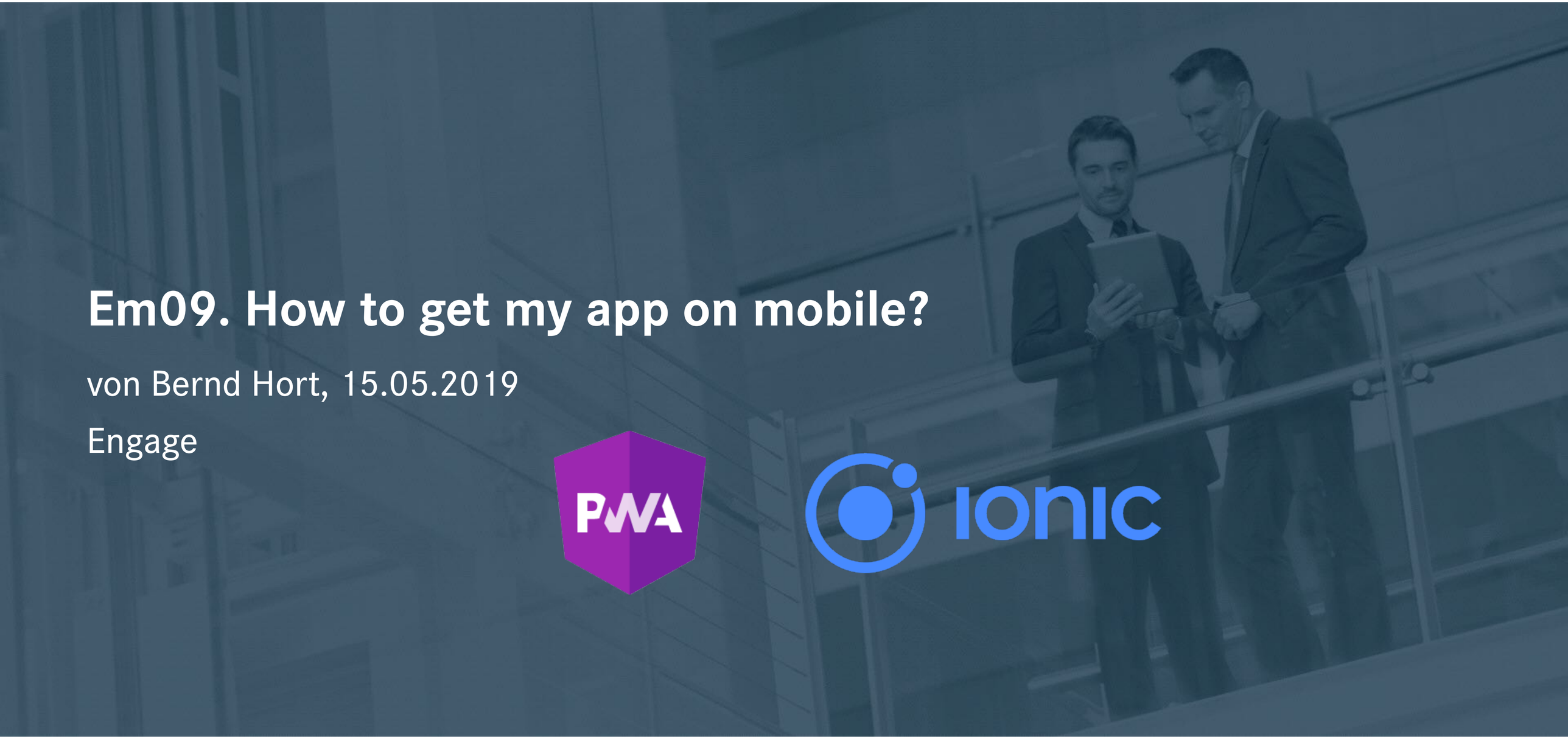
assono

www.assono.de



Em09. How to get my app on mobile?

von Bernd Hort, 15.05.2019

Engage

PWA

IONIC

Bernd Hort

- Master degree in computer science
- Lotus Notes developer since 3.3
- „It’s all about giving the user the right tool to get the job done.“

I have this cool idea for an app!

How do I get it on mobile?

Classic Webapp

PWA

IONIC

Android Studio

Complexity

assono

Progressive Web Apps

Progressive Web Apps - PWA

- Progressive Web Apps enhance web apps to make them behave (a little bit more) like native apps
  - Offline
  - Start from home screen
  - Access the camera
  - Local storage

assono

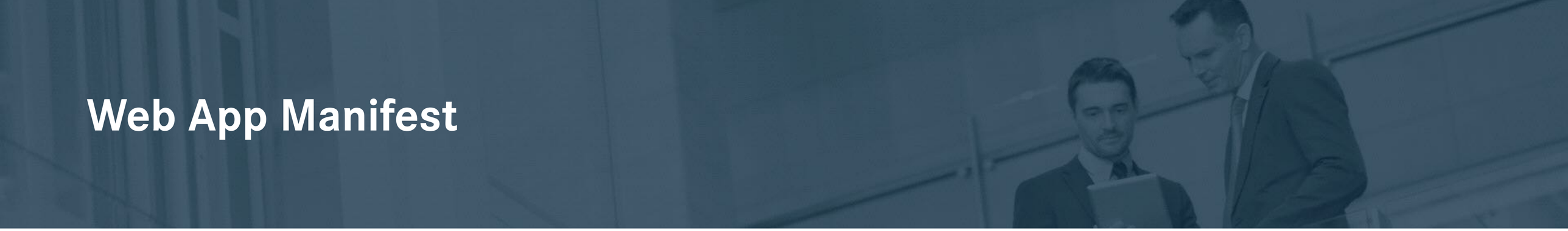
PWA Principles

- **Discoverable -** so the contents can be found through search engines.
- **Installable -** so it's available on the device's home screen.
- **Linkable -** so you can share it by simply sending a URL.
- **Network independent -** so it works offline or with a poor network connection.
- **Progressive -** so it's still usable on a basic level on older browsers, but fully-functional on the latest ones.
- **Re-engageable -** so it's able to send notifications whenever there's new content available.
- **Responsive -** so it's usable on any device with a screen and a browser – mobile phones, tablets, laptops, TVs, fridges, etc.
- **Safe -** so the connection between you and the app is secured against any third parties trying to get access to your sensitive data.

Source: Mozilla Developer Network: Introduction to progressive web apps

assono

Web App Manifest



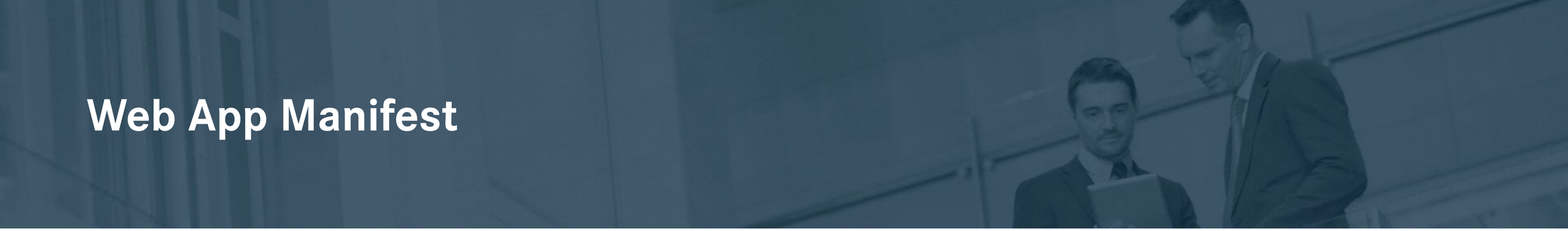
- The „Web App Manifest“ defines the main settings and appearance
- It is a JSON file linked via

```
<link rel="manifest" href="/manifest.json">
```

- Definition from Google Developers - The Web App Manifest

assono

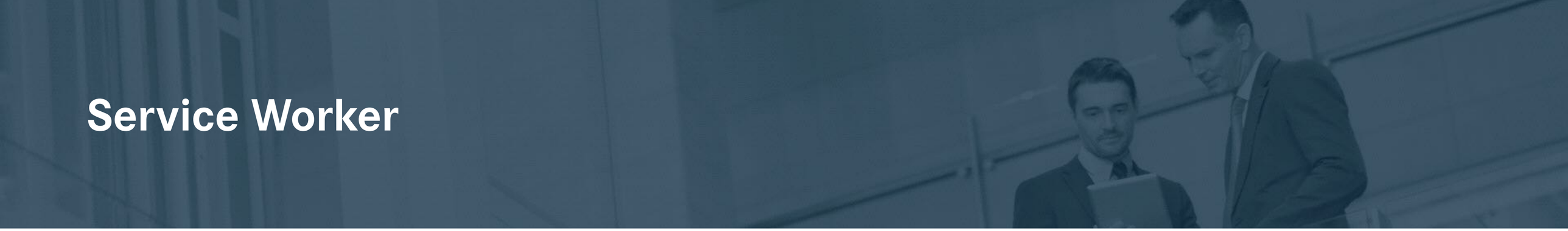
Web App Manifest



```
{
  "short_name": "Maps",
  "name": "Google Maps",
  "icons": [
    {
      "src": "/images/icons-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/images/icons-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": "/maps/?source=pwa",
  "scope": "/maps/",
  "background_color": "#3367D6",
  "display": "standalone",
  "theme_color": "#3367D6"
}
```

assono

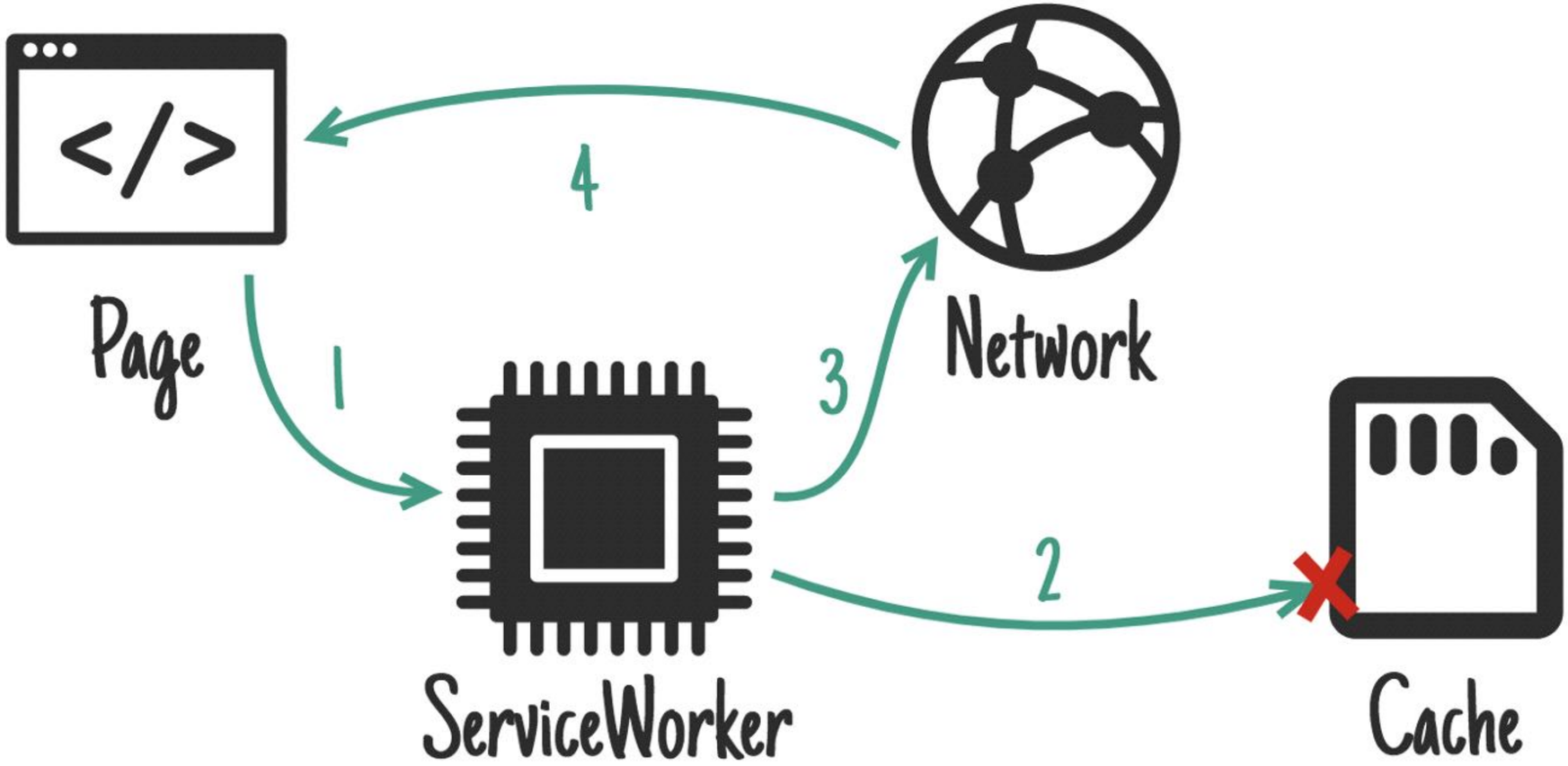
Service Worker



- The Service Worker is a JavaScript file, which will be executed in a separate thread per PWA in the background.
- No direct access to frontend elements
- Implements hooks to react on a number of events

assono

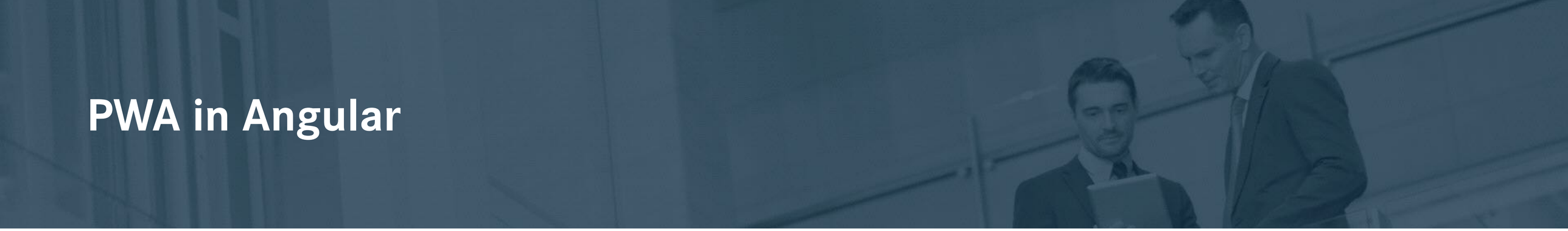
Service Worker



Quelle: https://christianlydemann.com/how-to-cache-http-requests-in-an-angular-pwa/

assono

PWA in Angular



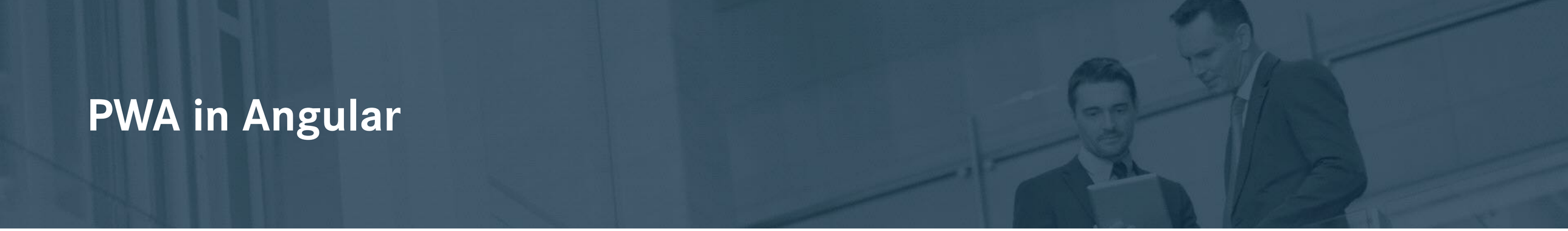
- Since version 5.0 Angular supports Progressive Web Apps directly.
- For existing Angular project the package @angular/pwa needs to be added

```
$ng add @angular/pwa --project <projectname>
```

- Creates manifest.json and ngsw-config.json files
- Provides icons in sizes the platform needs

assono

PWA in Angular



- Build Angular app for production

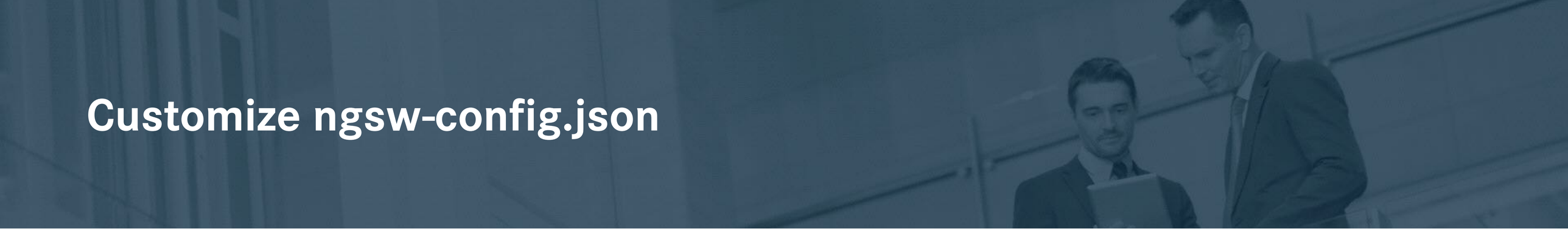
```
$ng build --prod
```

- The default development server is not sufficient for testing pwa
- Use the http-server package which comes with node.js
- Run via

```
$http-server -p 8080 -c-1 dist/<project-name>
```

assono

Customize ngsw-config.json



- Per default all app assets will be cached
  - entry „assetGroups" in ngsw-config.json
- Handling for REST-API data requests must be configured manually
  - entry „dataGroups“
  - definition of URL patterns
  - „api-performance“ vs „api-freshness“
  - configure individual cache settings

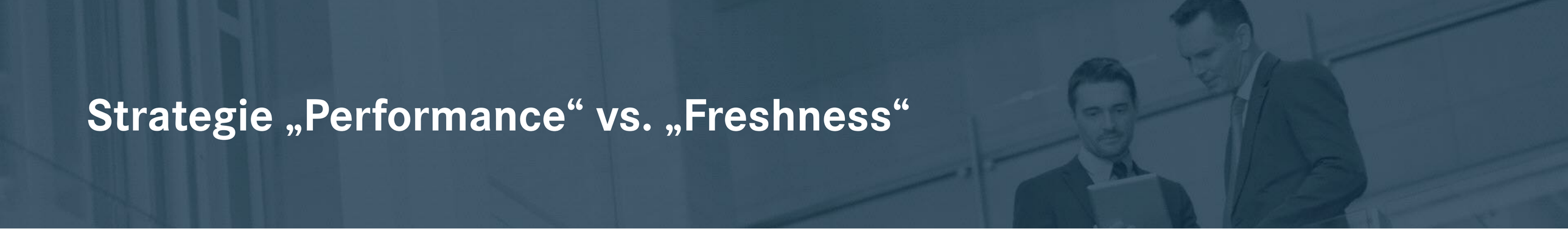
assono

„Performance“ vs. „Freshness“

- With the cache strategy „Performance“ data will be fetched from **cache first** until the expiration date will be reached.
- With the cache strategy „Freshness“ the first attempt is to get **fresh data** from the the **server**.
  Only if the timeout is reached the data will be get from the cache.

assono

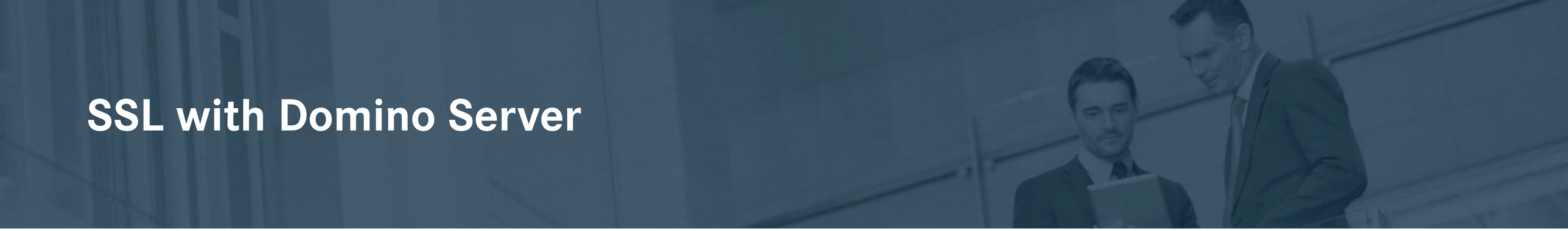
Strategie „Performance“ vs. „Freshness“



```
{"dataGroups": [{
    "name": "api-performance",
    "urls": [
      "/assets/i18n/**",
      "/api/**"
    ],
    "cacheConfig": {
      "strategy": "performance",
      "maxSize": 100,
      "maxAge": "3d"
    }
  },
  {
    "name": "api-freshness",
    "urls": [
      "/api/fresh-todo-list"
    ],
    "cacheConfig": {
      "strategy": "freshness",
      "maxSize": 100,
      "maxAge": "3d",
      "timeout": "10s"
    }
  }
]
}
```

assono

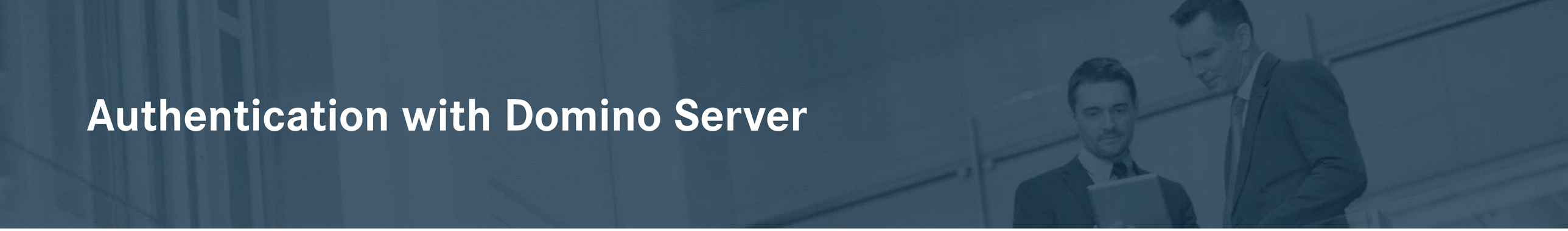
SSL with Domino Server



- Without SSL connection the Service Worker will not operate
- For public Domino Servers try
  midpoints Let's Encrypt for Domino (LE4D) (it’s free)
- For private Domino Servers use self-signed certificate
  - Good instructions: Setting up https with a valid certificate for a local domain – and use it with Node.js
  - Import Certificate Authority (CA) in your test browser

assono

Authentication with Domino Server



- Before PWA I used to check for Domino specific Cookies for first check if a user is authenticated
- With PWA the session based cookie will not disappear if the browser is closed
- Since IBM Domino 9.0.1 FP10 release new notes.ini setting DOMINO_FORCE401_WITH_HTML_LOGIN_PAGE=1 to force a 401 HTTP return code

assono

Chrome Developer Tools

- The best support for developing PWA is provided by the Chrome Developer Tools
- Initialize an audit to check PWA status

assono

Hybride Apps

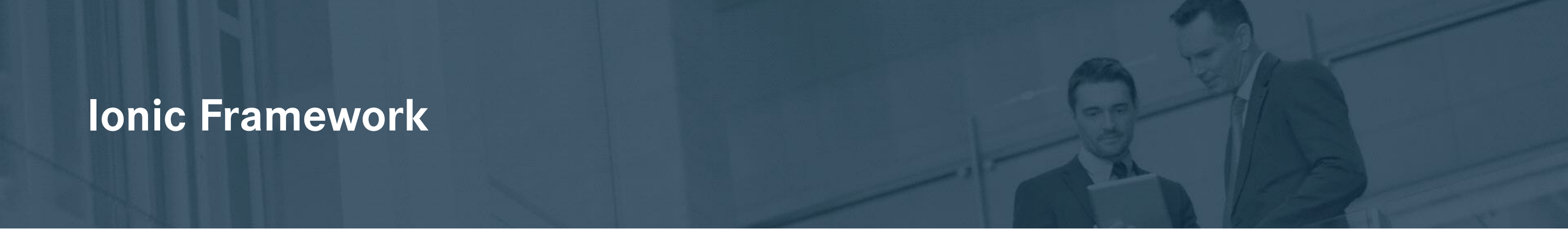
IONIC

Hybride Apps

- The programming model for hybrid apps are HTML and JavaScript.
- Once the app is ready a supporting framework will generate code for native iOS or Android apps.
- Using Xcode or Android Studio to compile the generated code to a native app.
- Apache Cordova is the base for most of the existing frameworks.

assono

Ionic Framework

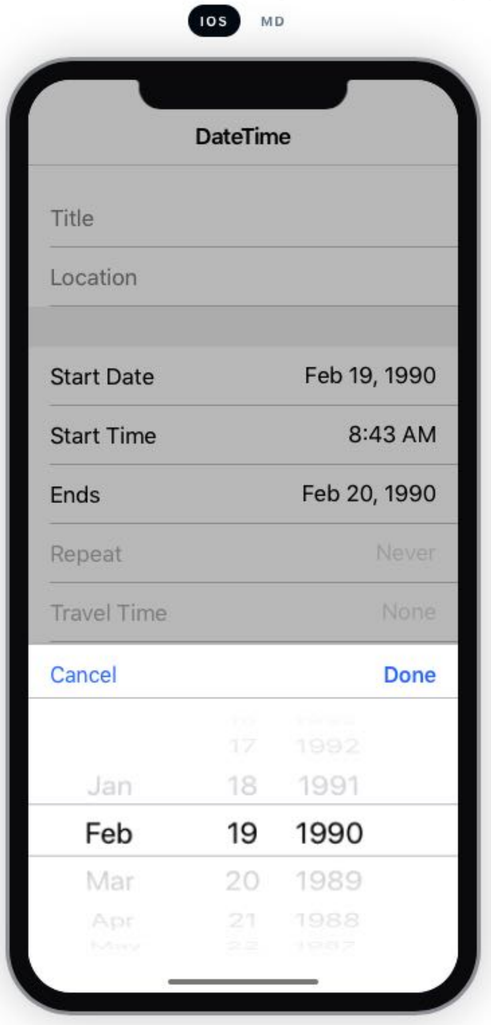


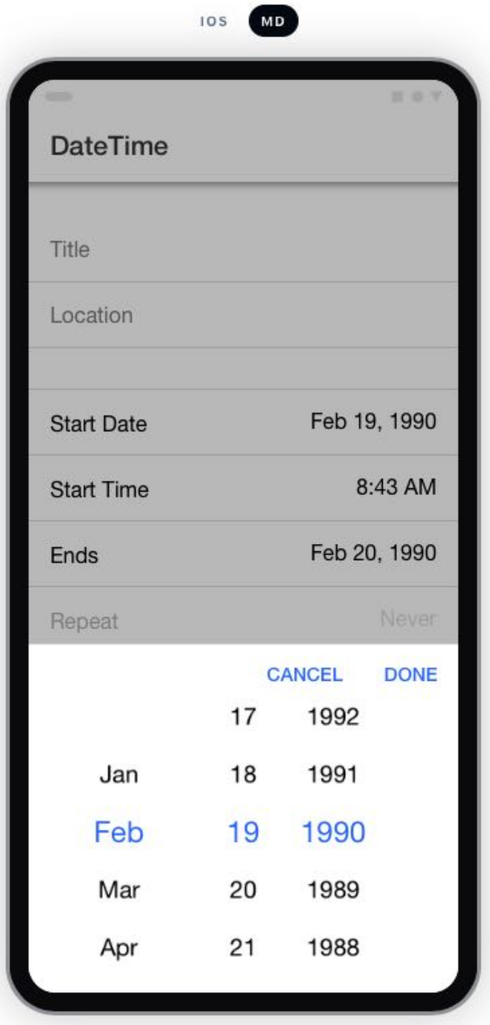
- Ionic uses also Apache Cordova
- Originally Ionic only supports Angular (AngularJS)
- Since version 4.0 also React or Vue.js are supported.
- First approaches to move away from Cordova
  - Ionic own project Capacitor
- Use payed services for build support
  - Creating iOS apps without a Mac

assono

Ionic Look & Feel

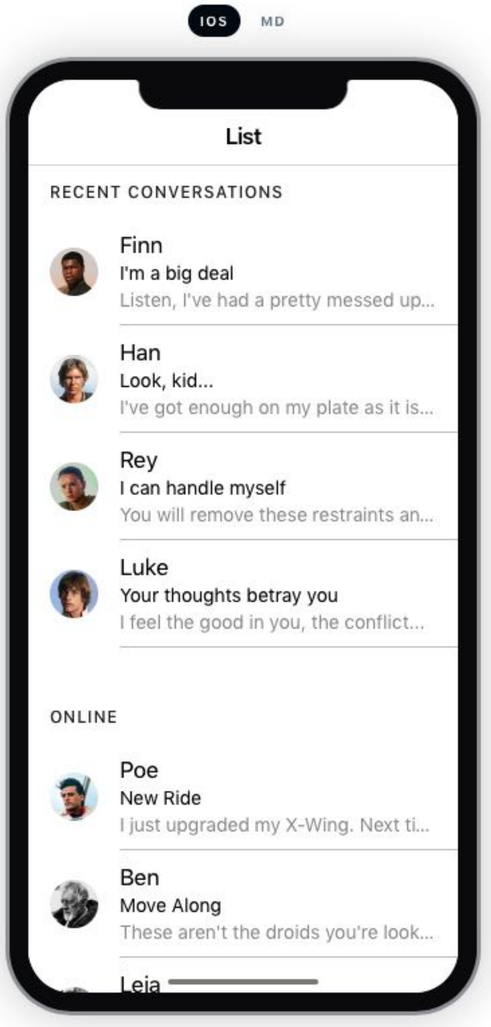
- Ionic uses native Look & Feel depending on platform

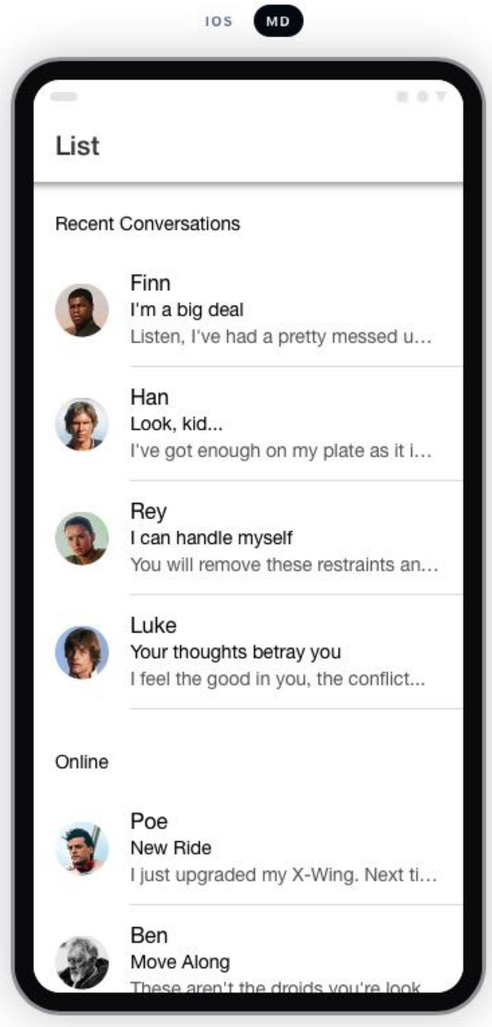




Ionic Look & Feel

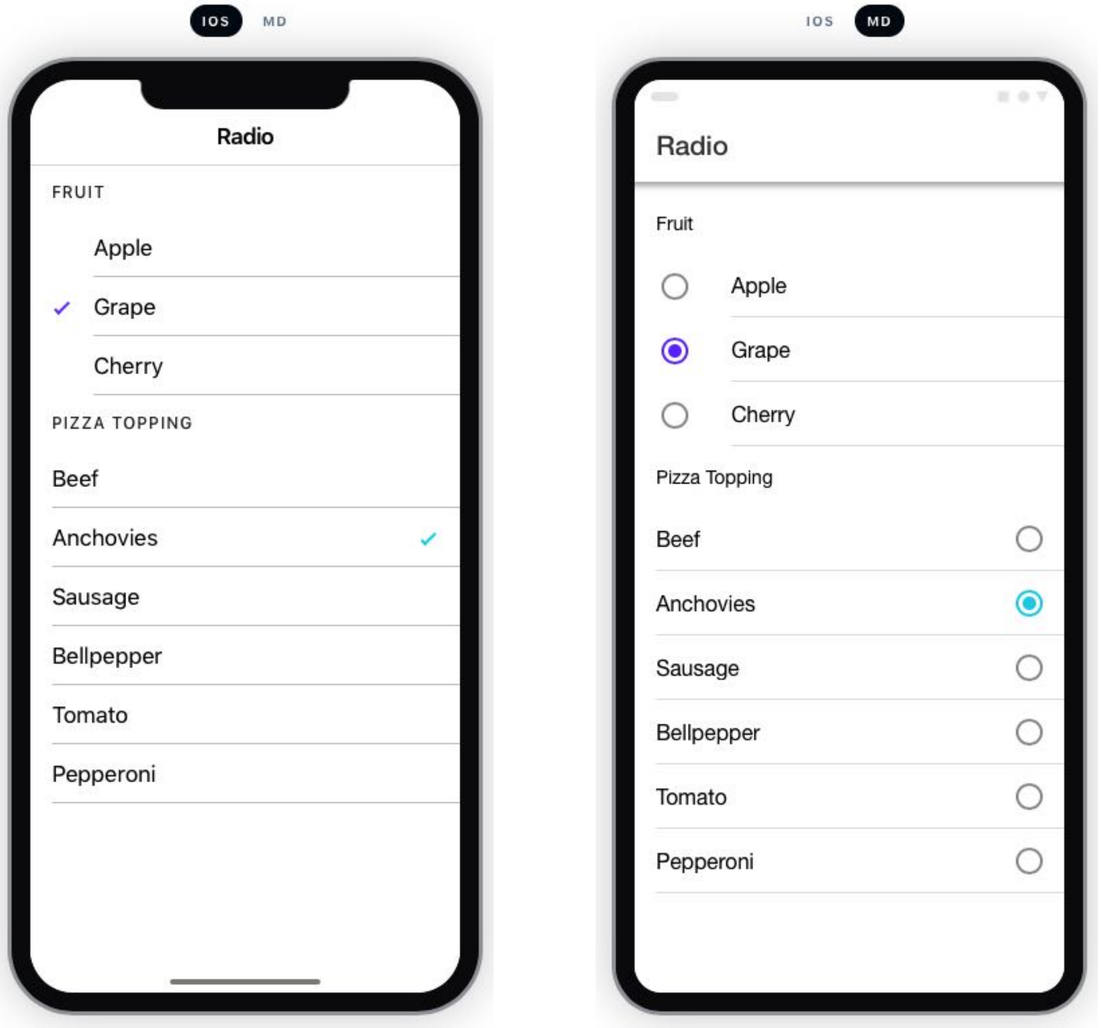
- Appearance is similar but respects the UX guidelines of the platform





Ionic Look & Feel

- During build the corresponding UI element is chosen



ionic

assono

Ionic Look & Feel

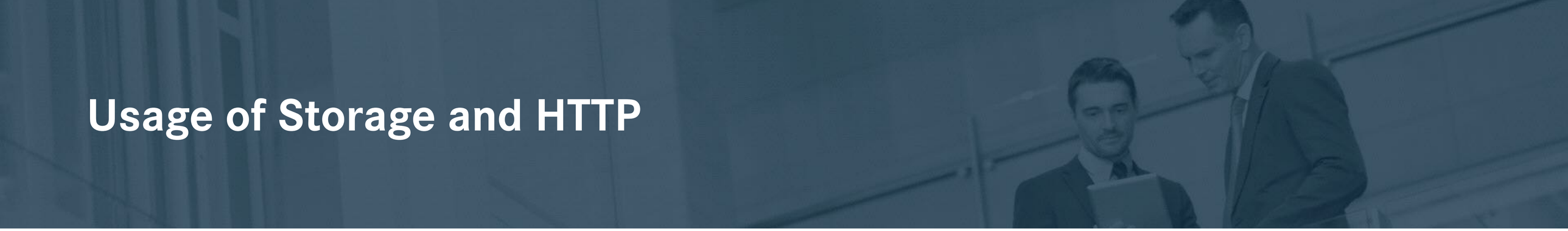
```
<ion-list>
        <ion-radio-group>
                <ion-list-header>
                        <ion-label>Name</ion-label>
                </ion-list-header>

                <ion-item>
                        <ion-label>Biff</ion-label>
                        <ion-radio slot="start" value="biff" checked></ion-radio>
                </ion-item>

                <ion-item>
                        <ion-label>Griff</ion-label>
                        <ion-radio slot="start" value="griff"></ion-radio>
                </ion-item>

                <ion-item>
                        <ion-label>Buford</ion-label>
                        <ion-radio slot="start" value="buford"></ion-radio>
                </ion-item>
        </ion-radio-group>
</ion-list>
```

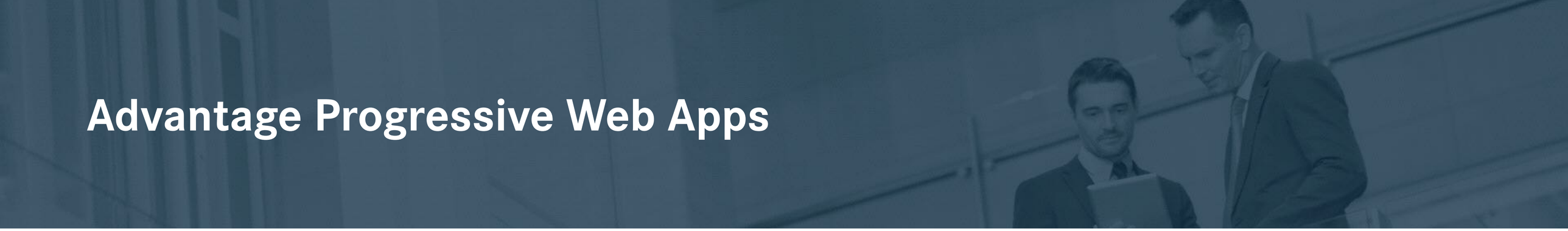
Usage of Storage and HTTP



- Separate packages for dealing with storage and HTTP
- The developer is responsible for providing a strategy how to combine the Storage package and the HTTP package:
  - Store an additional timestamp while storing the data into the cache
  - Decide on the expiration time for each type of resource
  - Decide whether to test first the cache or getting „fresh“ data

Comparison

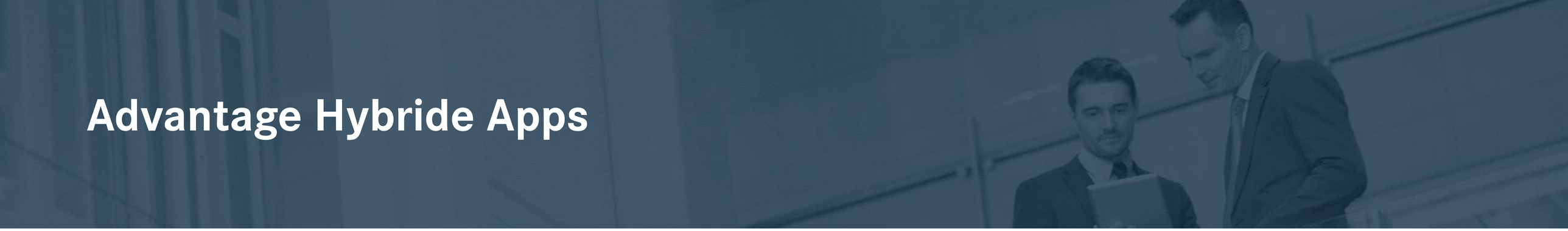
Advantage Progressive Web Apps



- Faster development since testing is done in the browser
- No need to deploy the app on test devices
- No approval needed for Apple iOS App Store or Android Play Store
- No Apple developer license
- No company app store
- No Mac needed

assono

Advantage Hybride Apps



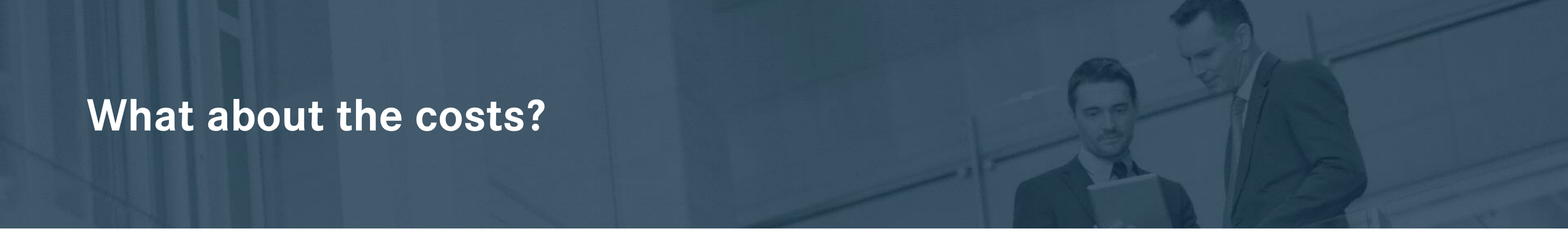
- Access to native resources
  - Barcode scanner
  - Live preview
  - Push notifications in iOS
- Nativ Look & Feel
- Higher prestige!

assono

Let’s talk about money

Photo by Ramiro Mendes on Unsplash

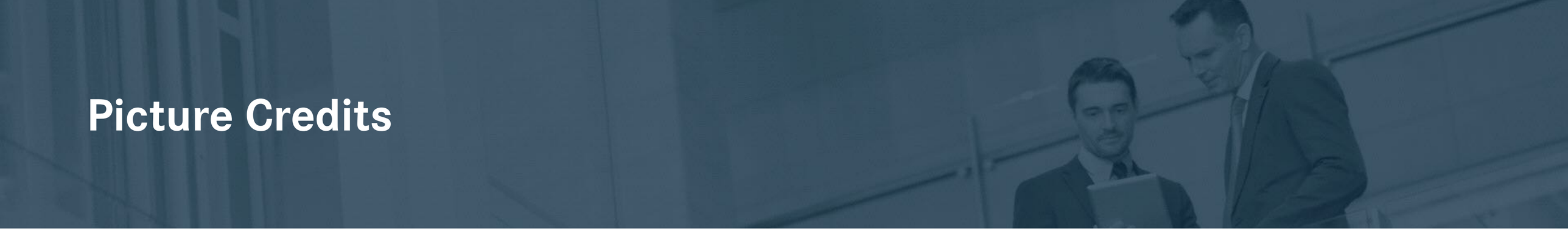
What about the costs?



- Both approaches come free to start with
  - PWA just depends on browser support
  - Angular and Ionic both uses MIT license
- Ionic offers paid support
  - It is possible to build production apps without support
- **Main factor is development time / costs!**

assono

Questions?



Picture Credits

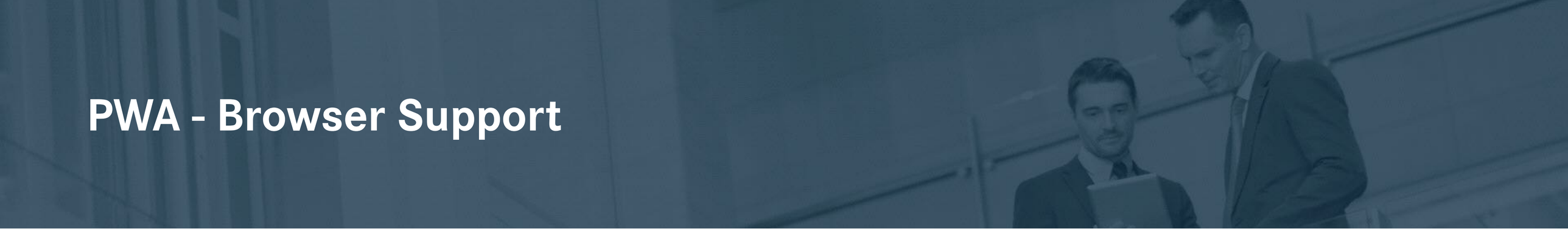
- PWA-Logo
  https://github.com/webmaxru/progressive-web-apps-logo
- Ionic-Logo
  https://de.wikipedia.org/wiki/Datei:Ionic_Logo.svg
- Unplash
  https://unsplash.com/
- Pexels
  https://www.pexels.com/

assono

Slide backup

Photo by jesse orrico on Unsplash

PWA - Browser Support

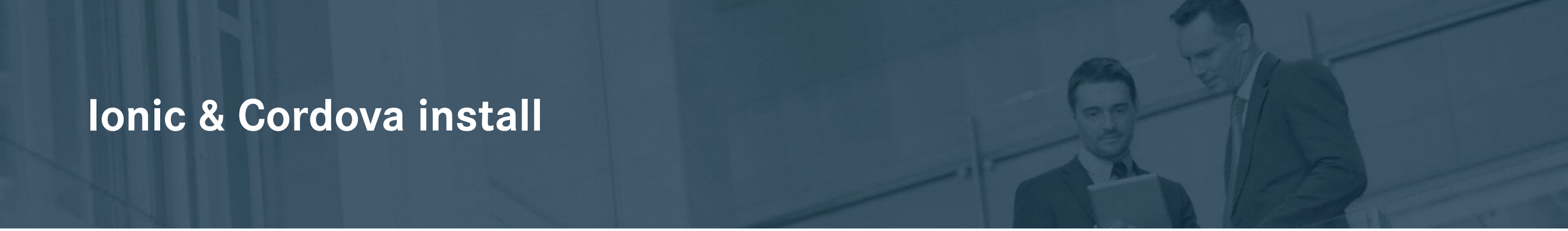


- Web App Manifest
  - No support for IE, Edge (non Chrome), Firefox, Safari (Desktop)
- Service Worker
  - Service Worker itself - Most browsers
  - Background Sync API - Only Chromium based browsers
  - Fetch - Most browsers

Source: Can I use...?

assono

Ionic & Cordova install



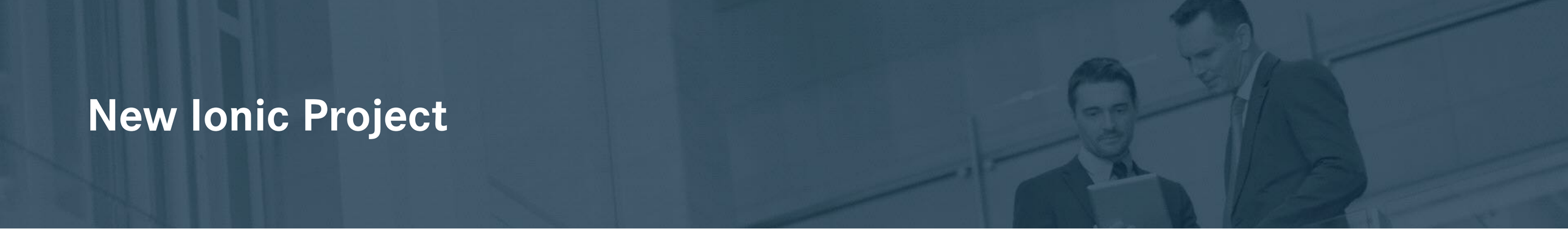
- Ionic and Cordova both use the npm Package manager
- Install with

```
$npm install -g ionic
```

and

```
$npm install -g cordova
```

assono

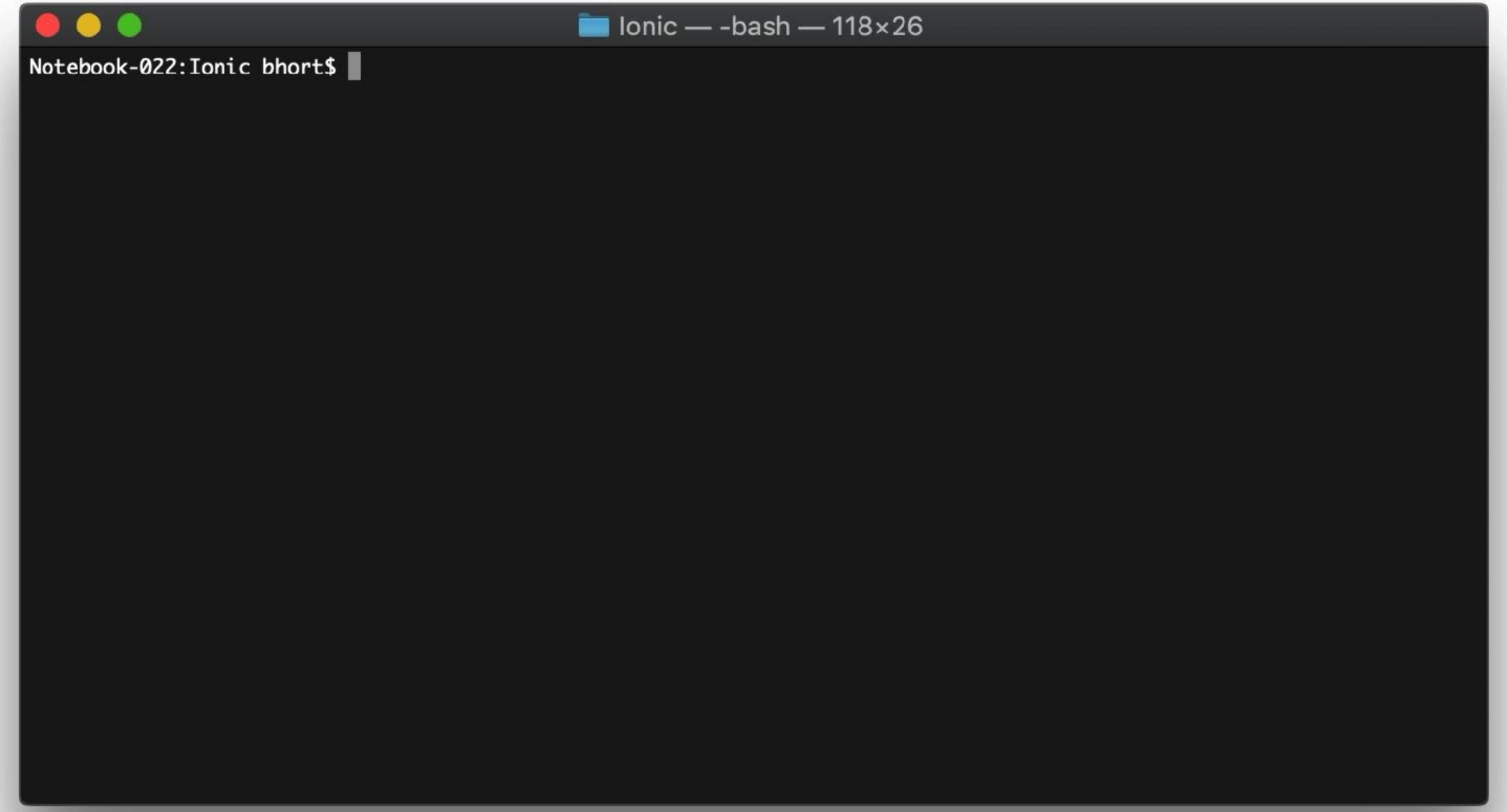


New Ionic Project

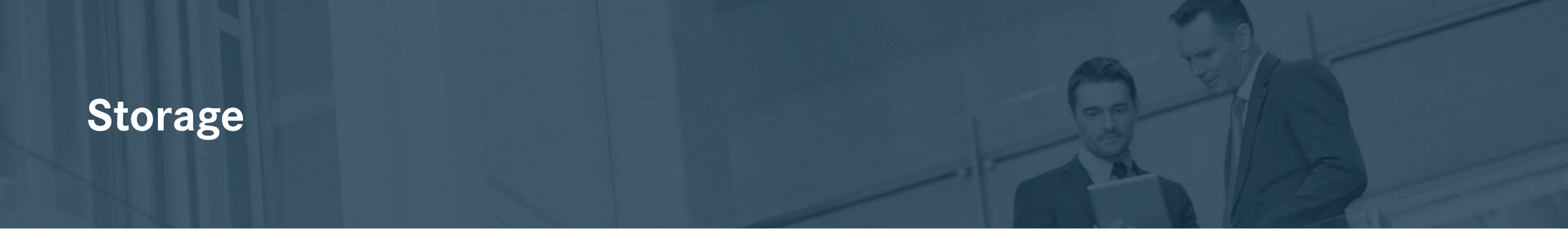
- Create a new project with

```
$ionic start {Projektname}
```

- Use additional parameter to define the typ of application and some basic template


Ionic — -bash — 118×26
Notebook-022:Ionic bhort$

Storage



- The Storage package encapsulates different storage options
- Install with

```
$npm install @ionic/storage
```

- Install the Cordova plugin to use SQLLite

```
$ionic cordova plugin add cordova-sqlite-storage
```

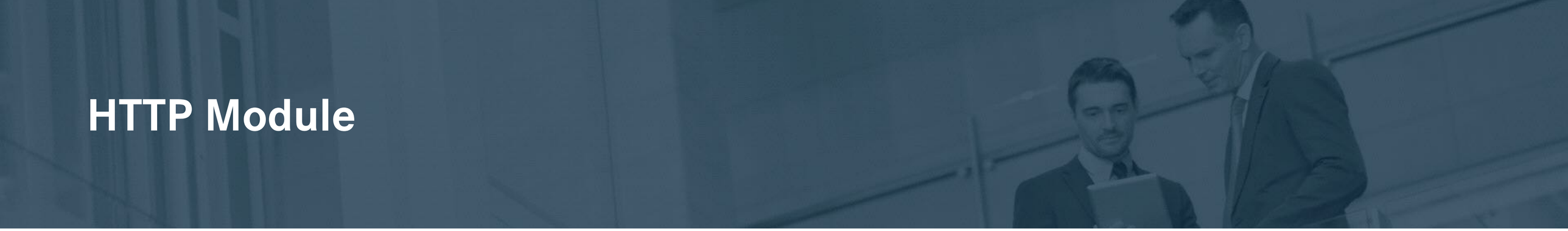
Dependency Injection of the Storage Module

- As described on the Ionic web page
- Modify src/app/app.module.ts
  - Add the needed import

```
import { IonicStorageModule } from '@ionic/storage';
```

```
@NgModule({
  …
  imports: […, IonicStorageModule.forRoot()],
  …
})
```

assono



HTTP Module

- The HTTP package encapsulates the network usage
- Instal with

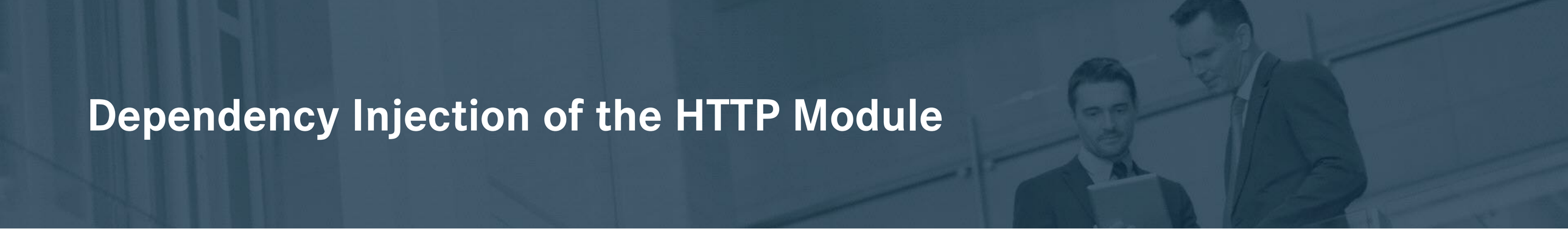
```
$npm install @ionic-native/http
```

- Install the corresponding Cordova Plugin

```
$ionic cordova plugin add cordova-plugin-advanced-http
```

assono

Dependency Injection of the HTTP Module



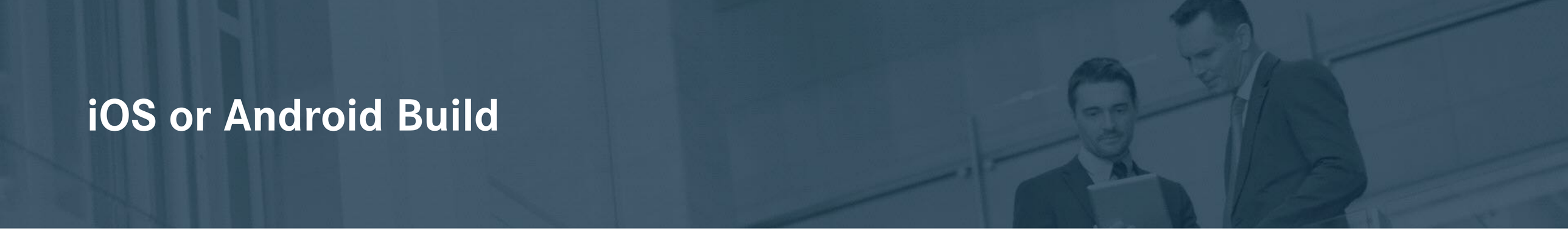
- As described on the Ionic web page
- Datei src/app/app.module.ts anpassen
  - Import hinzufügen

```
import { HTTP } from '@ionic-native/http/ngx';
```

```
@NgModule({
  …
  imports: […, HTTP],
  …
})
```

assono

iOS or Android Build



- iOS Build

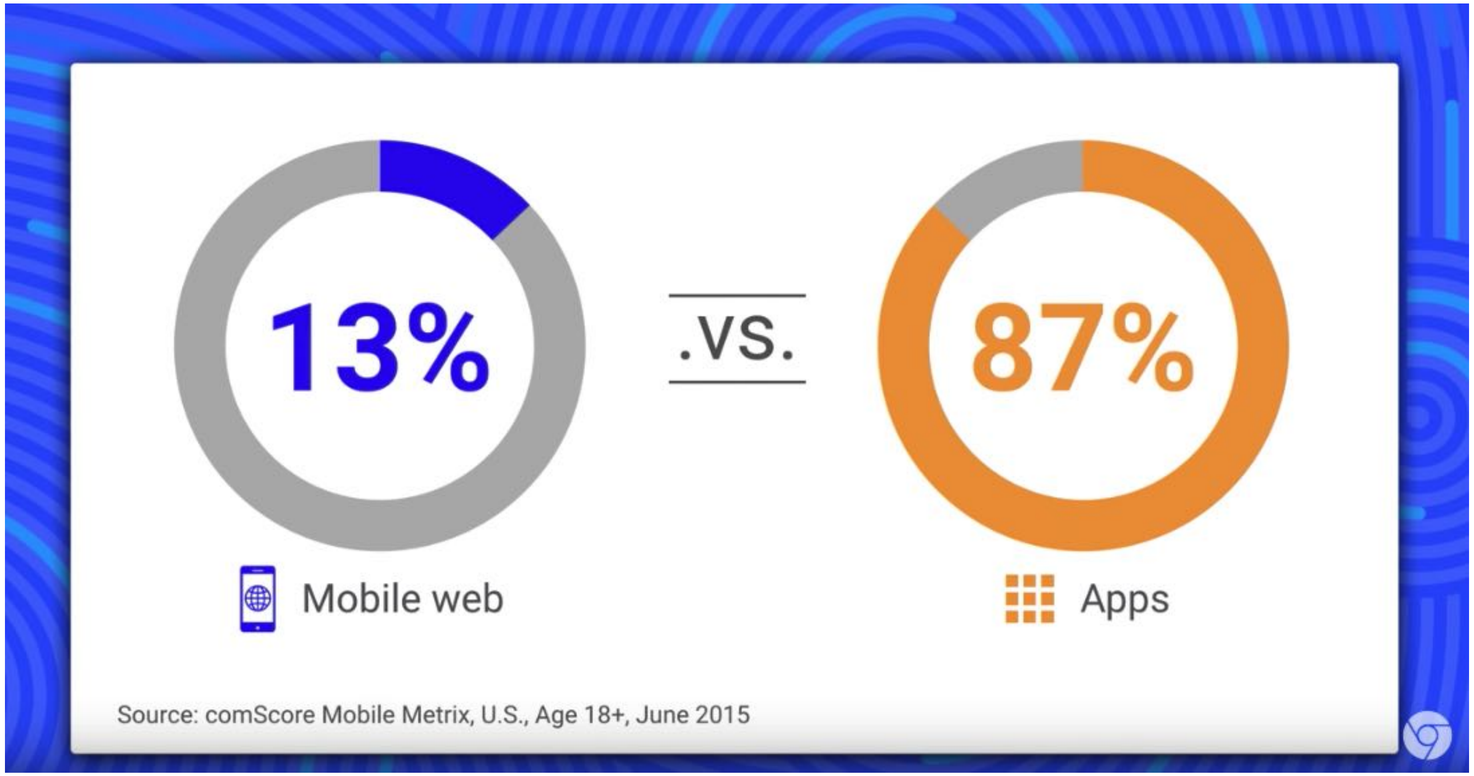
```
$ionic cordova prepare ios
```

  - open afterwards in Xcode Project
- Android Build

```
$ionic cordova prepare android
```

  - open afterwards in Android Studio

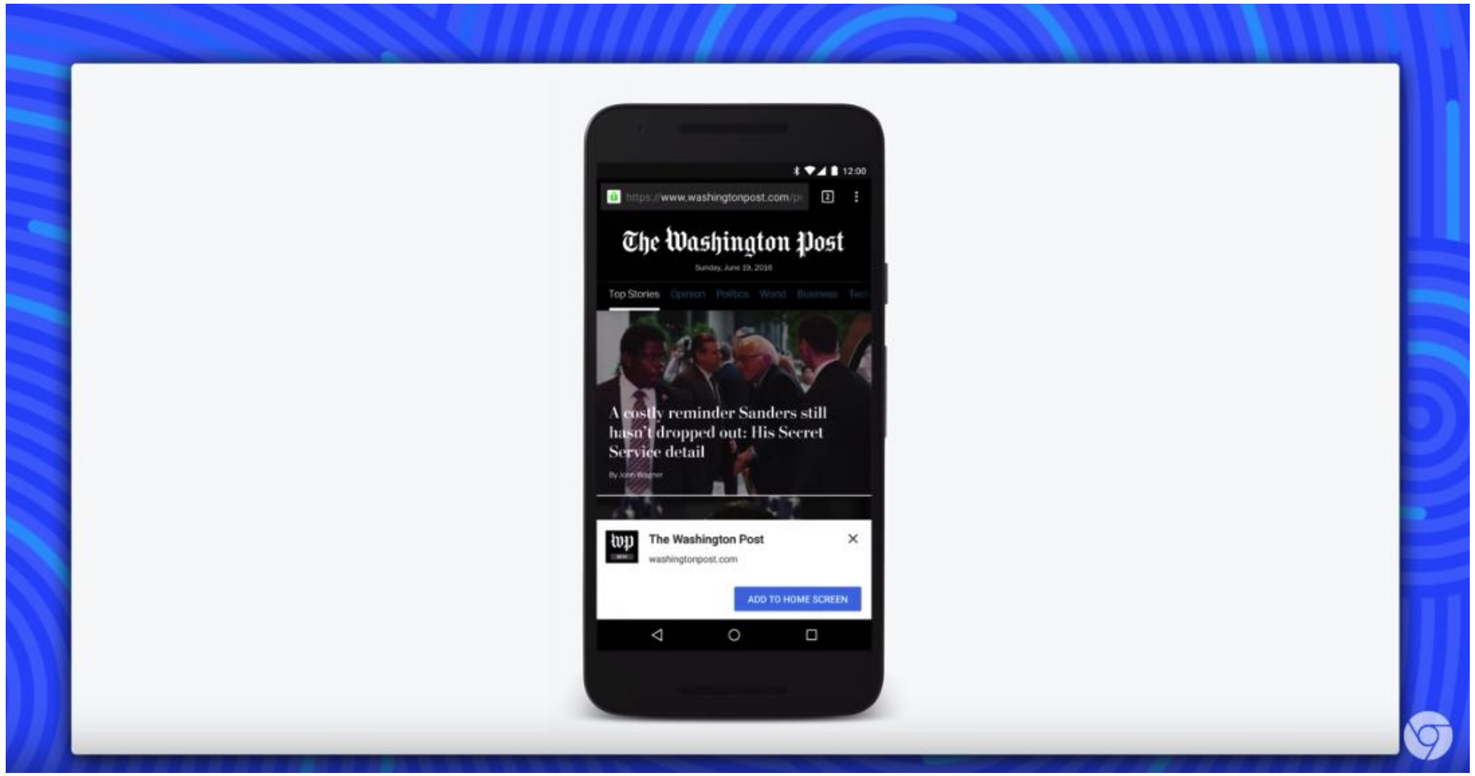
Mobile Web vs. Apps



assono

„Add-to-Home-Screen“

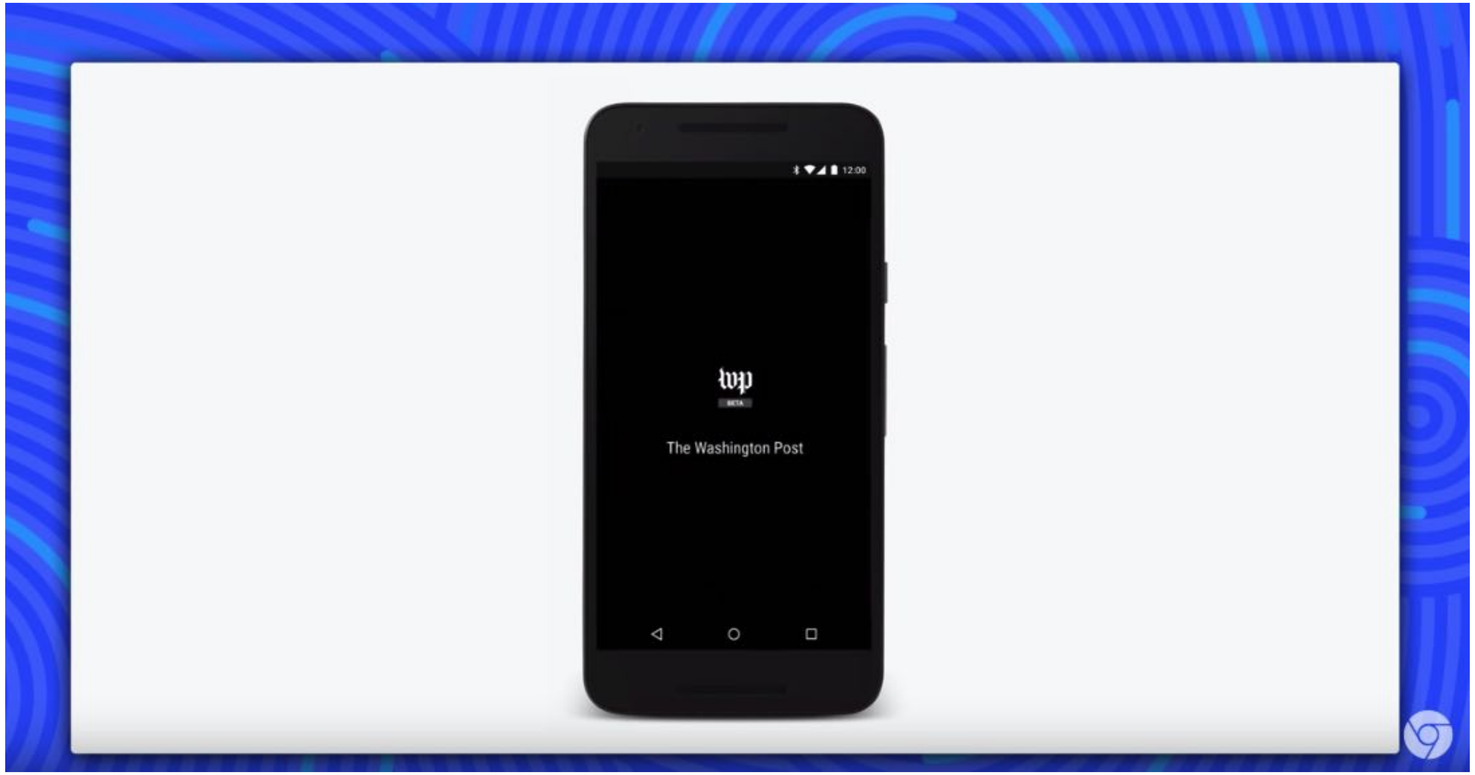
- Dialog will displayed automatically after frequent visit on the web site
- The browser decides!
- The developer can not push it!



assono

„Splash-Screen“

- Will be shown while the web app will be loaded.



assono