

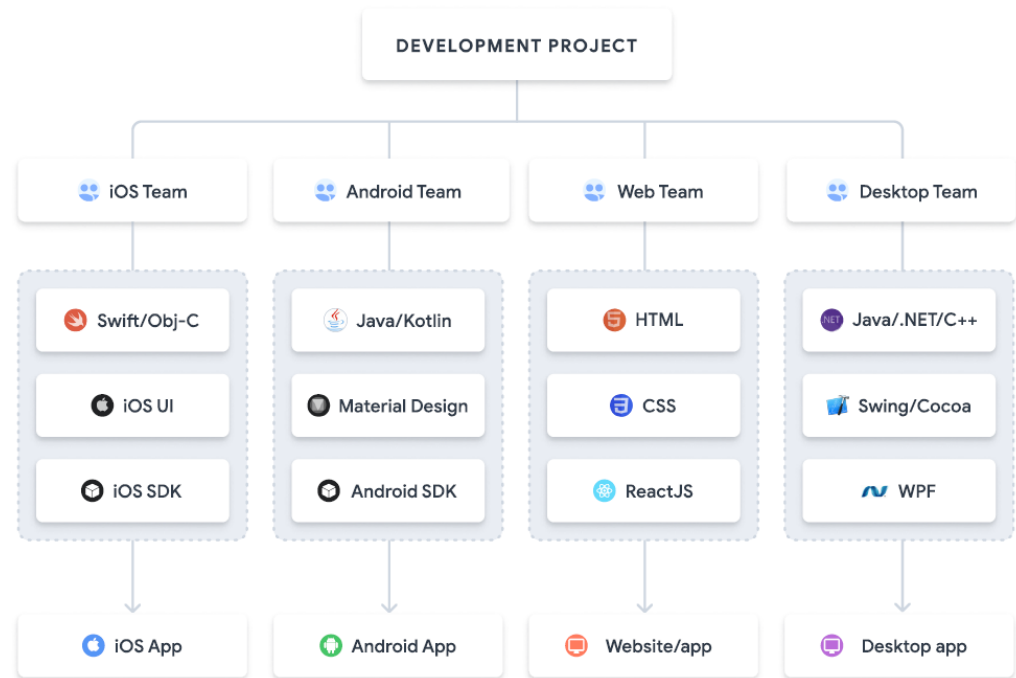
# **Initial Experiences with Progressive Web Apps**

R. Bacher, MCS

22.7.2020

# Native App

- Single – Platform App
- No common code base
- Platform - specific SDK
- App deployment / installation through platform – specific App Store (e.g. Google Play Store)



# Hybrid App

- Native, multi-platform App
- Common code base
- Web technologies
- Platform – specific WebView component / native device wrapper (e.g. Cordova)
- App deployment / installation through platform – specific App Store (e.g. Google Play Store)

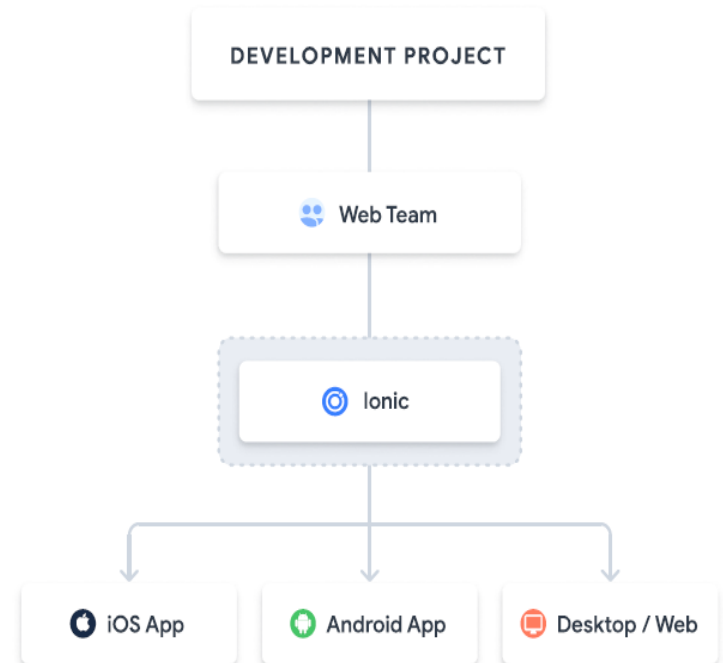


# Progressive Web App

- Multi-platform Web App
- Common code base
- Most recent Web technologies (i.e. pure HTML5, no platform – / browser – specific dependencies),
- Apps runs in browser engine
- App deployment / installation through enterprise web server / browser
- App look and feels like a native app
- App includes offline storage and access to native features
- App works with all form factors (large / medium / small / portrait / landscape)
- App loads fast / works offline
- App is easily discovered through a simple web search and can be installed on the user's home screen

# Ionic Framework

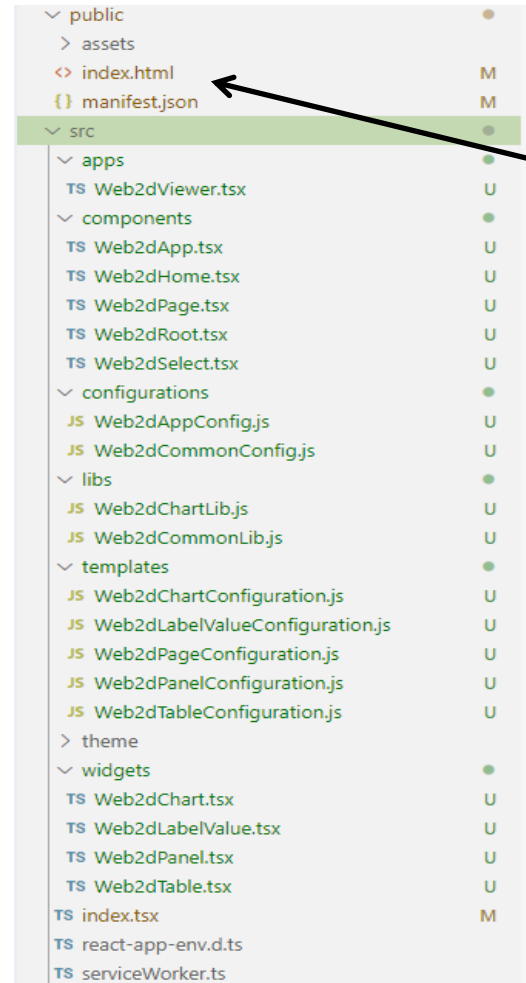
- Free to use and open source
- Platform – agnostic (by using cross-platform app runtime [Capacitor](#))
- SDK – agnostic, including:
  - [Angular](#)
  - [React](#)
  - [Vue.js](#)
  - No SDK at all, just using plain Javascript
- Common code base
- Provides GUI toolkit
- Provides ready-to-use build workflows, including:
  - Native / Hybrid Apps
  - Progressive Web Apps



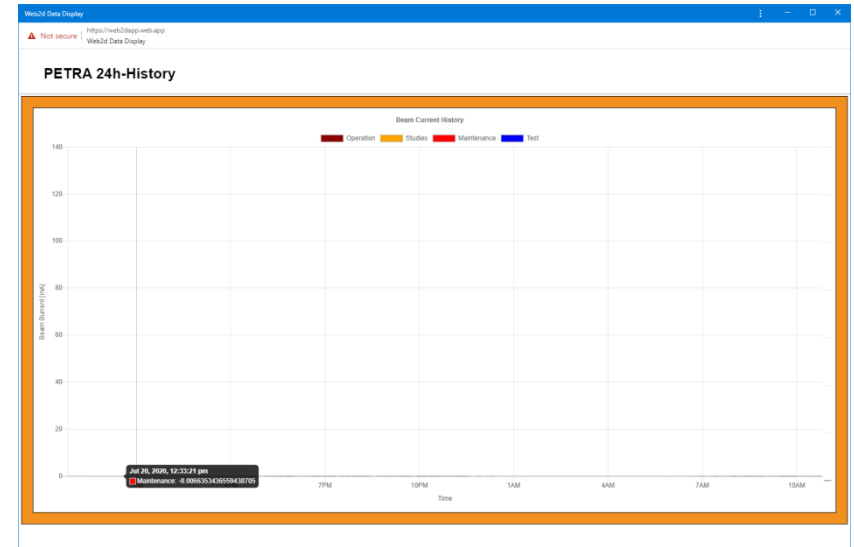
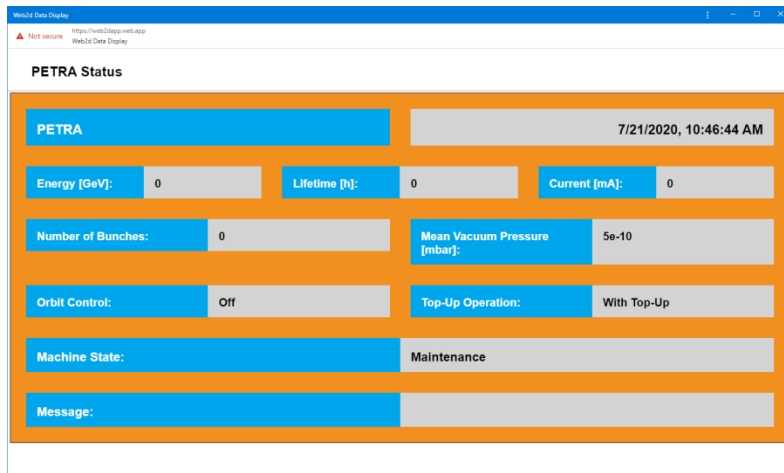
<https://ionicframework.com/>

# Project: Web Data Display

- Progressive Web App
- Implemented with Ionic / React
- Conceptual design is similar to jddd or Web2cToolkit
- Provides generic, customizable GUI widgets (e.g. Web2dPage, Web2dChart, Web2dTable, Web2dLabelValue, ...)
- Provides a run-time environment executing user – customizable GUI setups



# A Simple Example: PETRA Overview



The screenshot shows the 'PETRA Beamlines PU08 - PU14' table. The table has columns for PU08, PU09, PU10, PU11, PU12, PU13, and PU14. The 'Gap [mm]' row shows a value of 210 for all PUs. The 'Status' row shows green bars for all PUs, indicating they are operational.

	PU08	PU09	PU10	PU11	PU12	PU13	PU14
Gap [mm]	210	210	210	210	210	210	210
Status	Operational	Operational	Operational	Operational	Operational	Operational	Operational

<https://web2dapp.web.app>

# Preliminary Findings

- Easy app coding (in particular for JAVA programmers)
- Easy app deployment
- Responsivity is a fundamental app design feature (i.e. size / position of widgets adapts flexibly to app size / aspect ratio)
  - How to deal with detailed, fixed – size / position operation panels without losing benefit of app responsivity?
- PWA approach definitely applicable for
  - Dashboard-like applications (e.g. status panels, archive data viewer, ...)
  - Logbook applications
  - ...