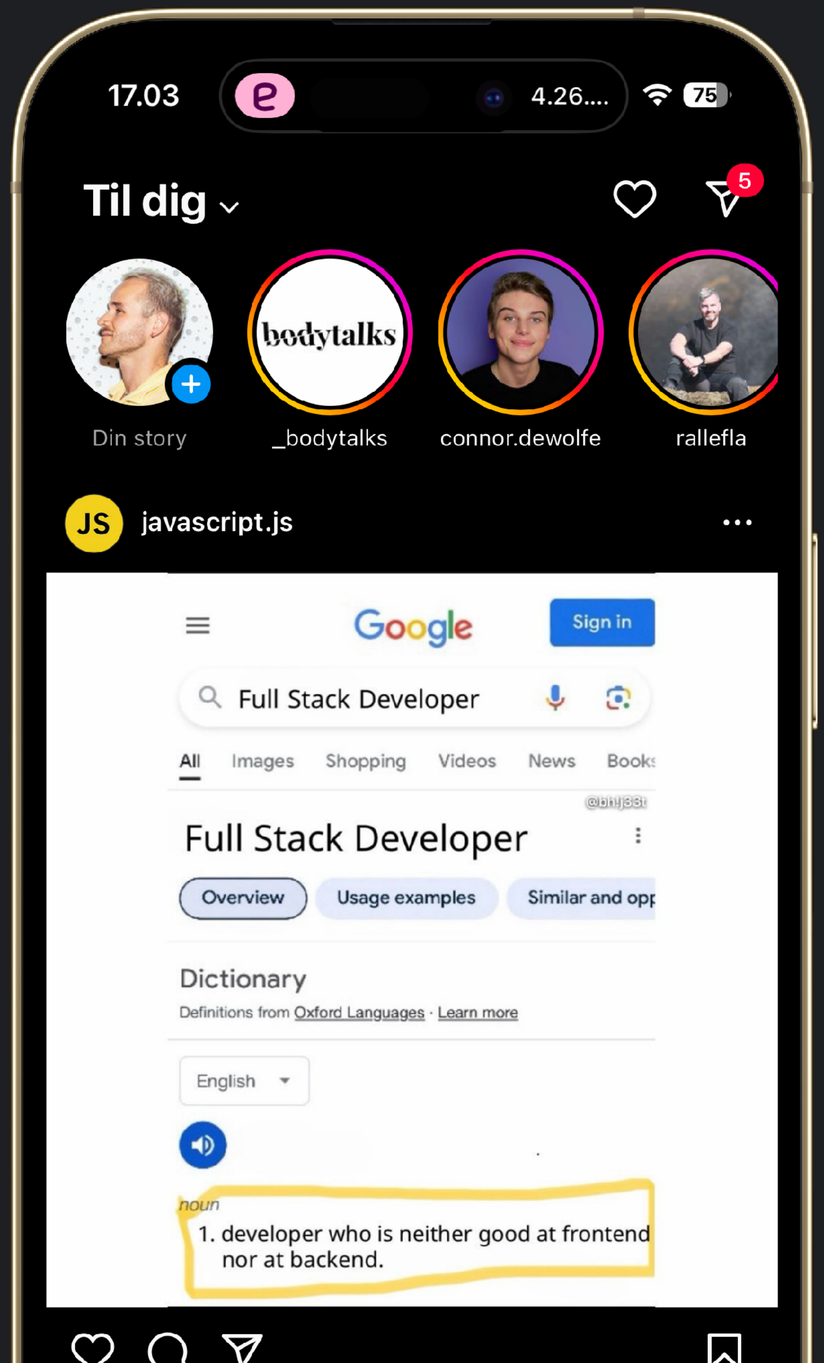
Er du

Fullstack Developer?



Progressive Web App

“[...] is basically just a website with some added features, which enable it to provide an app-like user-experience. “

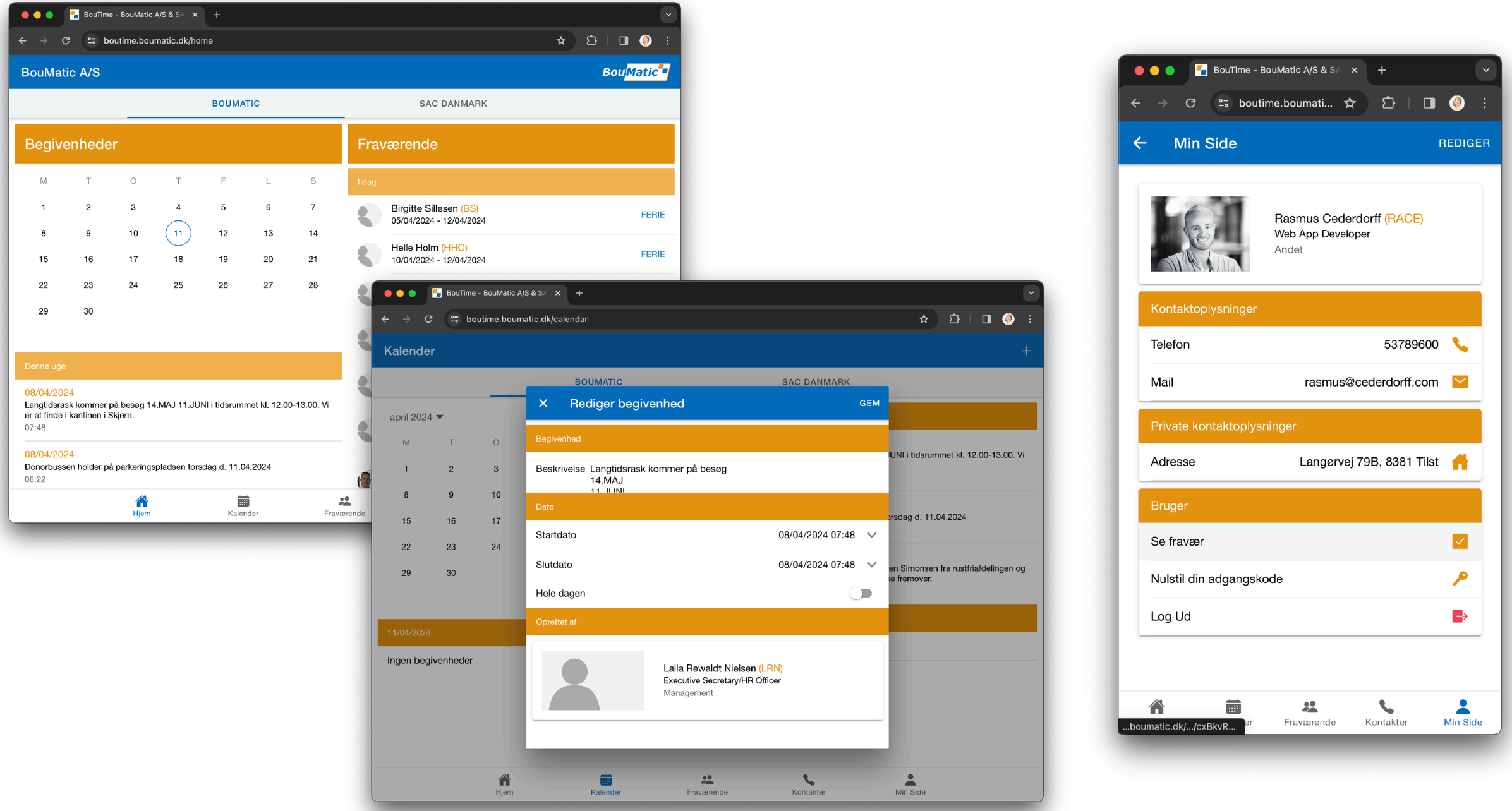
https://whatpwacando.today/info

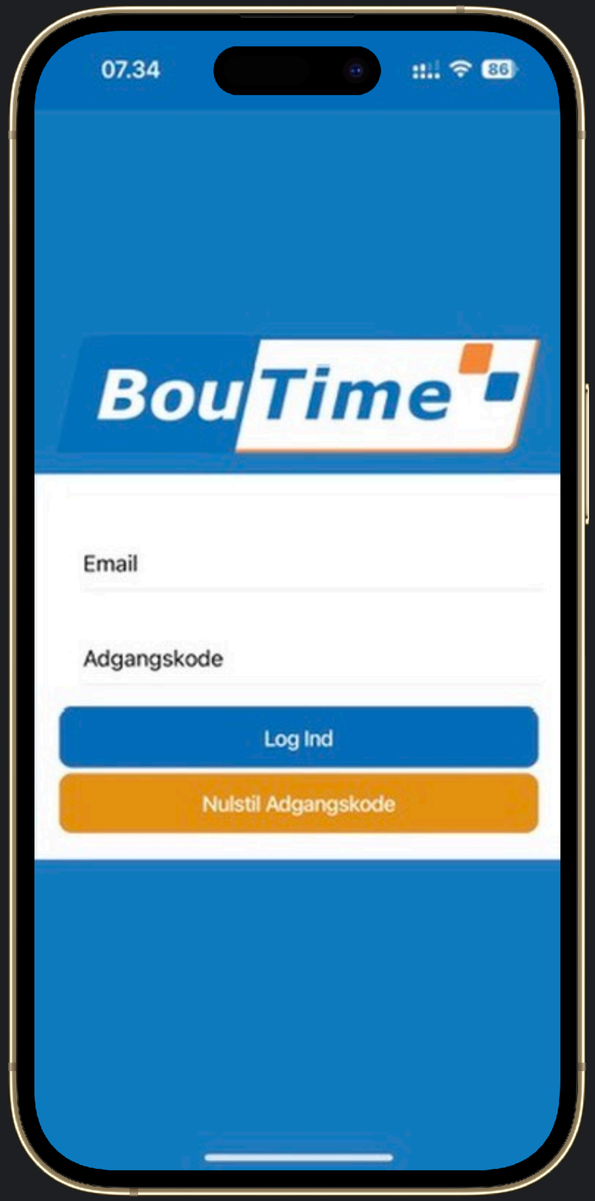
“This means it can work practically just like a native iOS or Android app. It can be installed to the home screen of your mobile device, work offline and receive push notifications, among other things.”

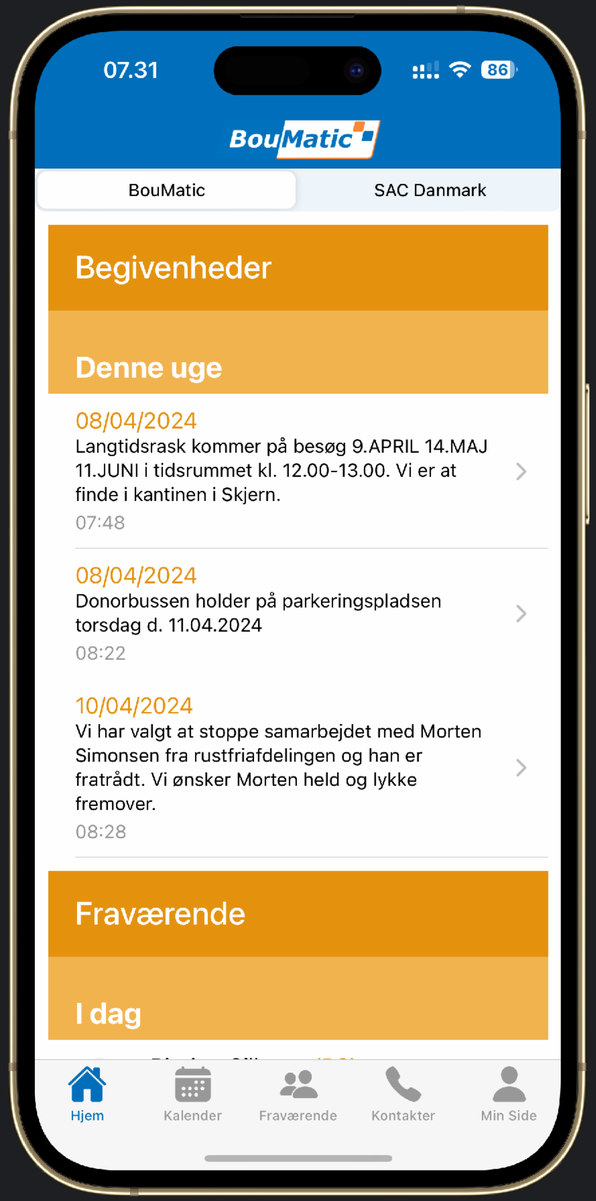
https://whatpwacando.today/info

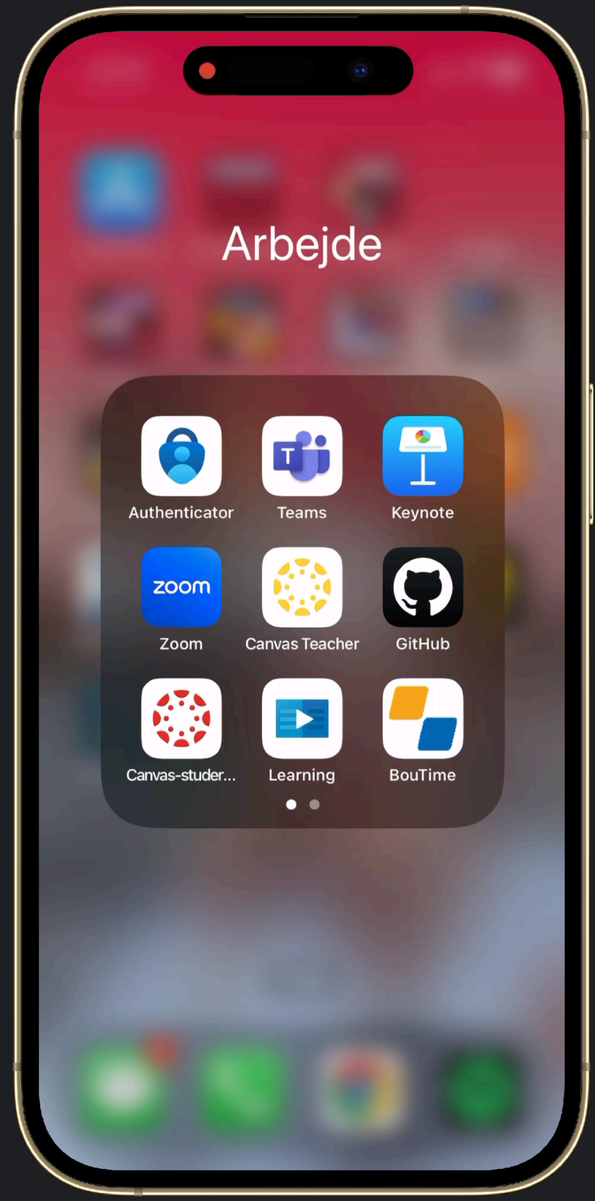
https://www.youtube.com/watch?v=sFsRylCQblw


BouTime


BouTime - BouMatic A/S & SA
boutime.boumatic.dk/home
BouMatic A/S
BouMatic
BOUMATIC
SAC DANMARK
Begivenheder
M T O T F L S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
Denne uge
08/04/2024
Langtidsrask kommer på besøg 14.MAJ 11.JUNI i tidsrummet kl. 12.00-13.00. Vi er at finde i kantinen i Skjern.
07:48
08/04/2024
Donorbussen holder på parkeringspladsen torsdag d. 11.04.2024
08:22
Hjem
Kalender
Fraværende
Fraværende
I dag
Birgitte Sillesen (BS)
05/04/2024 - 12/04/2024
FERIE
Helle Holm (HHO)
10/04/2024 - 12/04/2024
FERIE
BouTime - BouMatic A/S & SA
boutime.boumatic.dk/calendar
Kalender
BOUMATIC
SAC DANMARK
april 2024
M T O
1 2 3
8 9 10
15 16 17
22 23 24
29 30
Rediger begivenhed
GEM
Begivenhed
Beskrivelse Langtidsrask kommer på besøg
14.MAJ
Dato
Startdato 08/04/2024 07:48
Slutdato 08/04/2024 07:48
Hele dagen
Oprettet af
Laila Rewaldt Nielsen (LRN)
Executive Secretary/HR Officer
Management
11/04/2024
Ingen begivenheder
Hjem
Kalender
Fraværende
Kontakter
Min Side
BouTime - BouMatic A/S & SA
boutime.boumati...
Min Side
REDIGER
Rasmus Cederdorff (RACE)
Web App Developer
Andet
Kontaktoplysninger
Telefon 53789600
Mail rasmus@cederdorff.com
Private kontaktoplysninger
Adresse Langørvej 79B, 8381 Tilst
Bruger
Se fravær
Nulstil din adgangskode
Log Ud
...boumatic.dk/.../cxBkvR...
Fraværende
Kontakter
Min Side


07.34
BouTime
Email
Adgangskode
Log Ind
Nulstil Adgangskode


07.31
BouMatic
BouMatic
SAC Danmark
Begivenheder
Denne uge
08/04/2024
Langtidsrask kommer på besøg 9.APRIL 14.MAJ 11.JUNI i tidsrummet kl. 12.00-13.00. Vi er at finde i kantinen i Skjern.
07:48
08/04/2024
Donorbussen holder på parkeringspladsen torsdag d. 11.04.2024
08:22
10/04/2024
Vi har valgt at stoppe samarbejdet med Morten Simonsen fra rustfriafdelingen og han er fratrådt. Vi ønsker Morten held og lykke fremover.
08:28
Fraværende
I dag
Hjem
Kalender
Fraværende
Kontakter
Min Side


Arbejde
Authenticator
Teams
Keynote
Zoom
Canvas Teacher
GitHub
Canvas-studer...
Learning
BouTime

(Re)Use the expertise you already have in-house

Traditional Approach

Web (Native) Approach

Combining the best of web and native apps

“[...] feels as good to use as a platform-specific application [...]”

https://web.dev/articles/what-are-pwas

One Codebase
Any Platform

What are PWAs?

The three pillars of PWA design

Reliable

Works offline and performs well on low-quality networks.

Capable

Web apps are more capable than ever given modern APIs.

Installable

Runs as a standalone window and launches from the home screen.

Website, Web App, Native App, Hybrid App & Web Native

Website

The user wants to **get** something!

Web App

The user wants to **do** something!

Remix Post App

localhost:3000/posts

POSTS

Posts

Dan Okkels Brendstrup
Lecturer

A cozy morning with coffee

Likes: 785

Tags: morning, coffee, cozy, food

Rasmus Cederdorff
Senior Lecturer

Serenity of the forest

Likes: 799

Tags: forest, nature, serenity

Maria Louise Bendixen
Senior Lecturer

A beautiful morning in Aarhus

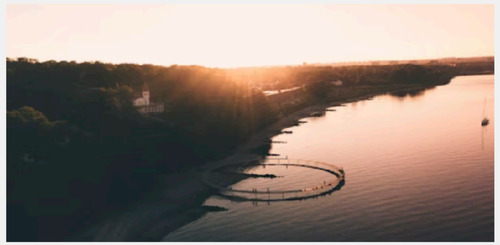
Likes: 99

Tags: morning, Aarhus, beautiful, AROS

Anne Kirketerp
Head of Department

Rasmus Cederdorff
Senior Lecturer

Maria Louise Bendixen
Senior Lecturer



Remix Post App

localhost:3000/posts

POSTS | ADD POST | PROFILE

Posts

Serach by caption | Filter by tag | Sort by

Search | select tag | newest

Dan Okkels Brendstrup
Lecturer

A cozy morning with coffee

Likes: 785

Tags: morning, coffee, cozy, food

Rasmus Cederdorff
Senior Lecturer

Serenity of the forest

Likes: 799

Tags: forest, nature, serenity

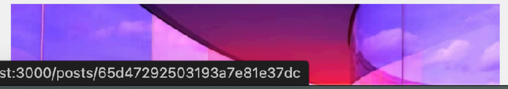
Maria Louise Bendixen
Senior Lecturer

A beautiful morning in Aarhus

Likes: 99

Tags: morning, Aarhus, beautiful, AROS

Anne Kirketerp
Head of Department



Rasmus Cederdorff
Senior Lecturer

Maria Louise Bendixen
Senior Lecturer

localhost:3000/posts/65d47292503193a7e81e37dc

Web App

- Any program executed in the browser.
- Any program that performs a specific function by using a web browser as its client.
- The user wants to complete a task - to do something.
- Stored on a web server.
- Accessible anywhere, at any time.
- Build with HTML, CSS & JS.
- Not just about mobile devices.

Types of Web Apps

WEB BASED

BROWSER BASED

API BASED

SERVER APPS

PORTAL APPS

SOFTWARE AS A SERVICE

PROGRESSIVE WEB APP

SPA & MPA

iOS

Android

Native

- Build with native & platform-specific languages & tools.
- iOS - Xcode with Swift or Objective-C
- Android - Android Studio with Kotlin or Java

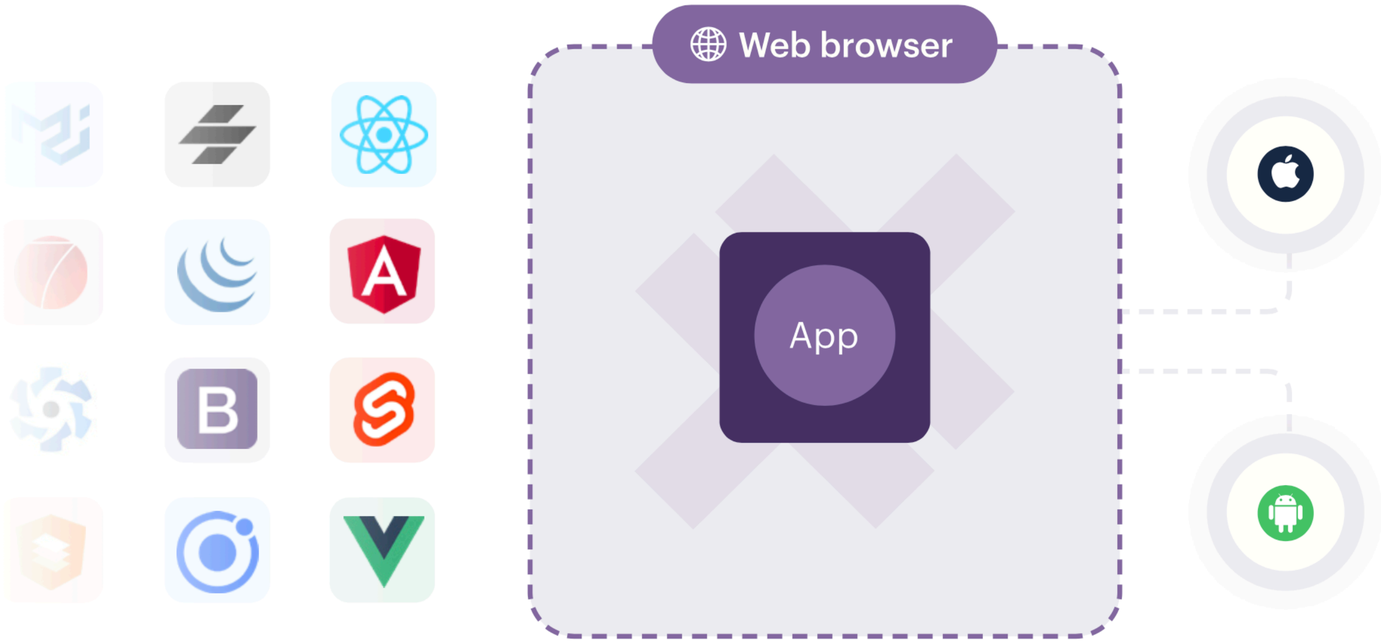
- Full and easier access to the device’s capabilities.
- “tend to also be more performant since their code is closer to the ‘metal’”.
- Native user interface (UI) controls and layouts.
- Android Apps cannot run on iOS and vice versa.

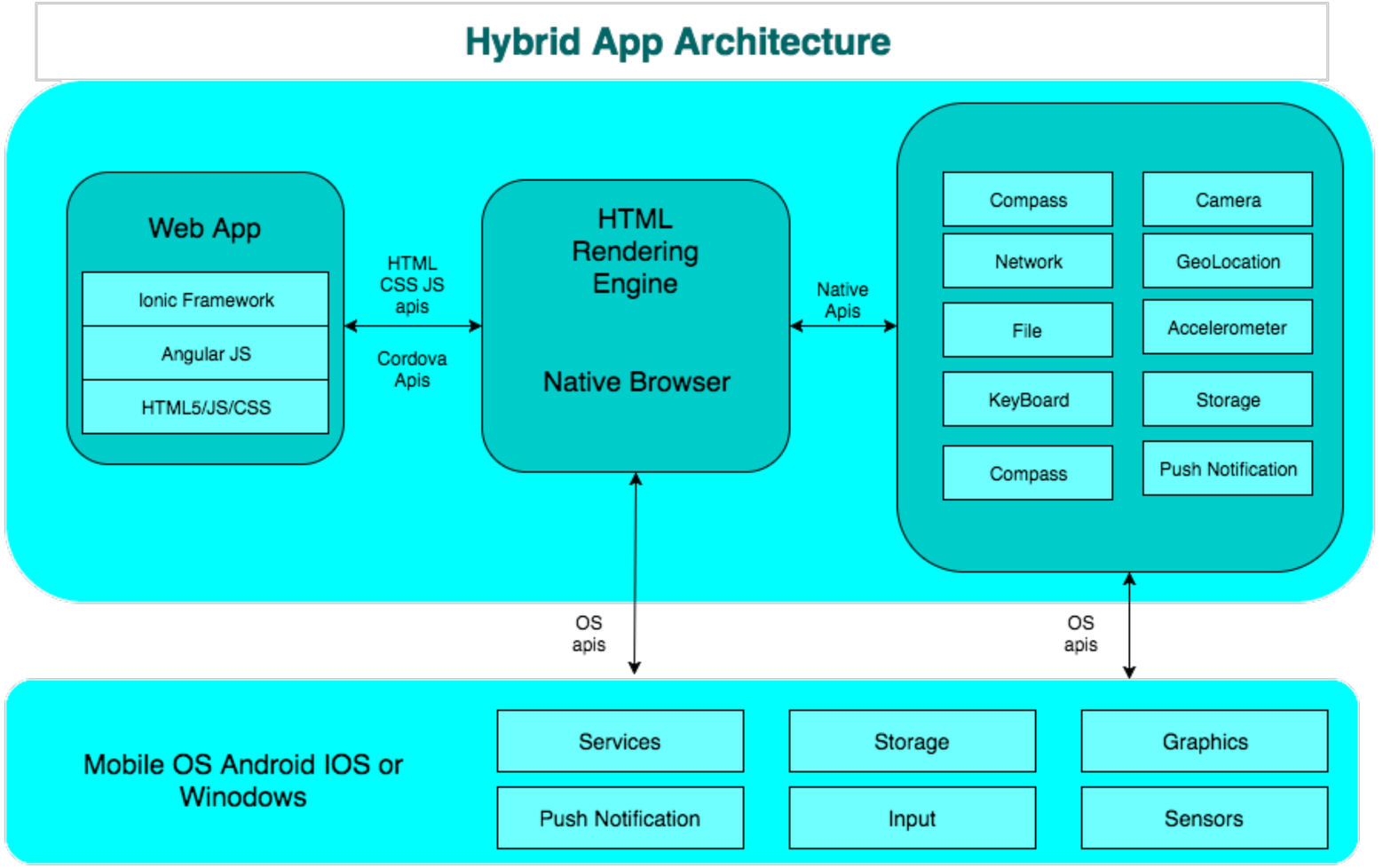
Hybrid

- “Hybrid apps are like native apps. They’re downloaded from the platform’s app store or marketplace and offer the same native features, offline support, and hardware-based performance acceleration as any app built with a native SDK.”

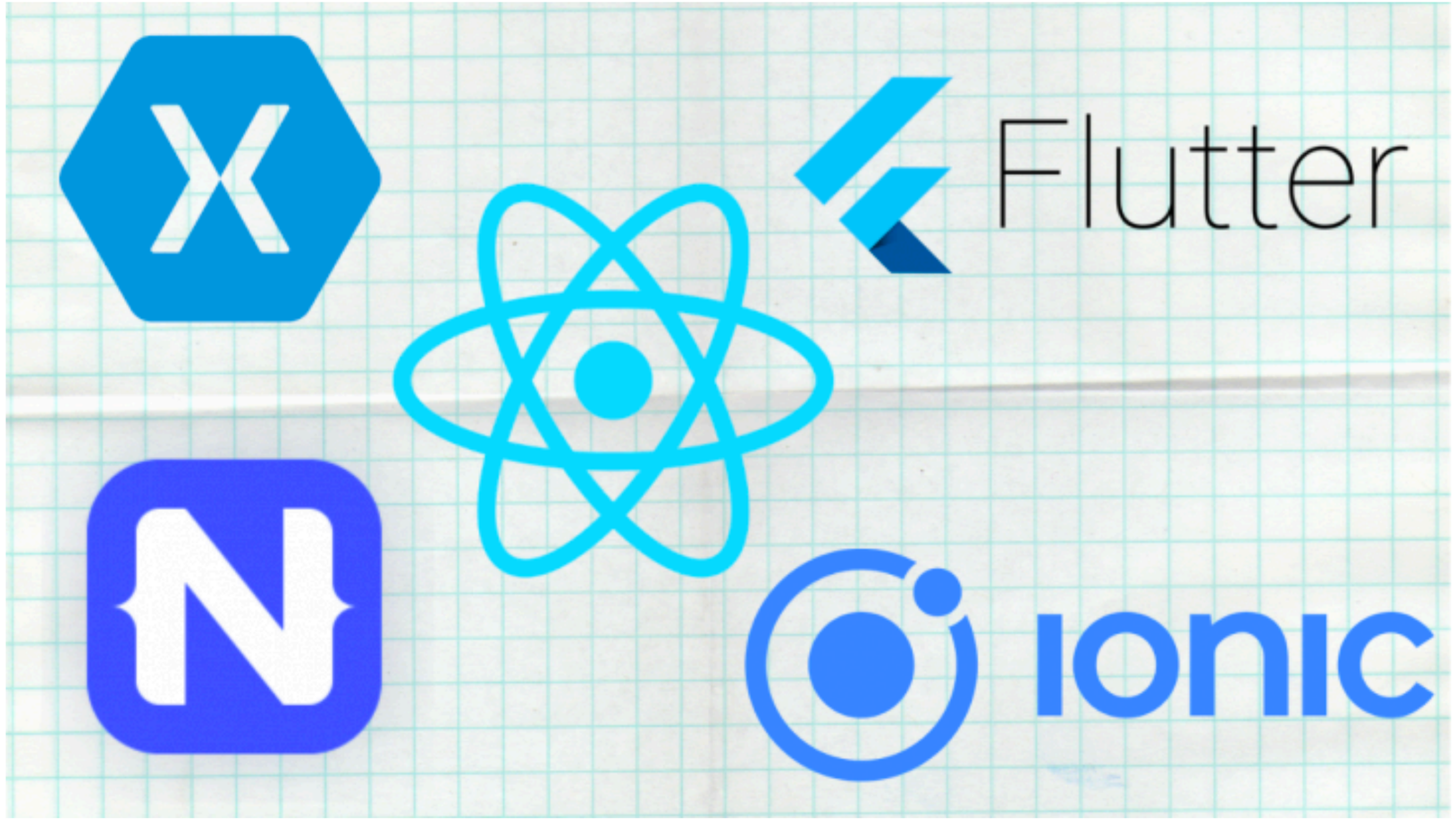
- A blend - both native & web.
- Build with HTML, CSS & JS (+ framework) & cross-platform like Ionic, NativeScript, Xamarin, React Native.

- With plugins: full access to native features.
- Embedded in WebView.
- With a UI Framework: feels and looks like a pure native.
- Bridge & plugins - Cordova & Capacitor.


Web browser
App


Hybrid App Architecture
Web App
Ionic Framework
Angular JS
HTML5/JS/CSS
HTML
CSS JS
apis
Cordova
Apis
HTML
Rendering
Engine
Native Browser
Native
Apis
Compass
Camera
Network
GeoLocation
File
Accelerometer
KeyBoard
Storage
Compass
Push Notification
OS
apis
OS
apis
Mobile OS Android IOS or
Winodows
Services
Storage
Graphics
Push Notification
Input
Sensors

Cross-platforms



Web Native

More than *just* Hybrid

Web Native

Is the full capability and access to developers of the Web Platform, with the full functionality and performance benefits of traditional native apps.

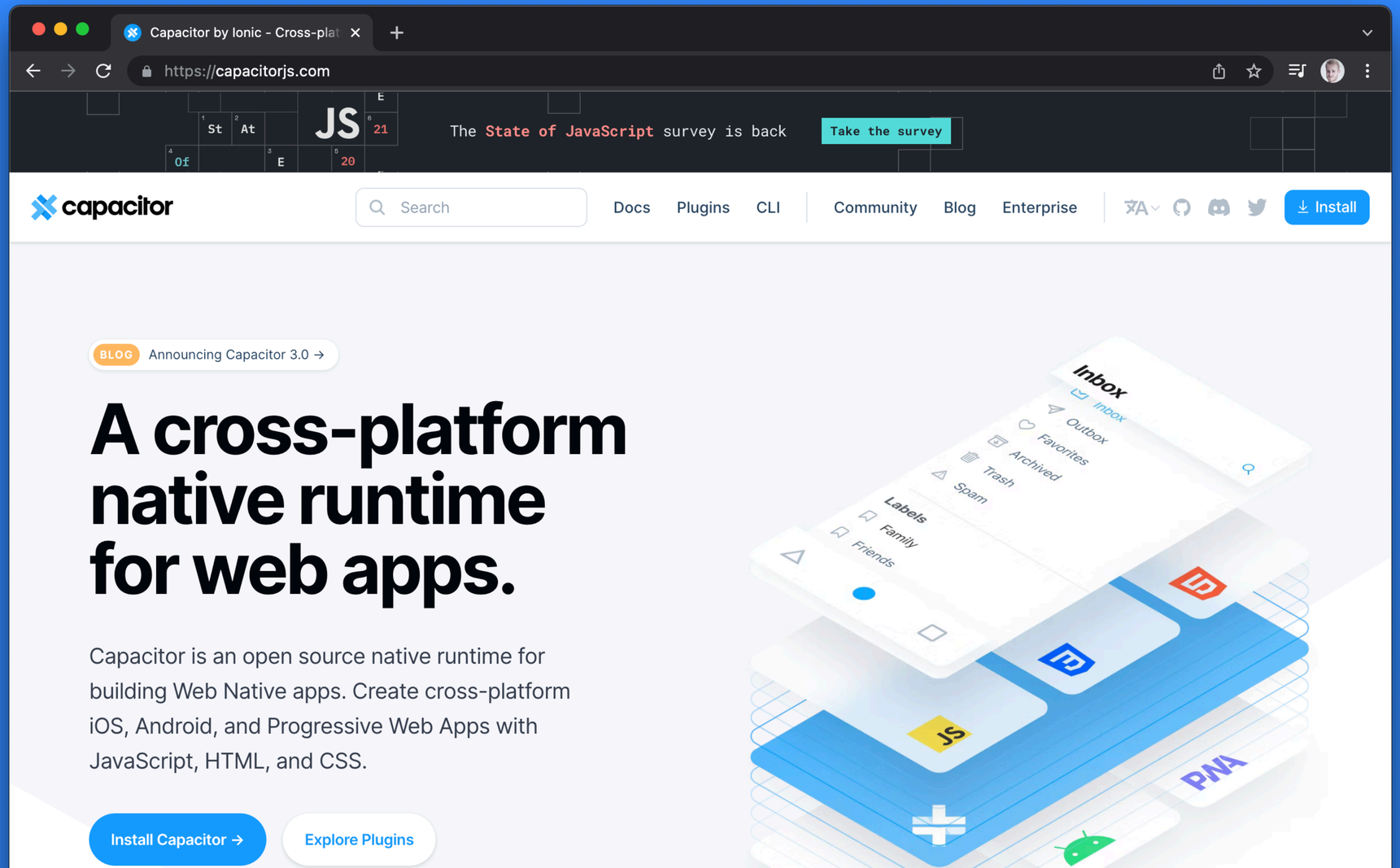
https://webnative.tech/

Web Native is "hybrid" done right, and it's the future of mobile app development.

https://webnative.tech/

Capacitor

A cross-platform native runtime for web apps.



Back in the 2010s

Feature	Native	Web-only	Hybrid
Device Access	Full	Limited	Limited
Performance	High	Medium to High	Low
Development Language	Platform Specific	HTML, CSS, Javascript	HTML, CSS, Javascript
Cross-Platform Support	No	Yes	Yes
User Experience	High	Medium to High	Low
Code Reuse	No	Yes	Yes

Today

Feature	Native	Web-only	Hybrid
Device Access	Full	Limited	Full (with plugins)
Performance	High	Medium to High	Medium to High
Development Language	Platform Specific	HTML, CSS, Javascript	HTML, CSS, Javascript
Cross-Platform Support	No	Yes	Yes
User Experience	High	Medium to High	Medium to High
Code Reuse	No	Yes	Yes

Capacitor

A cross-platform native runtime for Web Native Apps.

The Web View and the native app communicate through the use of Capacitor or Cordova plugins. Plugins provide native APIs such as camera, geolocation, and filesystem access to your web app.

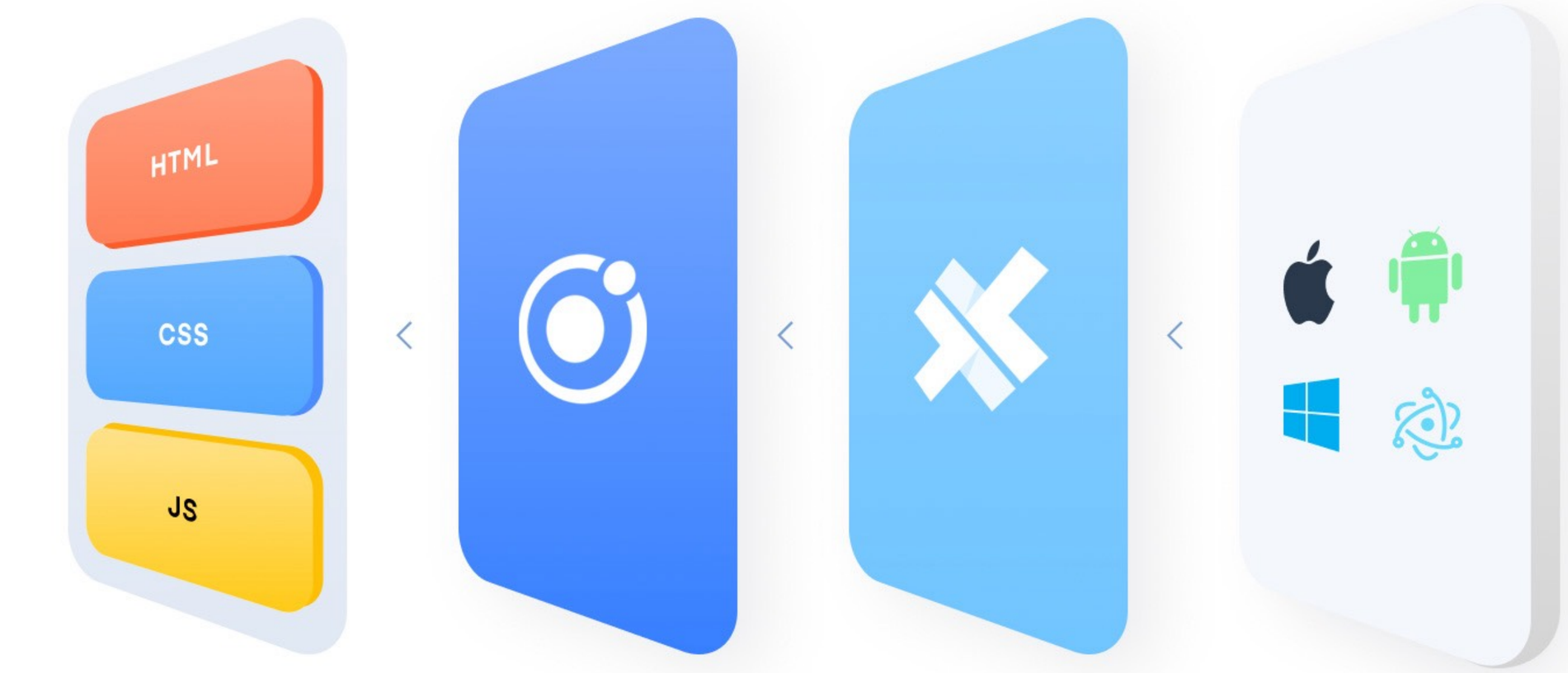
UI (Web View)

JS, HTML & CSS | **Ionic React**

Capacitor

iOS, Android & PWA

https://capacitorjs.com/


HTML
CSS
JS
YOUR APP (ANGULAR, REACT, VUE...)
UI CONTROLS (IONIC)
NATIVE ACCESS (CAPACITOR)
DISTRIBUTION PLATFORMS

Native vs Web Native (Hybrid)

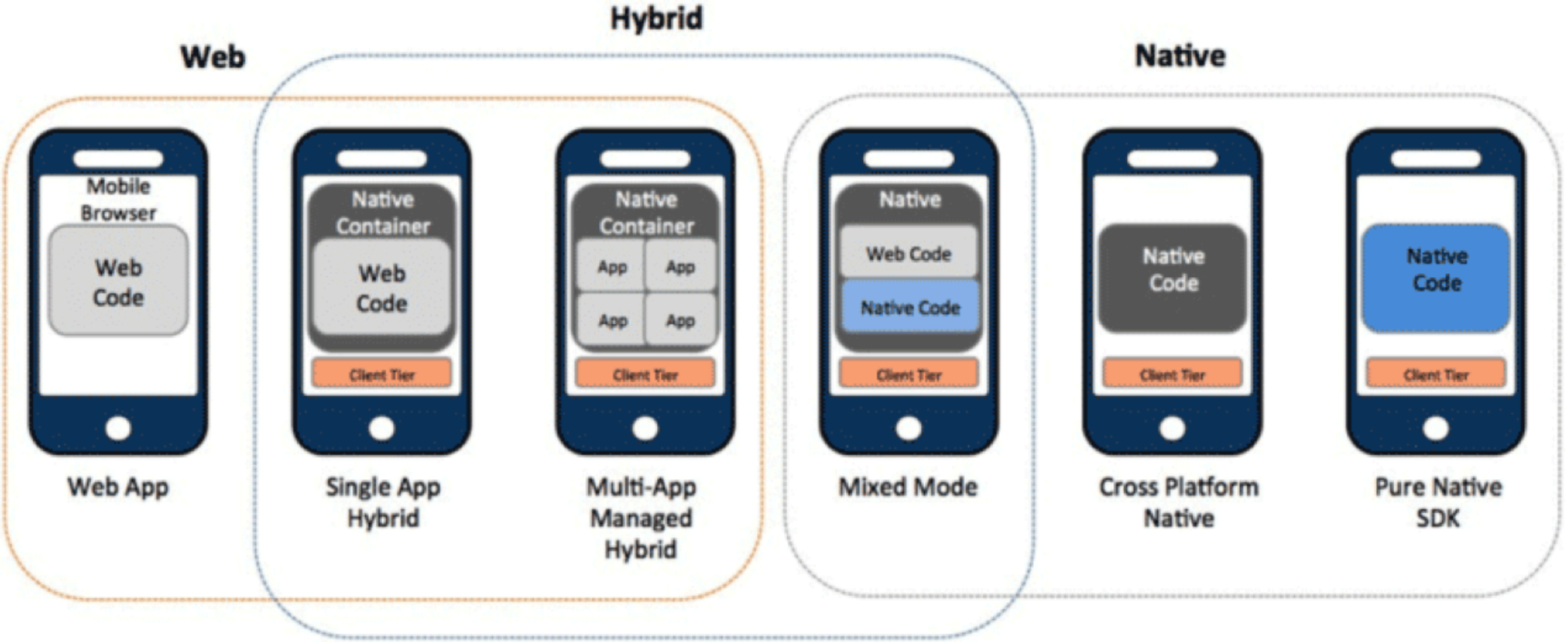
What are the differences? Pros & Cons?

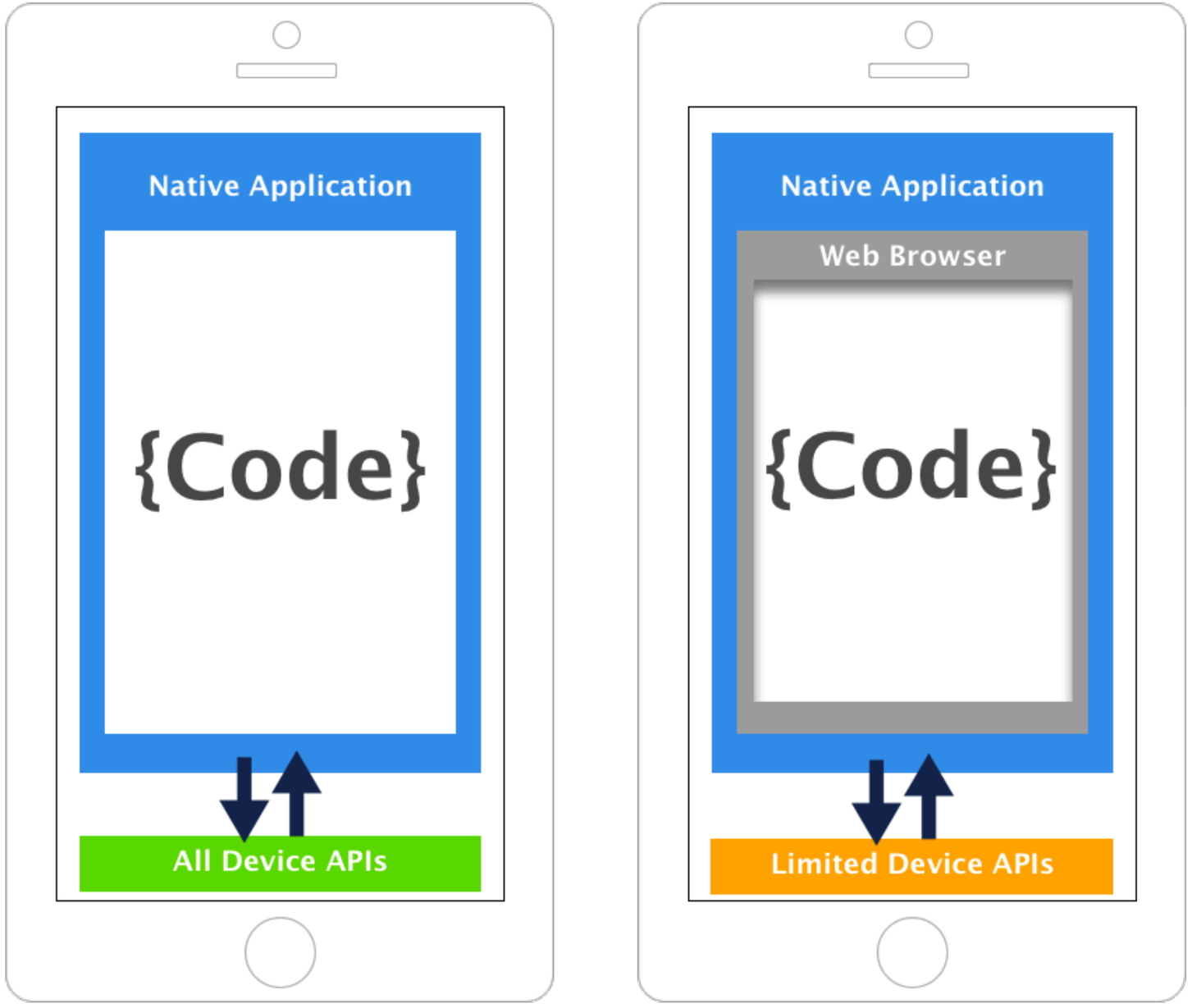
“The key difference is that hybrid apps are built using open web technologies like HTML, CSS, and JavaScript, rather than the proprietary or specialized languages used by iOS, Android, and others. That means anyone with a modern web developer skill-set can begin building an app using the hybrid approach.

Hybrid apps run in a full-screen browser, called a webview, that is invisible to the user. Through customizable native plugins, they can access the native features of specific mobile devices (such as the camera or touch ID), without the core code being tied to that device.

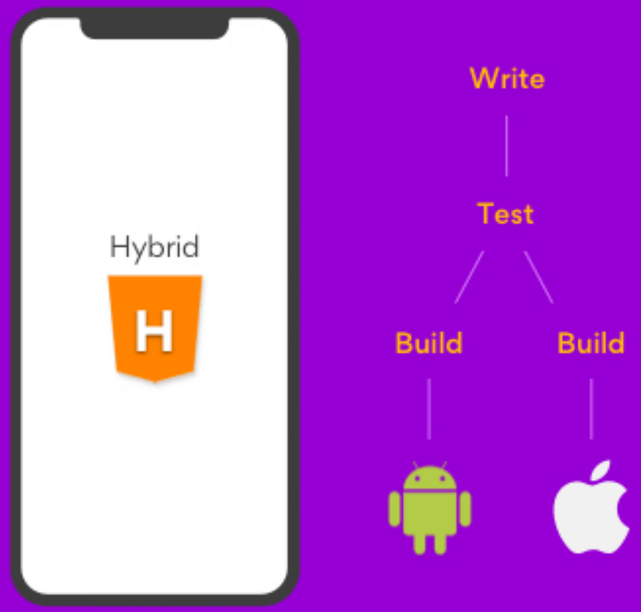
That means cross-platform hybrid applications can run on any platform or device, all from a single codebase, while still delivering native performance.“

https://go.ionic.io/hybrid-vs-native-guide


Hybrid
Web
Native
Mobile Browser
Web Code
Native Container
Web Code
Client Tier
Native Container
App
App
App
App
Client Tier
Native
Web Code
Native Code
Client Tier
Native Code
Client Tier
Native Code
Client Tier
Web App
Single App Hybrid
Multi-App Managed Hybrid
Mixed Mode
Cross Platform Native
Pure Native SDK


Native Application
{Code}
All Device APIs
Native Application
Web Browser
{Code}
Limited Device APIs

HYBRID APPS



NATIVE APPS

Why Native? Why Hybrid?

Hybrid vs Native (ebook): http://go.ionic.io/hybrid-vs-native-guide

Why hybrid?

Let's take a look at each of these.

Write once, run anywhere

Use the talent you already have

Deliver a great user experience across platforms

Build for the future

Progressive Web App

Core Features of PWAs

- **Responsive**: Work on any device (desktop, tablet, smartphone).
- **Connectivity Independent**: Service workers allow work offline or on low-quality networks.
- **App-like**: Feel like a native app due to the use of an app shell model.
- **Fresh**: Always up-to-date due to the service worker update process.

Core PWA Checklist

- Starts fast, stays fast
- Works in any browser
- Responsive to any screen size
- Provides a custom offline page
- Is installable

https://web.dev/articles/pwa-checklist#core

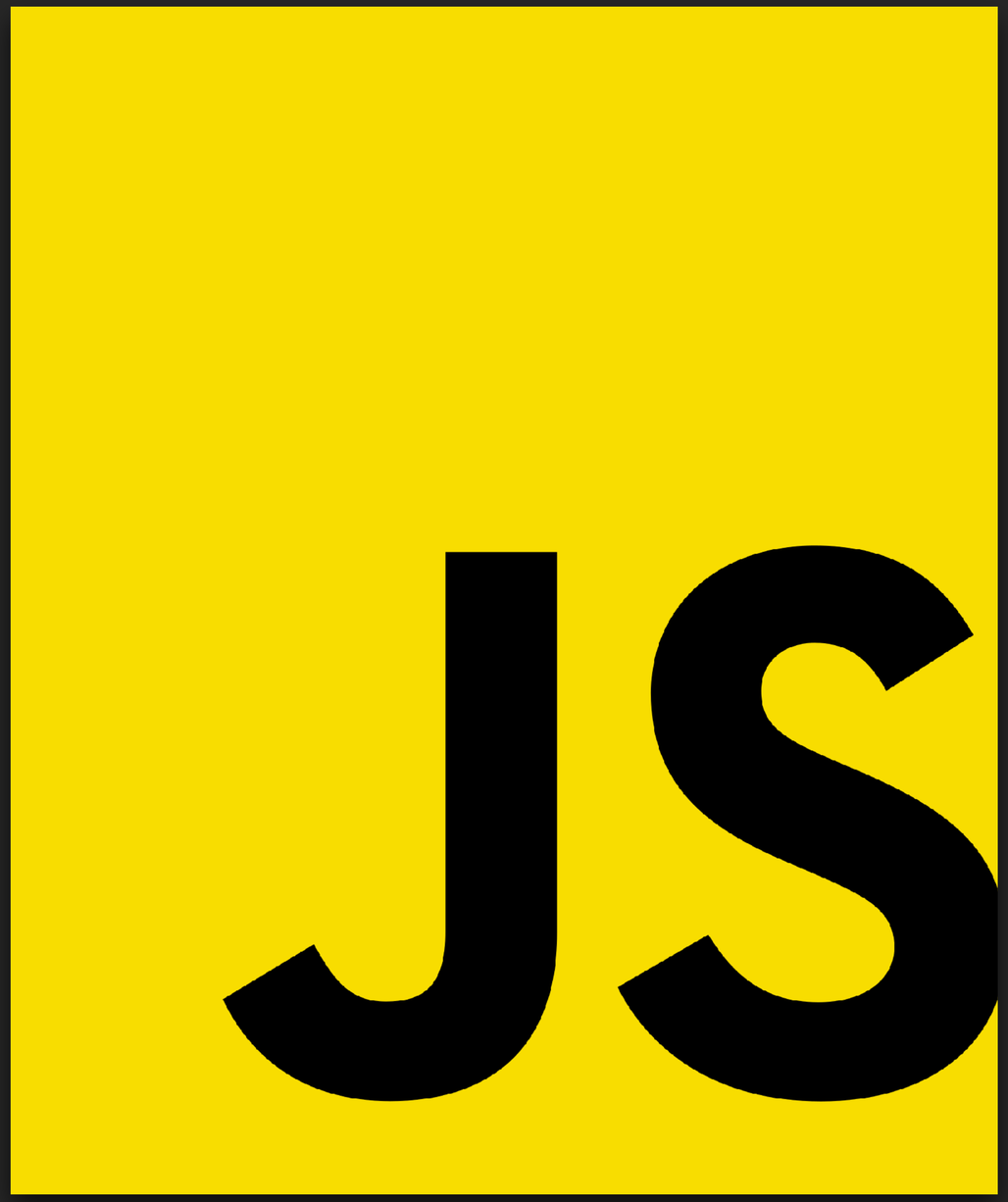
Optimal PWA Checklist

- Provides an offline experience
- Is fully accessible
- Can be discovered through search
- Works with any input type
- Provides context for permission requests
- Follows best practices for healthy code

https://web.dev/articles/pwa-checklist#optimal

Development Tools & Technologies

- **Languages**: HTML, CSS, JavaScript (React, Angular, Vue for SPAs).
- **Frameworks/plugins**: PWA Vite Plugin, Ionic PWA, Remix PWA, PWA Builder
- **Tools**: Lighthouse (for auditing), Workbox (for service workers), PWA Builder.
- **Platforms**: Ex. Firebase for hosting and backend services.



At the end of the day, it is all just JavaScript

🤷🏼

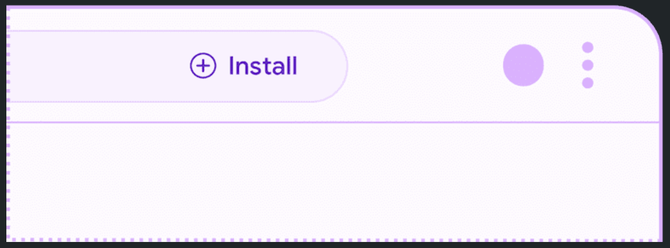
Installable Web Apps

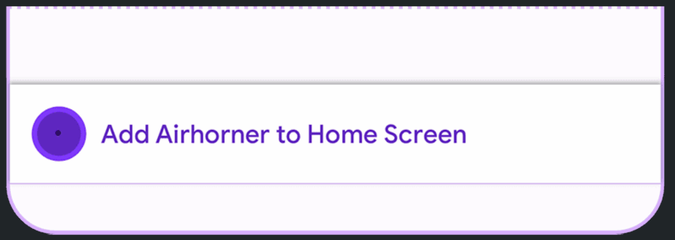
When a Progressive Web App moves out of a tab and into a standalone app window, it transforms how users think about it and interact with it.

https://web.dev/articles/what-are-pwas#installable

Installable Web Apps

- Manifest File
- Logo
- HTTPS
- (Service Worker)





https://web.dev/articles/install-criteria

Technical Foundations of PWAs

- Service Workers: For offline functionality and background tasks.
- Manifest File: For adding the web app to the home screen.
- HTTPS: For secure data transactions.

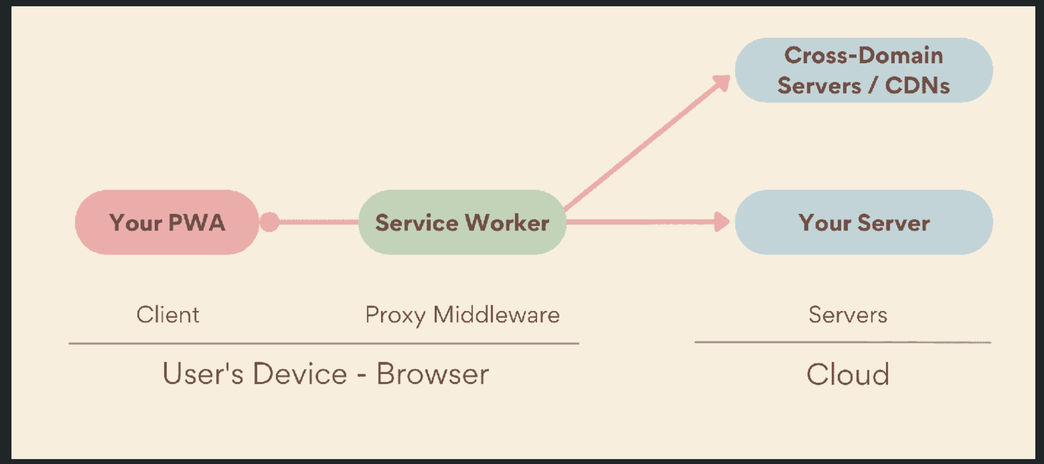
7 Web Features You Didn’t Know Existed

https://web.dev/articles/install-criteria

What is a Service Worker?

Service workers enhance web app performance and user experience.

- A script that runs in the background, separate from a web page.
- Acts as a proxy between the browser, the network, and the web application.
- Enables features like push notifications, background sync, and offline functionality.

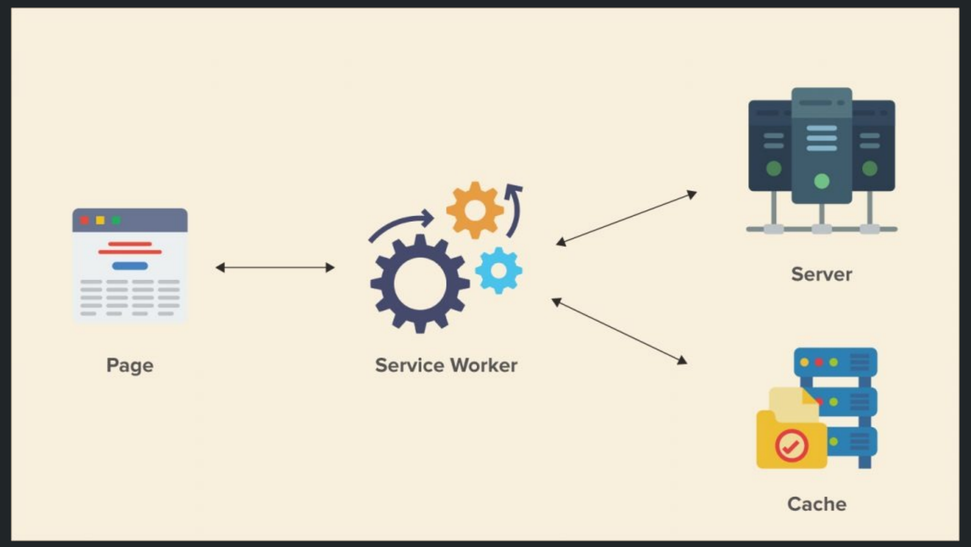


https://web.dev/learn/pwa/service-workers

PWA

Service Worker: Core Functionalities

- Proxy Server: Intercepts network requests to serve custom responses.
- Offline Capability: Caches assets and data for app use without an internet connection.
- Background Updates: Updates cached content in the background.
- Push Notifications: Sends notifications to re-engage users.

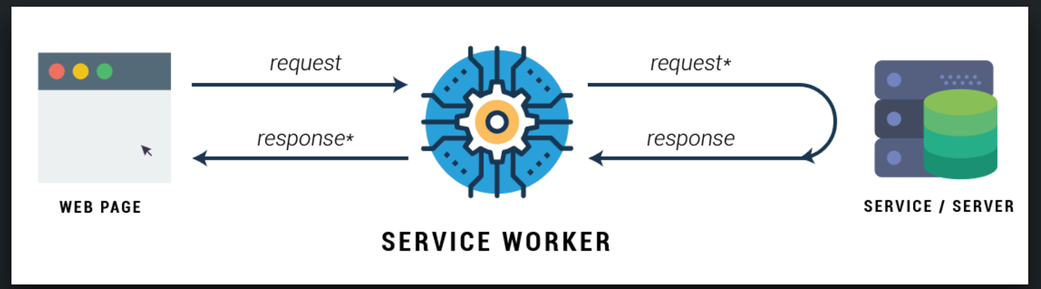


https://web.dev/learn/pwa/service-workers

PWA

Service Worker

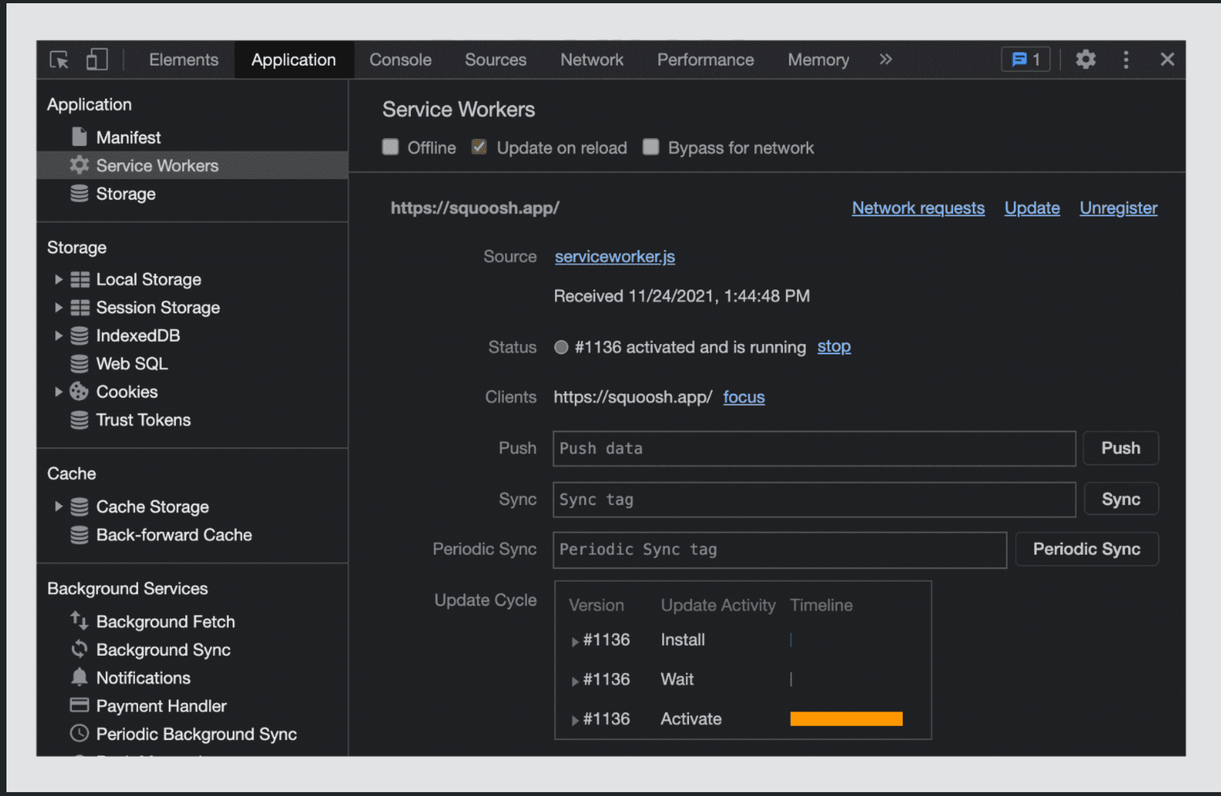
- Key Features
  - Event-Driven: Operates based on events like fetch, install, and activate.
  - Life Cycle: Independent from the web page, includes installation, activation, and updates.
  - Async Operations: Utilizes promises for asynchronous actions.
- Requirements & Limitations
  - HTTPS Required: Needs a secure context to operate, except on localhost for development.
  - Compatibility: Not all browsers support service workers fully.
  - Complexity: Requires careful design to manage caching and updates effectively.



https://web.dev/learn/pwa/service-workers

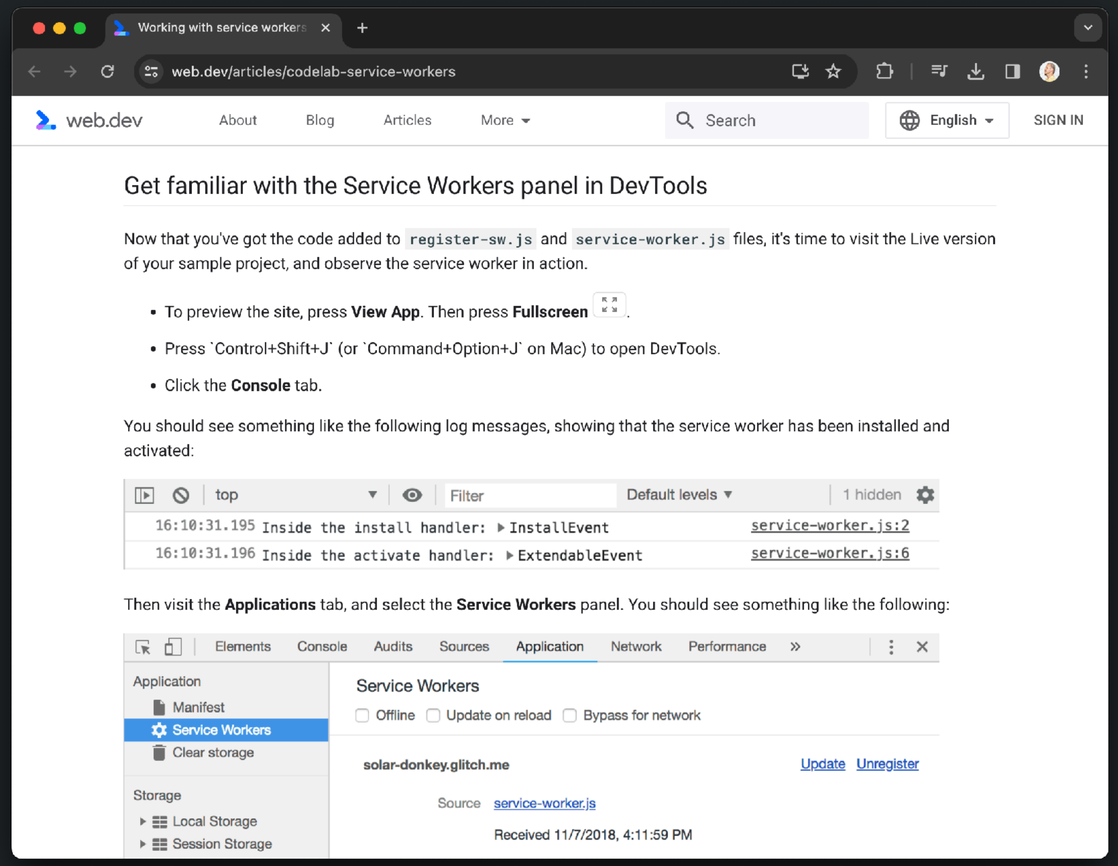
Service Worker: Tools

- Lighthouse: For auditing and performance metrics.
- Workbox: A set of libraries to simplify common service worker tasks and caching.
- Service Workers menu



https://web.dev/learn/pwa/service-workers

Working with Service Workers



https://web.dev/articles/codelab-service-workers

Benefits of PWAs

- Improved Performance: Faster load times and smoother interactions.
- Increased Engagement: Features like push notifications.
- Cost-Effective: Development and maintenance costs lower than native apps.

https://web.dev/articles/install-criteria

Benefits of PWAs

- It's just a website! You don't need to build separate apps anymore. If you have a website, you can easily turn it into and iOS and Android app as well!
- A PWA is much smaller than a native app. Your users no longer need to install tens of megabytes of code
- No need to get your app into the App Store or Play Store. Just share the link to your website and users can install it as an app
- There's no need to get users to install updates anymore. When you release a new version of your app, all your users automatically get the new version
- By default, PWAs are served over HTTPS and are therefore safe and secure
- PWAs are lightweight and offer high performance
- Especially on Android, a PWA can almost do anything a native app can

https://whatpwacando.today/info

Challenges & Considerations

- Browser Compatibility: Not all PWA features are supported across all browsers.
- Performance Optimization: Requires careful consideration, especially for complex applications.

Real-world Examples of PWAs

- Twitter
- Youtube
- Pinterest
- Uber
- Starbucks
- Canvas