

Cross Platform Development

Múlt, jelen és jövő



Magamról

- Metál Ádám - nak hívnak
- A viddo.com nál dolgozom
- JavaScript fejlesztő vagyok
- <https://github.com/devmetal>

Több mint egy platform

- Telefonok, tabletek, tv, asztali alkalmazás, böngészős alkalmazás
- Külön az adott platformra specializálódott szakemberek kellene.
- A felmerülő problémák is különbözhetnek.
- A megoldás sem feltétlenül egységes.
- Külön kódhalmaz, minden platformra.

JavaScript SPA

Az SPA, vagy single page application alapvetően a böngészőre tervezett és javascript nyelven írt vagy azon futó program. Az JS állítja elő a különböző nézeteket, kommunikál a backendel, és üzleti logikát is megvalósít.



Első cross platform megoldások

A legtöbb platform támogat valamilyen módot, hogy alkalmazásainkban egy böngésző engine fusson. Innen az ötlet:

Miért ne készítsünk spa-kat a telefonokra, azok böngésző motorjait beágyazva natív alkalmazásokba?

Időrend: Android (2008), AngularJS (2010), Phonegap (2012), ES6 (2015)



Cordova/Phonegap

Vanilla JavaScript
Rendelkezik cli-vel

2012



Ionic

A cordova biztosítja az alapot
Angular-t használ

2013



jQuery Mobile

jQuery alapú
ui framework

2010

Előnyök hátrányok

- JavaScript/TypeScript alapú.
- Jól megszokott html, css.
- Egy kódbázis.
- Egységes layout, ui.
- Sebességben jóval lemaradnak
- A komponensek, nem keltenek natív hatást.
- Limitált hozzáférés a telefonok api-jához.

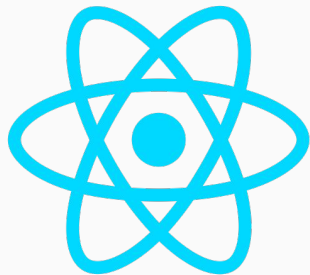
JavaScript engine az os-ben

A JavaScript aszinkron működése lehetővé tesz egy új fajta megközelítést.

Az ötlet, hogy az alkalmazást javascript vezérli egy natív hídon keresztül.

Például a yoga a react native esetében

A környezet, ahol a javascript fut pedig bele van ágyazva az alkalmazásba.
Ez nem egy böngésző, hanem egy javascript engine.



React Native

JSX/JavaScript/TypeScript

2015



NativeScript

Angular/Vue
Saját xml template

2014



Vue Native

React Native engine-t
használ

2018



Flutter

Dart

2017

Előnyök, hátrányok

- Egy kódbázis.
 - Egységes layout.
 - Hozzáférés a natív api-hoz
 - Natív sebesség a legtöbb esetben
 - Ismerős fe technológiák használhatók
- Bonyolult architektúra
 - Sok esetben szükséges valódi natív tudás és tapasztalat

React Native

Natív komponensek

Flex implementáció a layout managementhez, css használható egy stylesheet objektumon keresztül.

Számos web api implementáció megtalálható, például fetch

Remek tooling, jó közösség, és számos közösségi komponens

Vue Native

A react native-ot használja inkubátornak.

Mindent tud amit a react native.

Emellett megvan benne a vue összes tulajdonsága.

Aki kedveli az angular two-way data binding módszerét, de szeretne komponens orientáltan fejleszteni, annak szeretettel ajánlom

NativeScript

Használhatunk vue, és angular komponenseket

Saját template nyelve van, amely xml alapú

Architektúrája alapvetően hasonlít a react native hoz

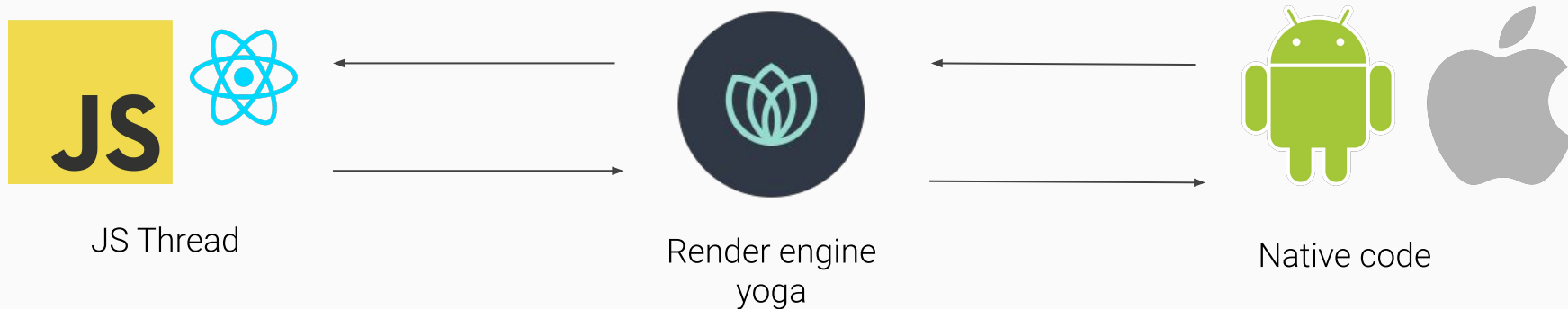
Flutter

Dart alapú

Megoldásban hasonlít az előzőekhez, a js runtime helyett azonban egy dart runtime van beépítve

Egyedi ui megoldás, *jobban hasonlít egy játék platformra*. Egy canvas technológiát használ

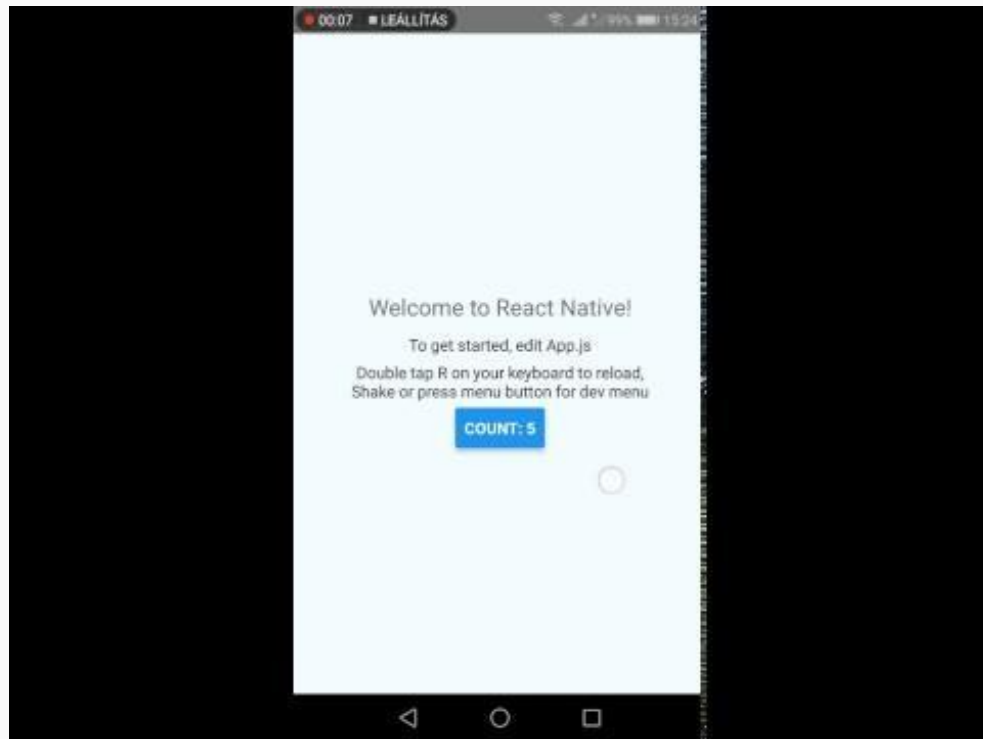
React Native



- UI Thread (Natív Kód)
- React Native Rendering engine (yoga)
- JavaScript Thread

React Native hasznos projektek

- Navigáció: <https://reactnavigation.org/en/>
- Állapot management:
 - React Hooks
 - React Context
 - Simple State
 - Redux
- Animációk:
 - <https://facebook.github.io/react-native/docs/animations>
 - <http://facebook.github.io/react-native/blog/2017/02/14/using-native-driver-for-animated.html>



Merre tartunk?

- **React Native**
 - Új többszálú render engine (Fiber)?
 - Egyedi JavaScript VM-ek
 - Több komponens, letisztult api.
- **NativeScript**
 - Fejlesztői élmény növelése
 - Teljesebb vue támogatás

Merre tartunk?

- **vue native**
 - Stabilizáció
 - Nincs túl sok elérhető infó
- **flutter**
 - Támogatni nem csak a mobilos platformokat
 - Szorosabb integráció a google egyéb szolgáltatásaival
 - Stabilizáció
 - Sebesség növelés

Köszönöm a figyelmet!