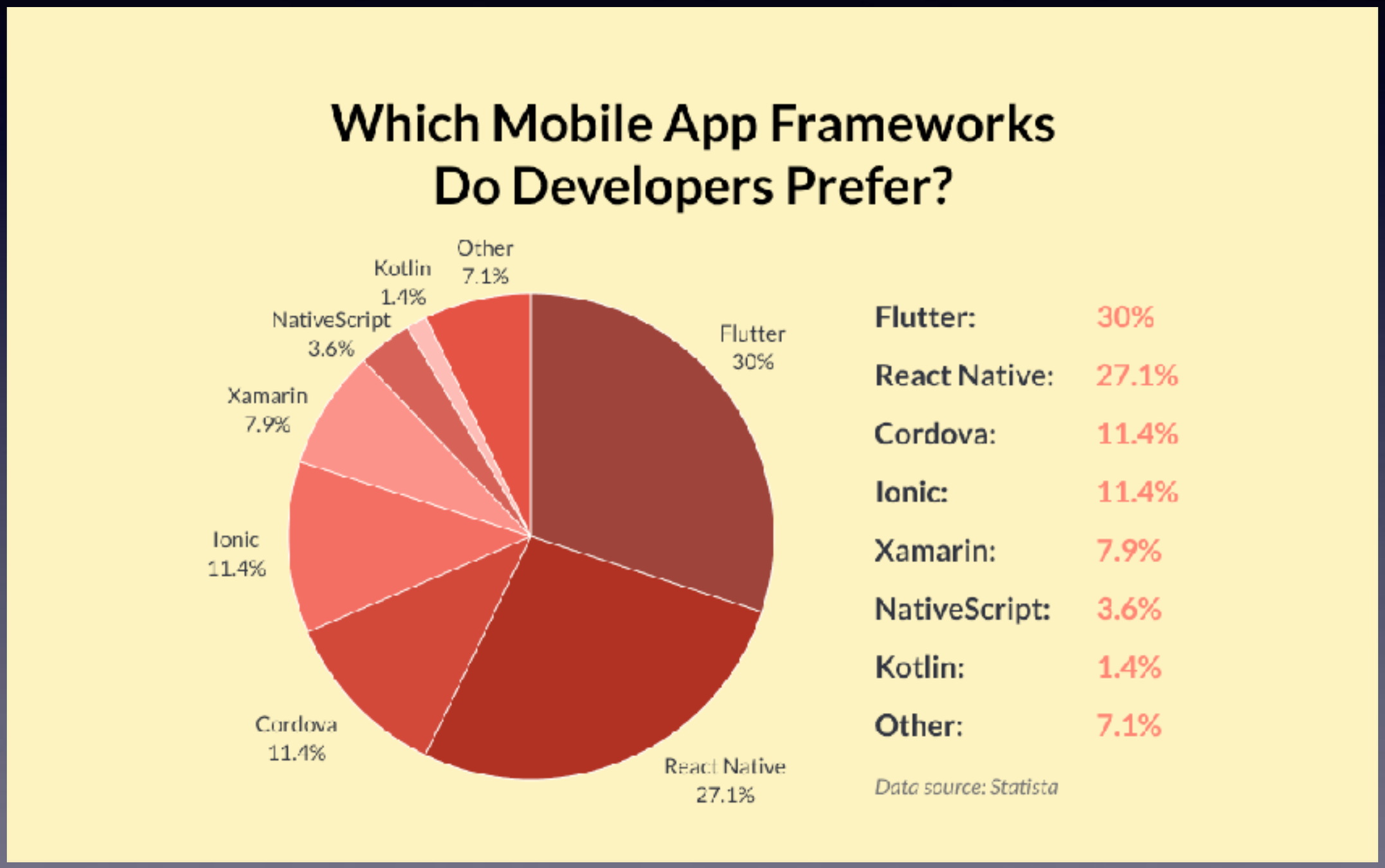
DIFFERENCE BETWEEN TYPES OF APPLICATIONS

	NATIVE MOBILE APPLICATIONS	CROSS-PLATFORM APPLICATIONS	HYBRID-WEB APPLICATIONS	PROGRESSIVE WEB APPLICATIONS
MEANING	Native mobile apps are developed by writing the programming language and designing frameworks that can function on a particular platform such as IoS and Android.	Cross-platform mobile applications are created by writing single codebase that can be functioned or deployed on multiple platforms.	The coding of Hybrid-Web Applications is written in languages like HTML, CSS, or JavaScript. This type of app works on multiple platforms and run in a webview.	Progressive Web Applications are hybrids of regular web pages and native apps.
ADVANTAGES	■ Great user experience. ■ Direct Access to the Device APIs.	■ Single Code for multiple platforms. ■ It is easy to maintain a cross platform app.	■ Only one codebase is required for hybrid apps. ■ Access to device's internal APIs and device hardware.	■ The same App is for Web and Mobile. Hence, it has Less Expensive Upfronts. ■ No need to install or update as it is accessible through URL.
DISADVANTAGES	■ Multiple codebases for each platform are required. ■ Direct AcAs compared to other approaches, developing & maintaining this app is quite expensive.cess to the Device APIs.	■ The user experience is affected due to a lack of integration with the target-ed operating systems. ■ Slow Code Performance.	■ Lower Performance as compared to Native Apps. ■ Less scope of customization due to ative features.	■ Slower and less responsive as compared to Native Mobile Apps. ■ User experience varies based on the browser in use.

App type	Native	Hybrid	Cross-platform
Tools	• XCode • AppCode • Android Studio	• Ionic • Apache Cordova • Visual Studio	• React Native • Xamarin • Flutter
Rendering engine	Native	Browser	Native
Ease of development			
Libraries	Not much dependency on open-source libraries and platforms	Highly dependent on different libraries and frameworks	Highly dependent on different libraries and frameworks
Debugging	Native debugging tools	Native + web development debugging tools	Depends on the framework*
Codebase	separate codebases - one per platform	single codebase with potential platform-specific abilities	single codebase with potential platform-specific abilities
Pros	• Full access to device's/OS's features • Powerful performance • Native UI (updated along with the OS) • Efficient App Running • High-quality functionality and UX • Access to all native APIs and the platform-specific functionality	• Lower development costs • Different OS support • Code reuse • Cost-effective development • Big customization capabilities	• Different OS support • UI performance is almost as fast as native • Code reuse • Cost-effective development (however, a new language or approach will need to be learned)
Cons	• No multi-platform support • High dev costs if different OS support is needed	• Slower performance • Limited access to OS features	• Slower performance • Limited access to OS features

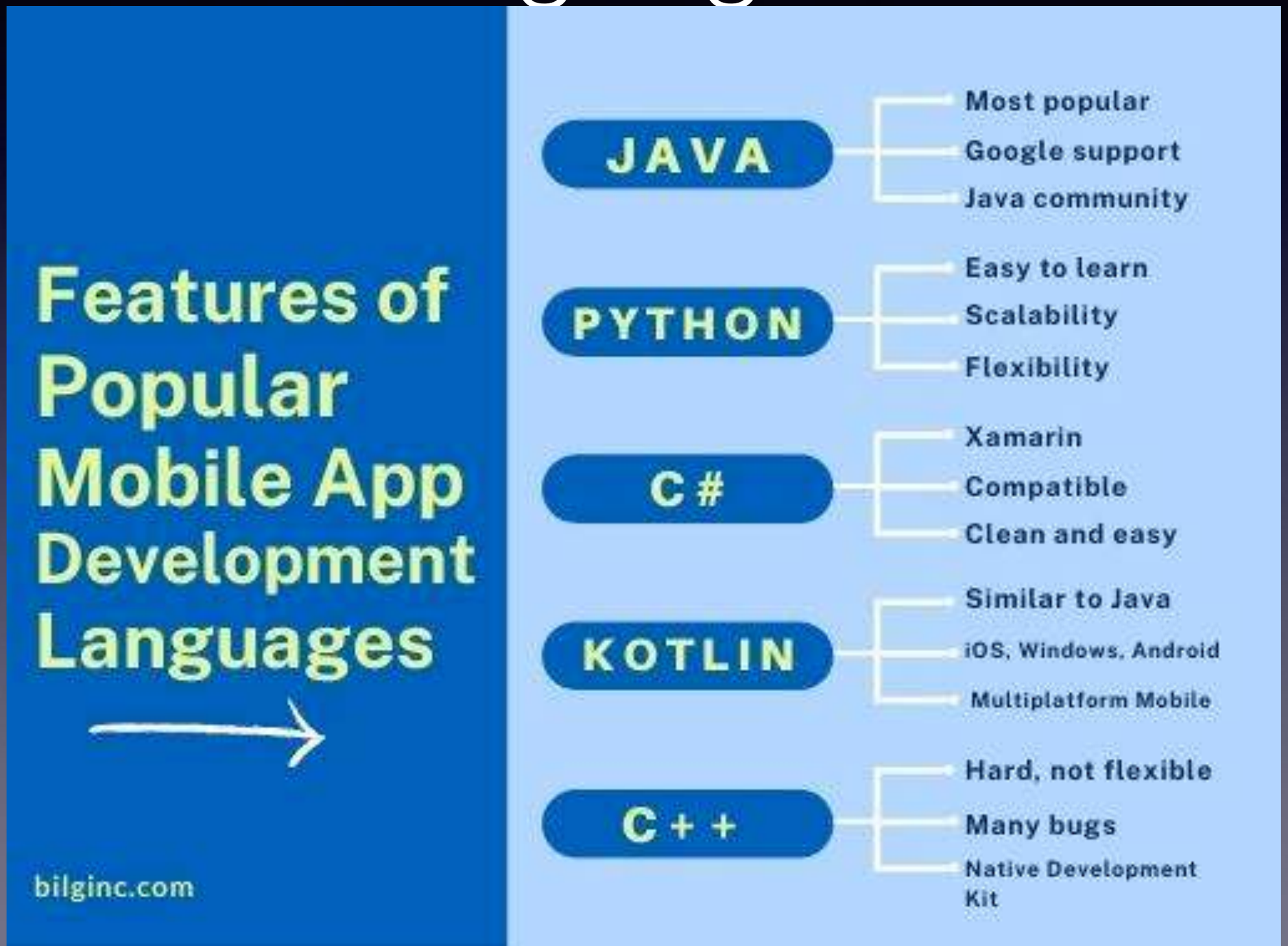
Programming Languages

Mobile App Frameworks

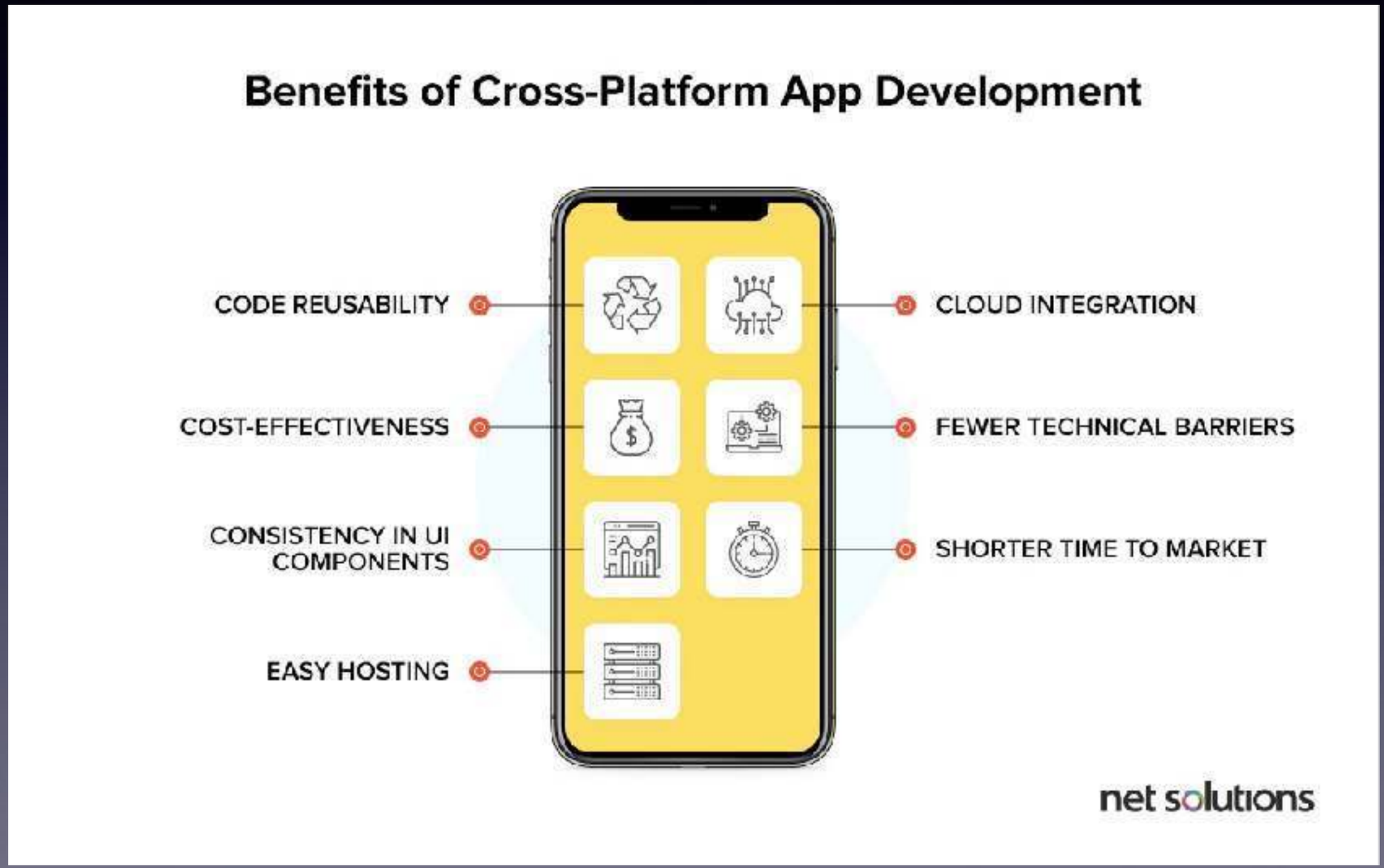


Mobile Programming Languages

Mobile Programming Languages



Cross-Platform Development



Cross-platform Tools

React Native

Pros	Cons
• Up to 30% faster development process, as the same code base can be used on multiple platforms	• Harder to debug, because React Native is developed using Javascript, Java, Objective-C, and C / C ++
• Reduction of the development cost as the company needs to hire one development team	• Inferior Android support
• Seamless integration with Native solutions	• Harder to test (compared to Native apps)
• Rich ecosystem with a variety of libraries and tools, which boosts productivity	• Requires creating a mechanism between Javascript and the native world, as React Native is not a native solution
• Support of the external tools like CircleCI, Bitrise, and Code Magic	• Creating a team of well-versed developers both for Native and Web technologies are challenging

Flutter

Pros	Cons
• Fast reloading, as any updates to the code, appear fast, which helps to debug big time	• Young tech, so it lacks advanced features
• Good portability allows running on multiple devices	• Dart, Flutter language, is a relatively new language that lacks many features, making it harder to make changes
• Faster development time, due to code reusability	• No 3-parties support
• Easy customization with different overlays, videos, graphics, pics, etc.	• Takes longer time to download and update
• Seamless integration and fast control of every pixel on the screen	• Flutter takes a lot of space

Ionic

Pros	Cons
• Single code-base shorten development time and costs	• Dubigging is time-consuming
• Ionic brings native app UI standards and device functionalities providing open web power and flexibility	• Local storage is vulnerable
• Allows adaptive style, as users can choose different app styles depending on the platform it will run on	• No hot reloading
• Built-in testing and debugging tools	• Big app size, as developing with CSS, HTML, and JavaScript adds weight to the program, yet pro developers no tricks to make it more light-weight

Apache Cordova

Pros	Cons
• Apache Cordova allows making integrated payments	• Performance latency
• Single code-base	• To make the app seem more native, the development takes more time, as there is a lack of pre-built-in transitions
• Better user experience, with a variety of UI libraries	• No the best option for gaming apps or applications that require a lot of graphics
• Flexibility results in faster development process	
• Large support community of Apache Cordova	

Benefits of Cross-Platform Tools

- Single code-base for iOS and Android Applications
- Better value for money
- Fast development process & time-to-market
- Consistency of UI/UX across platforms

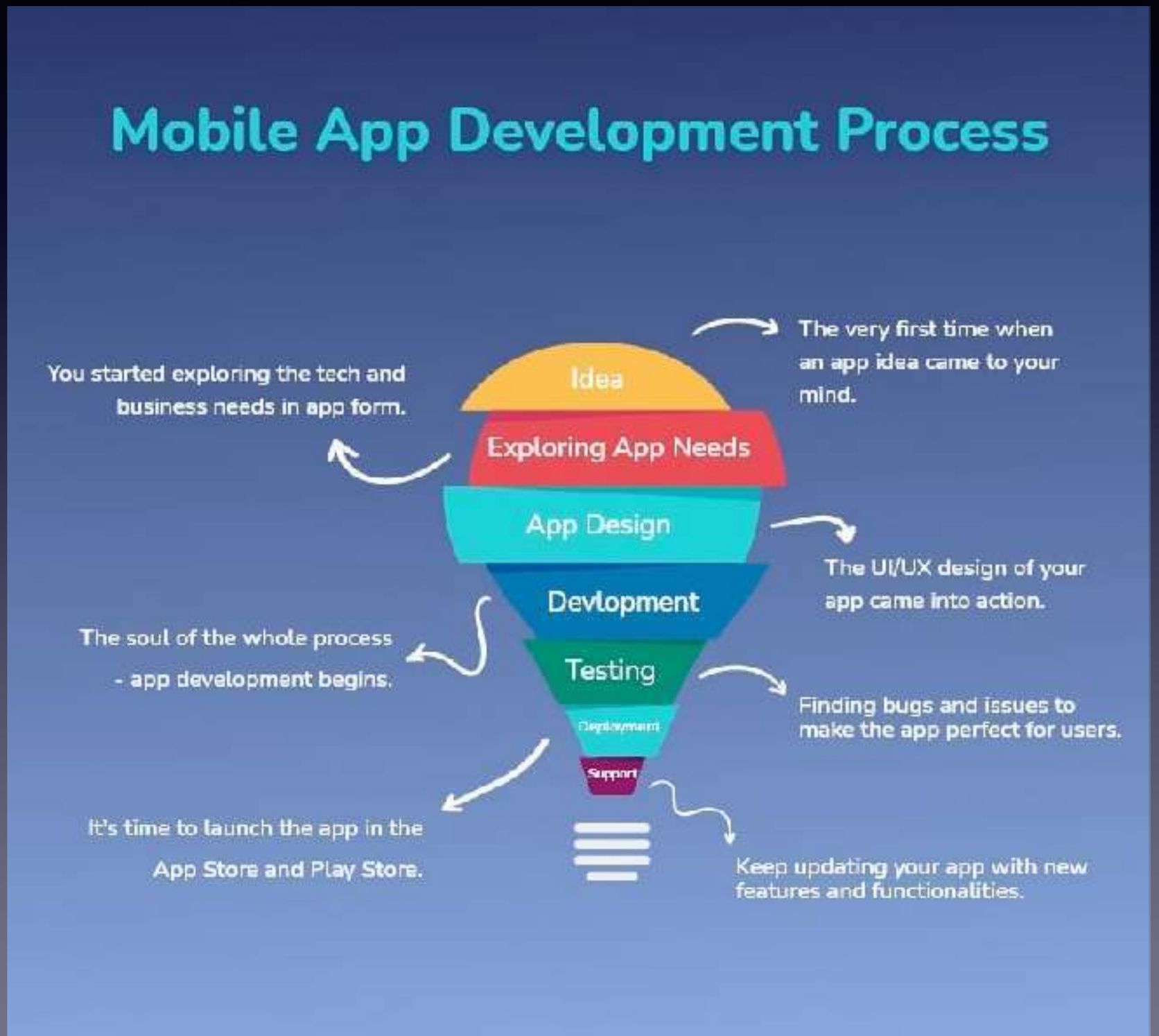
Drawbacks of cross-platforms

- Functionality limits
- The app is usually slower
- Longer wait time for updates & new features

Mobile App Development

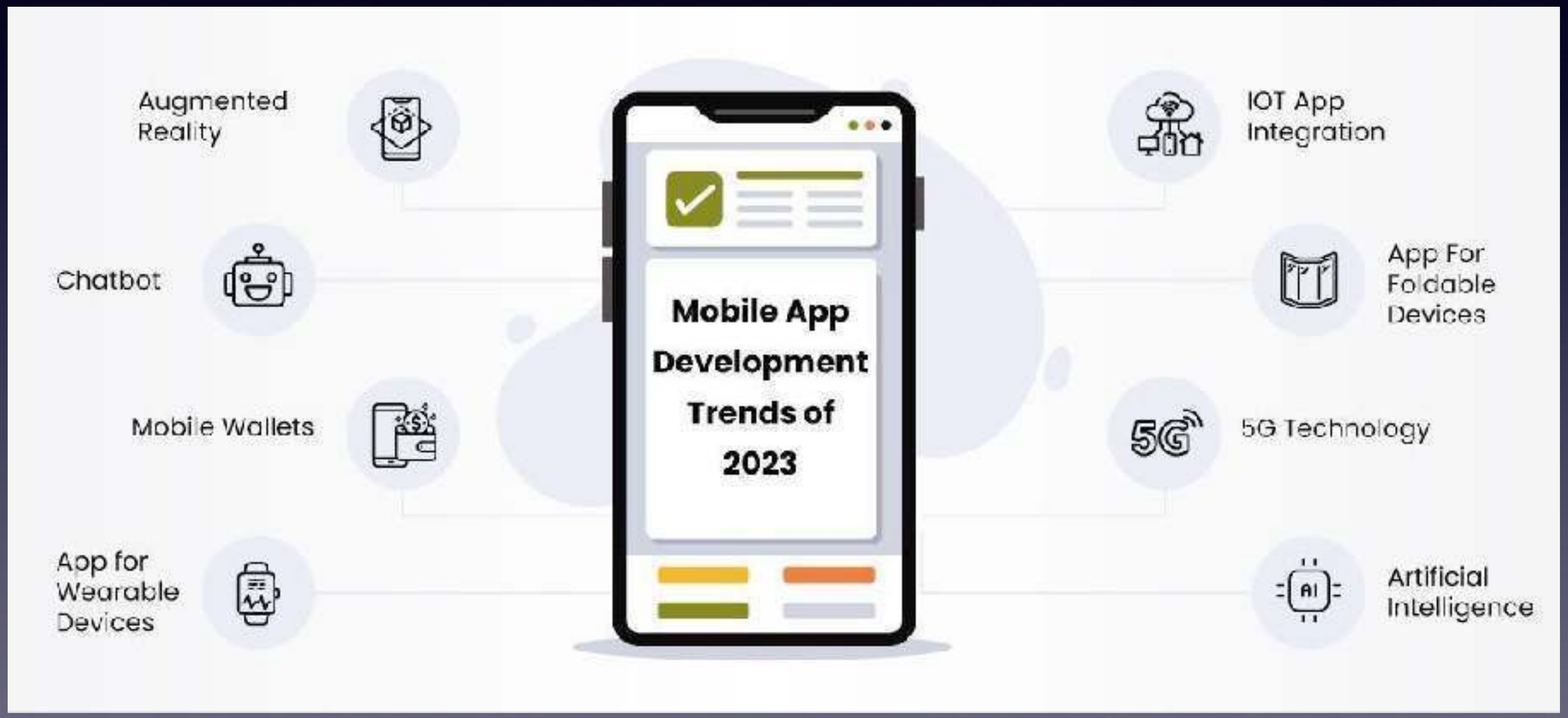
Mobile App Development

Mobile App Development



Mobile App Development Lifecycle

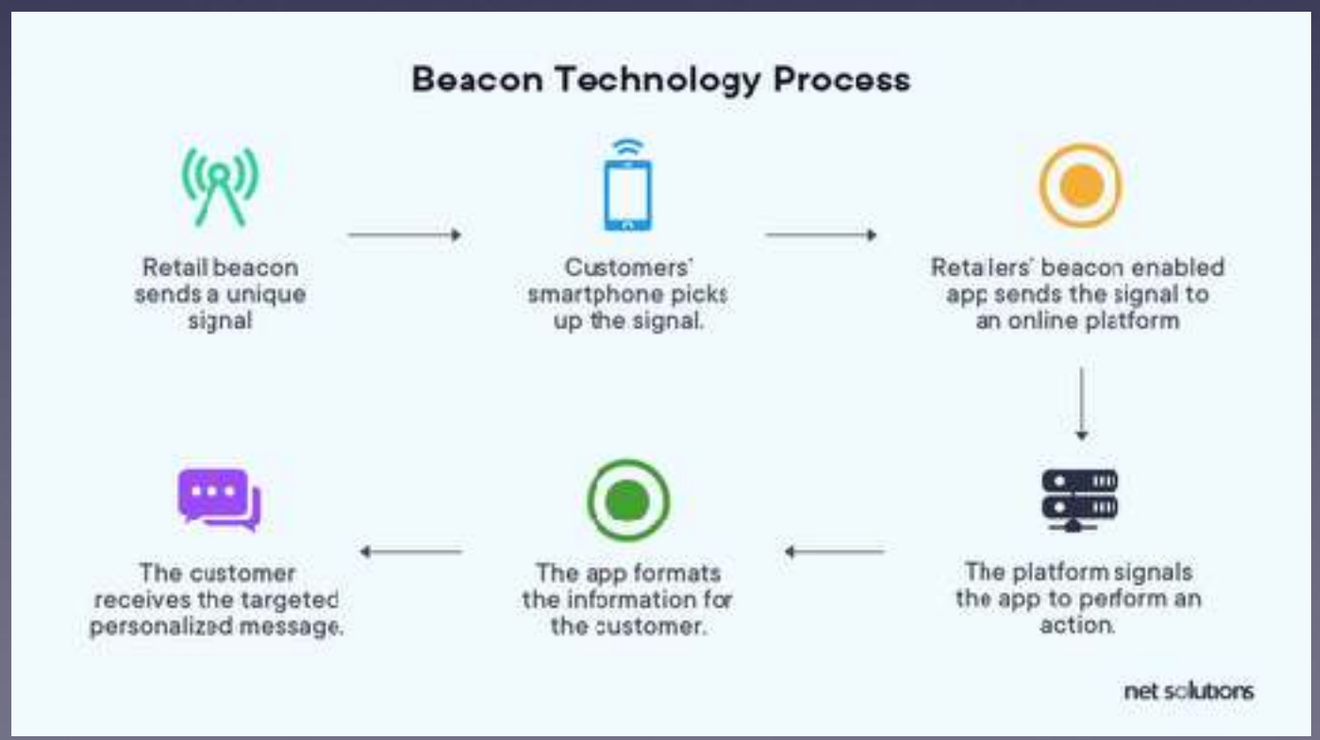
Mobile App Development Trends



Mobile App Development Trends

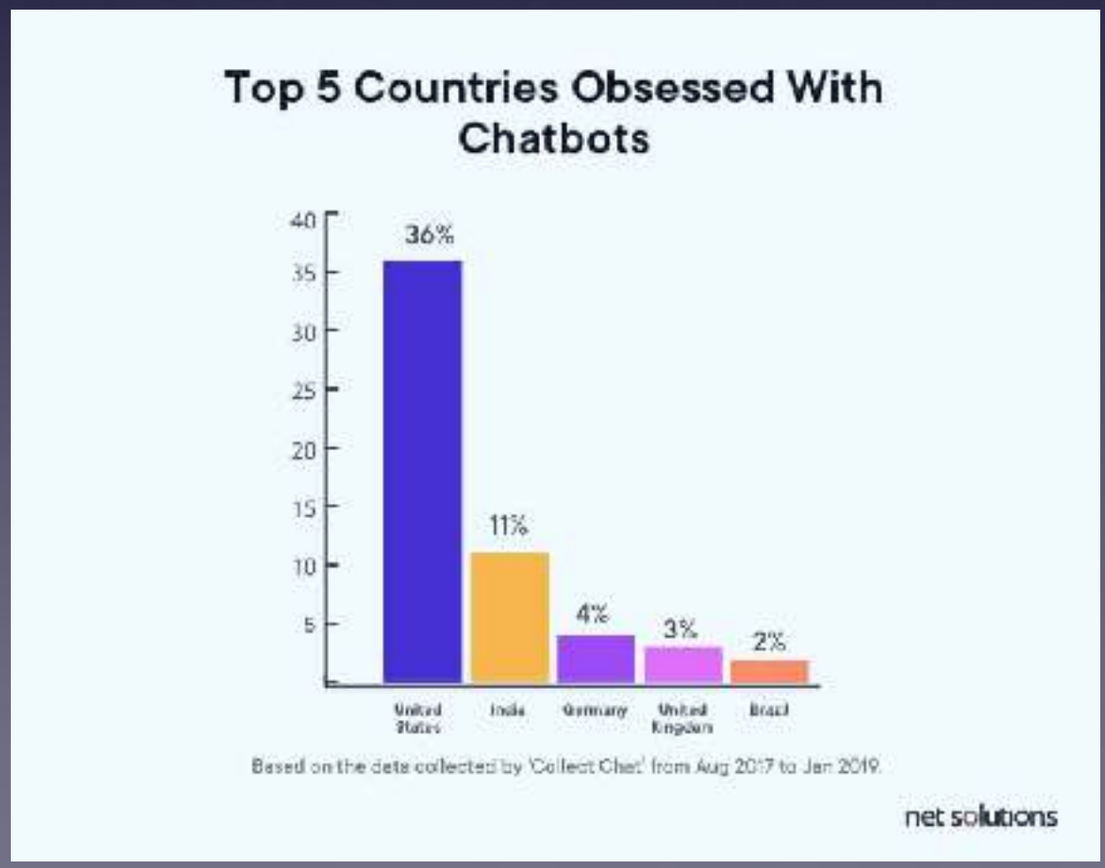
Mobile App Development Trends

- Beacon Technology
- Instant Apps are Coming Around
- More Demand for On-Demand Apps
- Cloud Storage Development

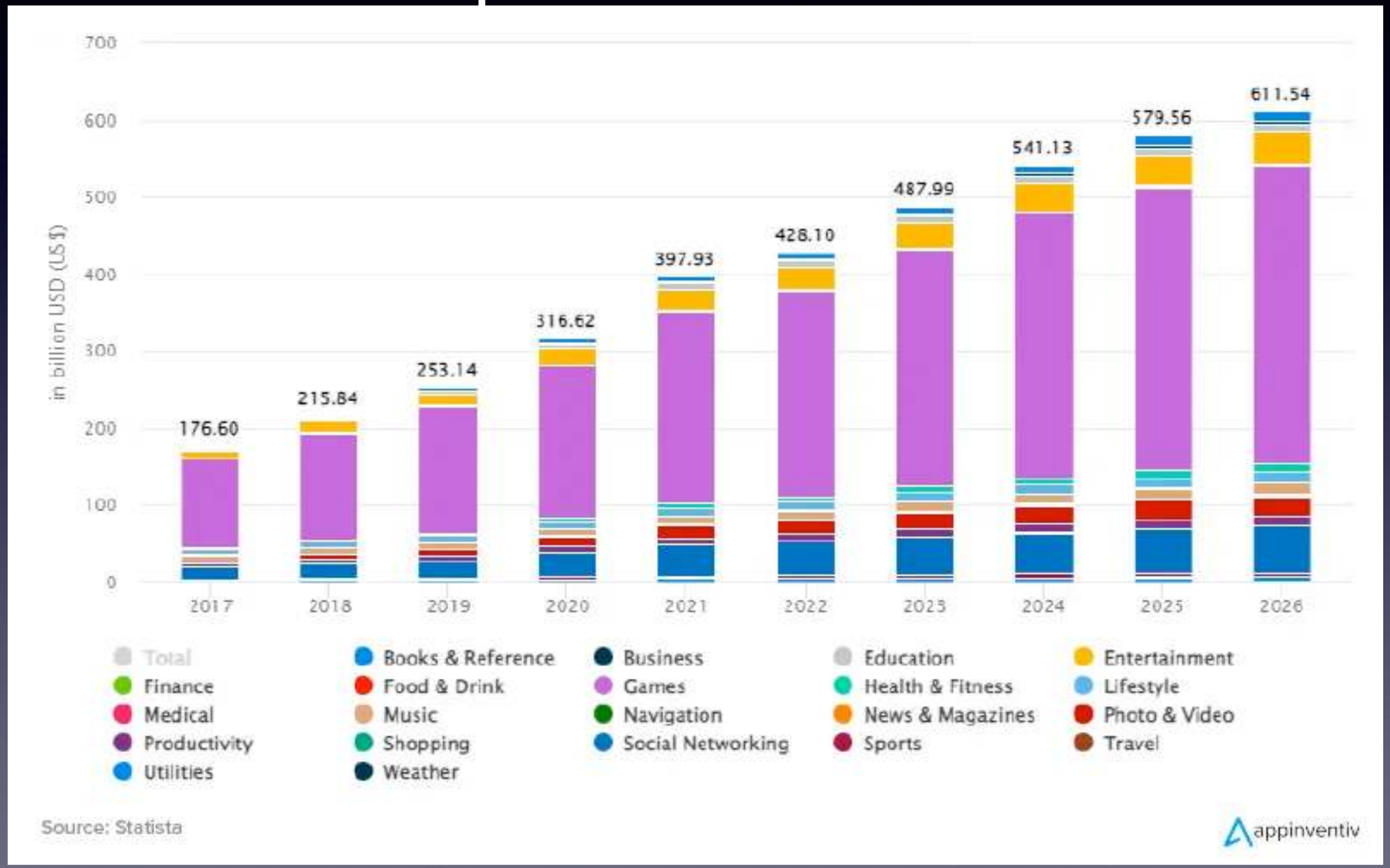


Mobile Application Development Trends

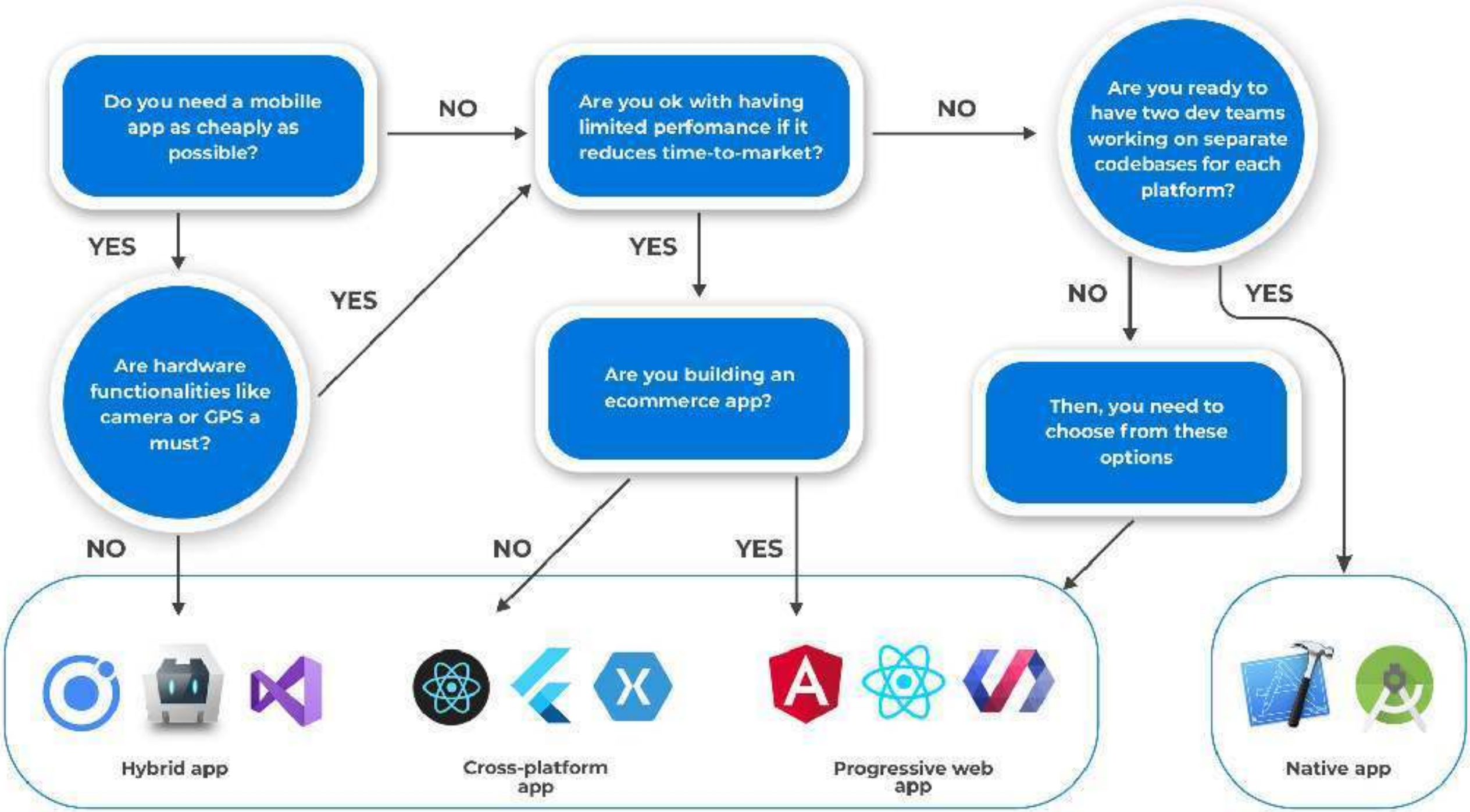
- Building Apps for Foldables
- Machine Learning (ML) and Artificial Intelligence (AI)
- The Rise of Chatbots
- Wearable App Integration
- IoT-Enabled Mobile Apps
- Augmented Reality (AR) and Virtual Reality (VR)
- Mobile Wallets
- Mobile App Security to Gain Extra Attention



Future Mobile App Development Trends



CHOOSE A DEV APPROACH FOR YOUR MOBILE APP



iOS vs Android App Development

iOS Benefits & Challenges	Android Benefits & Challenges
✓ iOS faster time to ship	✓ Flexibility and customization
✓ iOS cheaper timeline to bring to market	✓ Faster approval process
✓ iOS simpler coding languages	✓ Cheaper developer account fee
✓ Users with higher average spending power	✓ More robust developer resources
✗ Higher developer account fee	✓ Larger global market share
✗ Closed-source code	✗ More complicated to bring to market
✗ Higher publishing costs	✗ Increased fragmentation among devices, screens, resolutions
	✗ Increased chances of app release with bugs/ vulnerabilities

iOS vs Android App Development

- Budget
- Publishing
- Programming Languages
- Developer Resources
- Open vs. Closed-Source Code
- Fragmentation
- Market Share and Audience Demographics
- Engagment Opportunities and Support

Instant Apps

- You can instantly open and run the app without needing to install it.
- The first (and main) friction point (installing the app) is removed.
- Discoverability of an app massively increases by opening up a new way to find the app and experience it.
- Links to instant apps are easily shared – there are no barriers to entry.
- Instant apps help build first impressions about the main app.
- For the available features, they provide exactly the same product experience as the installable app, but through a URL.
- Instant apps are good for very small apps but also work well for bigger ones, such as shopping apps, podcast apps, and news apps. This is because when you click on the link, it only downloads the required module for your request.
- If you're using a device that runs Android 8.0 (API level 26) or higher and your app specifies a `targetSandboxVersion` of 2, then your data is transferred automatically to the full app version.

Limitations of Android Instant Apps

- Instant apps only work on Android 5.0 (API level 21) or higher.
- Instant apps run in a sandbox environment (a special kind of SELinux sandbox), and there's no 1-to-1 mapping of API behavior between the regular app and the instant app.
- There's a size limit of 15 MB (previously 4 MB).
- Binary should be as small as possible to load fast – use modules for each entry point and load only what's needed.
- Multiple activities are needed for multiple entry points.
- It's not possible to send and receive system broadcasts.
- Instant apps offer limited supported permissions and operations.
- Limited storage access.
- You can't share things like images with other apps.
- Instant apps can't create background services.

On-Demand Apps

Lower costs

On-demand platforms offer lower prices, compared to traditional providers. And cost's often the main driver for buyers.

Availability

Smartphones and tablets are everywhere, so any service with a reliable mobile platform is handy, accessible, and one-click easy to get.

Adjustment to users' preferences

On-demand businesses know what their users want, and they aim to deliver services as fast as possible. For instance, with Jiffy, a home-service platform, you'll get a plumber to fix the tap within 2-3 hours, not days.

Various payment methods

Mostly, on-demand apps use Stripe or Braintree payment gateways or e-wallets like PayPal. All these tools are highly secured, fast and reliable. Besides, on-demand delivery apps accept cash as well.

Minimum efforts

There's an option to save your details like payment info, delivery address, and so on. No need to enter this information again and again with each new order.

Variety

Tens of apps in each category, from babysitting to make-up service, meaning each user can find a solution that works best for them.

Mobile Application Success Metrics

- Performance Metrics
- User Metrics
- Engagement Metrics
- Business Metrics

Performance Metrics

- App Crashes
- API Latency
- End-to-End Application Latency
- App Load per Period
- Network Errors

User Metrics

- Monthly/Daily Active Users
- Device and OS Metrics
- Geo Metrics

Engagement Metrics

- Session Length
- Session Interval
- Retention Rate

Business Metrics

- Acquisition Cost
- Transaction Revenue
- Abandonment Rate
- LTV
- App Star Rating

Metric: Acquisition

KPI	What to track	Why
• New downloads	• No. of new downloads: daily, weekly, monthly	• Track your growth
• Download attribution	• Which channels your new users come from	• Use this data to optimize marketing spend and improve ROI/customer LTV

Metric: Activation

KPI

- Activation rate

What to track

- Percentage of downloads that launch the app

Why

- 85% and up: good. 84% and lower: fix the issue

Stats

- For new apps, ratio of 1st time app launch to total app launches over a rolling 30-day period = 5 to 15%

Metric: Retention

KPI	What to track	Why	Stats
• Day 3 and day 7 retention after 1st app launch	• No. of users who keep using app 3 and 7 days after 1st launch	• See whether your users are finding value in the app	• 24% of apps are uninstalled (or not used again) after just one launch
• Weekly and monthly retention cohorts	• How long each cohort stays active in your app	• See which in-app behaviors have a positive or negative effect on retention	

Metric: Engagement

KPI	What to track	Why	Stats
• DAU/MAU	• No. of active users on a given day or month	• See if your app "hooks" users	• Not all apps are meant to be used daily. Define what "active user" means.
• Avg. session length and frequency	• How often users are launching your app, & how long an average session lasts	• See the effectiveness of your engagement campaigns & the quality of your user experience	

Metric: Uninstalls

KPI	What to track	Why
• Daily/weekly uninstalls	• No. of uninstalls	• Get a net result of your mobile app growth.
• Churn rate	• (Users at start – users at end of period) / users at start	• More accurate forecast for growth & revenue.

Metric: Drop offs

KPI	What to track	Why	Stats
• Conversion rate	• Users who performed a key action / users who could have taken the action	• See if users are performing key activities in your app	
• Cost per conversion	• Total cost of user acquisition / number of conversions	• See how much it costs	
• CAC-to-Conversion	• Cost of acquiring a paying customer	• To improve ROI	• To improve ROI Ecommerce: $30-100 per user Subscriptions: $100+ per user

Metric: Reachability

KPI	What to track
• Click-through rates	• % of users who tap on push notifications, in-app notifications, and email links
• Opt-in rate	• % of users who opt-in to receive notifications or allow location access