

Conclusion

The browser: a wonderful application container platform

Press Space for next page →



Commonalities between frameworks

usage of components

Methodology to design your webApp

- Start With A Mock
- Step 1: Break The UI Into A Component Hierarchy
- Step 2: Build A Static Version in React, Vue.js, Angular
- Step 3: Identify The Minimal (but complete) Representation Of UI State
- Step 4: Identify Where Your State Should Live
- Step 5: Design your data-flow and the interactions between components

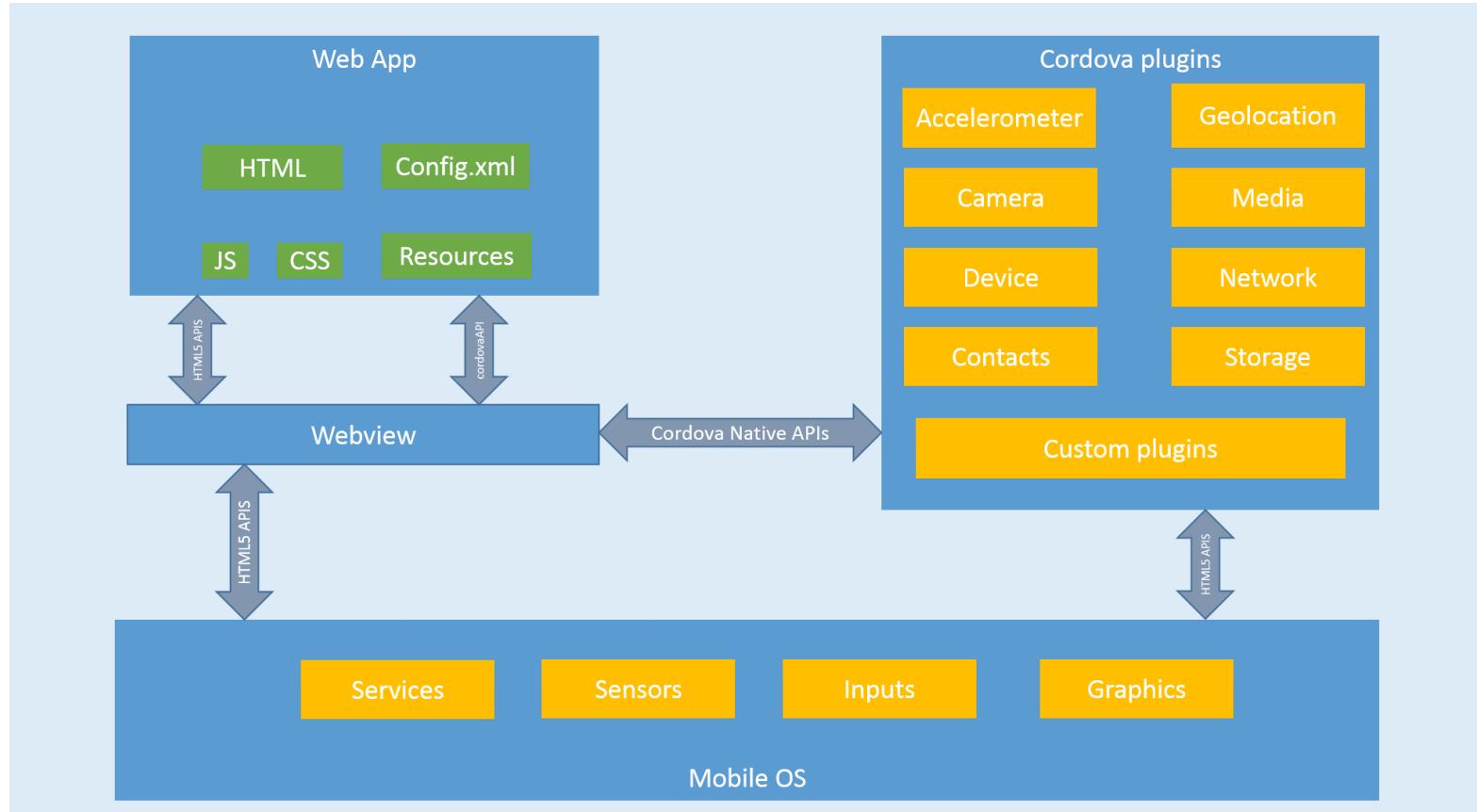
Personal recommandation

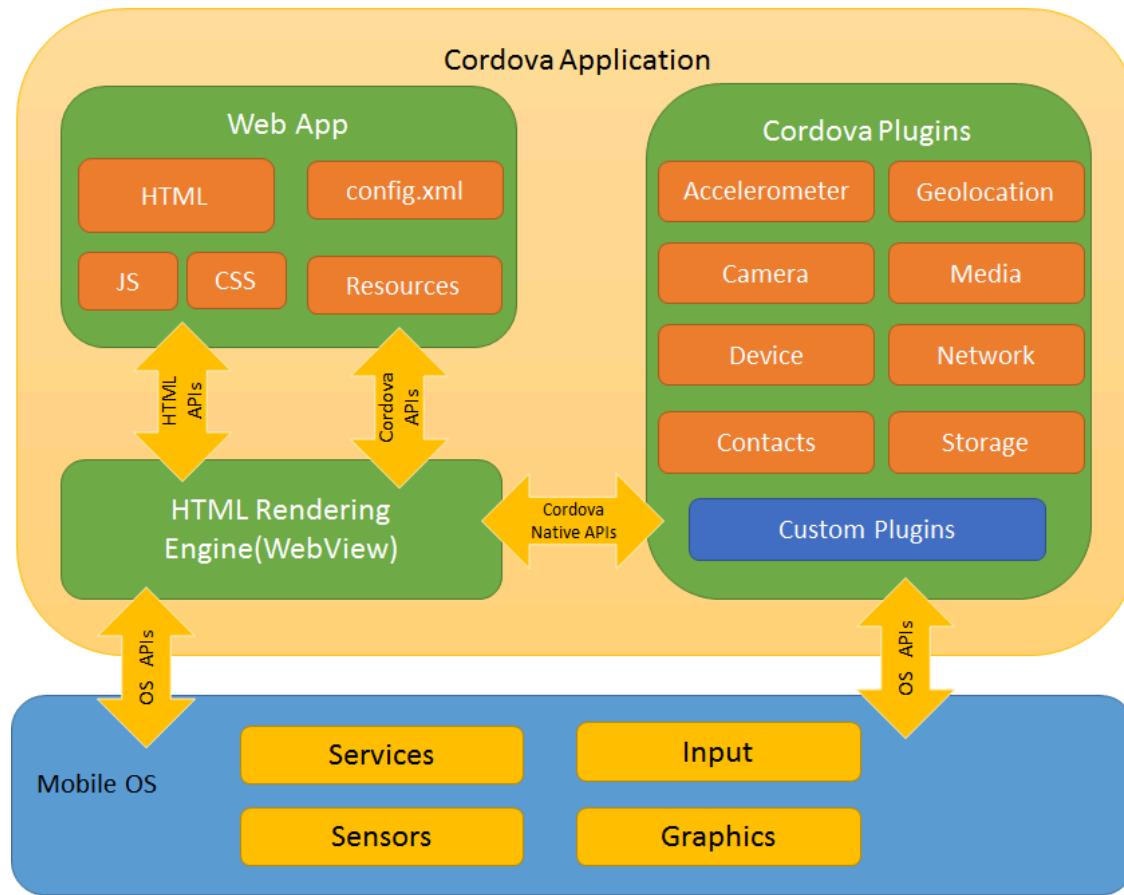
- Use TypeScript for all frameworks, even on the server side
- Use a CDN to deploy your front apps: You can use surge.sh, github page or netlify for deploying your SPA.

Hybrid development

- Build apps with web tech you know and love
- Ionic (Angular)
- ReactNative
- OnSen UI (Framework agnostic)
 - Written in pure JavaScript.
 - Support JQuery, AngularJS, Angular, React, Vue

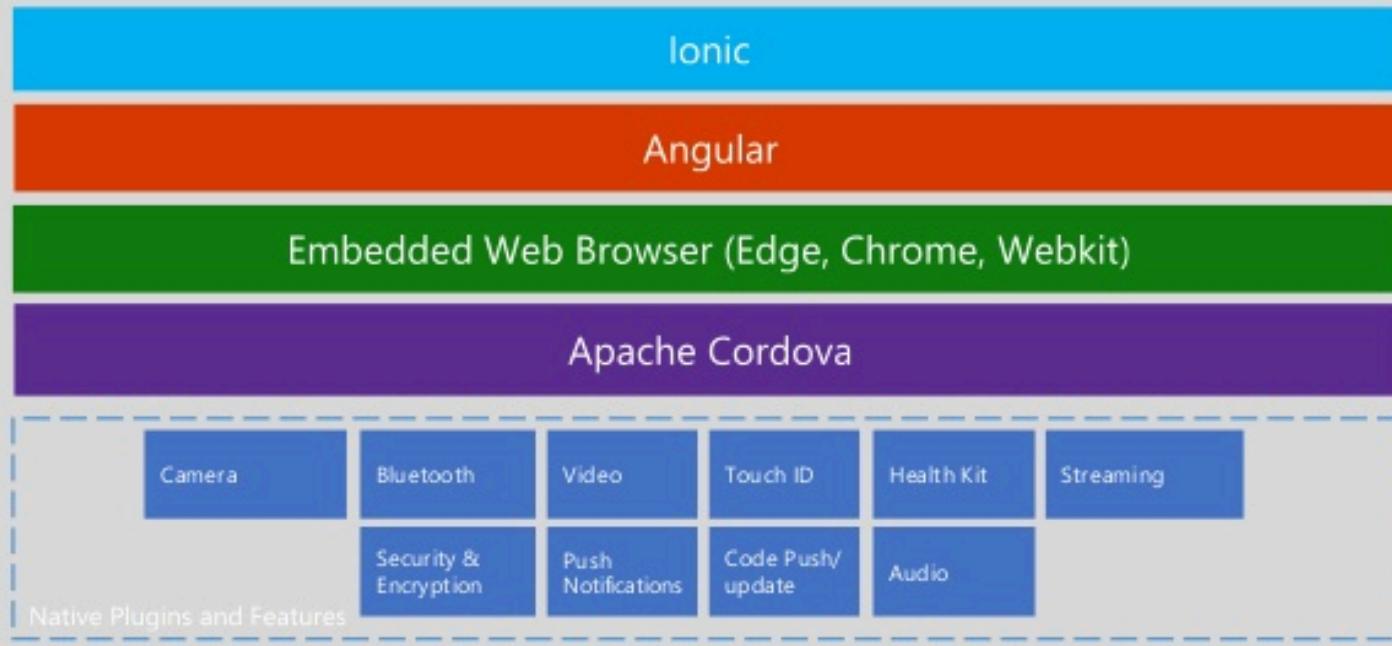
Hybrid development (Apache cordova)



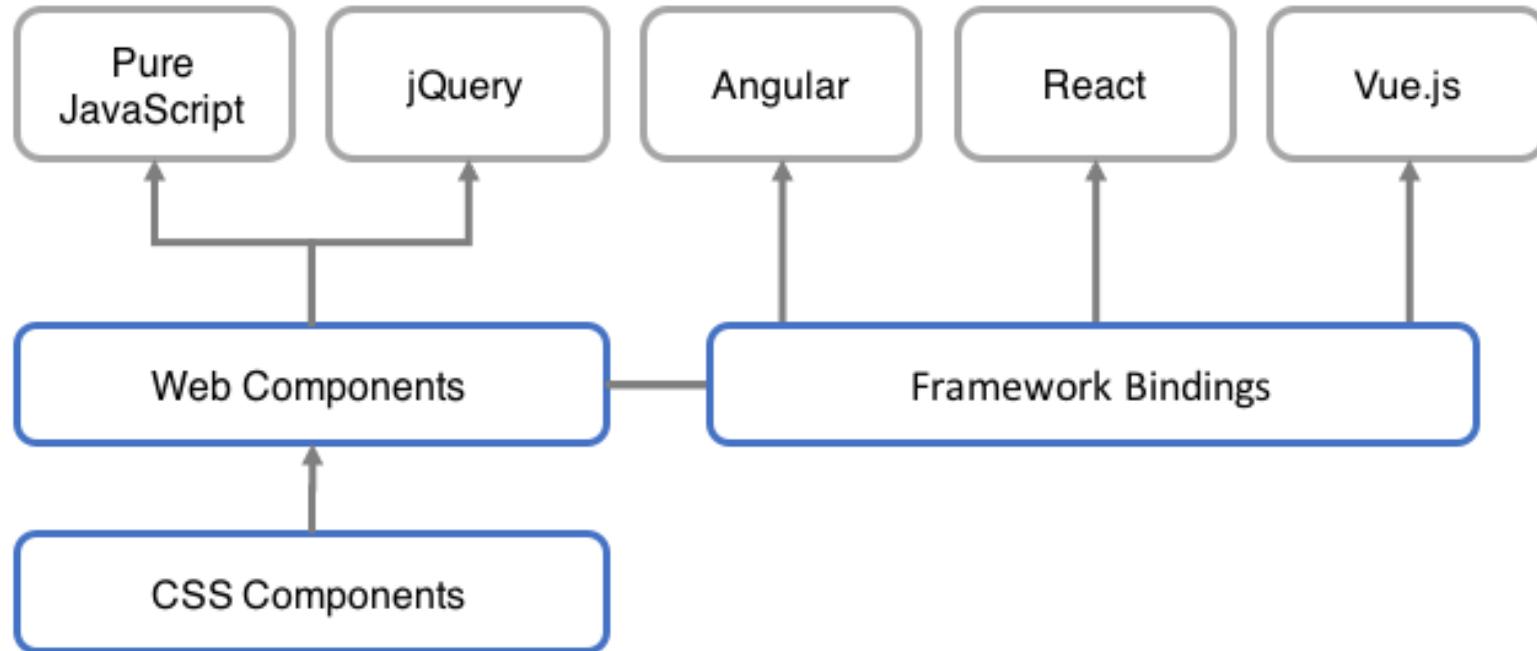


Ionic architecture

Full Runtime Architecture



OnSen architecture



Links

- <https://vuejs.org/>
- <https://facebook.github.io/react/>
- <https://angular.io/>
- <https://angularjs.org/>
- <http://getbootstrap.com/>
- <https://jquery.com/>
- <https://facebook.github.io/react-native/>
- <https://ionicframework.com/>
- <https://cordova.apache.org/>
- <https://onsen.io/>
- <https://books.ninja-squad.com/>