

ჰიბრიდული მობილური პროგრამირების შესაძლებლობები და თანამედროვე გამოწვევები

ბიორგი კაკაშვილი ინჟინერიის დოქტორი ინფორმატიკაში საქართველოს ეროვნული თავდაცვის აკადემია

აბსტრაქტი

დღეისათვის ინფორმაციული ტექნოლოგიების ეფექტიანად გამოყენების გარეშე, ნებისმიერ სფეროში, მნიშვნელოვანი პროგრესის მიღწევა პრაქტიკულად წარმოუდგენელია. თანამედროვე მსოფლიოში 5.67 მილიარდზე მეტი მობილური ტელეფონის მომხმარებელია. შესაბამისად, სულ უფრო მზარდია მოთხოვნა მობილური აპლიკაციების გამოყენებაზე. სწორედ ამიტომ ყოველდღიურად იზრდება მათი ჯამური რაოდენობა. ბიზნესისთვის აპლიკაციები უფრო სწრაფი გზაა ფართო აუდიტორიასთან დასაკავშირებლად. თუმცა მათ შესაქმნელად საუკეთესო პლატფორმის არჩევის შესახებ გადაწყვეტილების მიღება არც თუ ისე მარტივია.

ნაშრომის მიზანია, განვიხილოთ ჰიბრიდული მობილური პროგრამირების შესაძლებლობები და წარმოვაჩინოთ ამ მეთოდის უპირატესობა. მშობლიური მეთოდისგან განსხვავებით ამ დროს არ არის საჭირო რამოდენიმე დამოუკიდებელი პროექტის შექმნა სხვადასხვა ოპერაციული სისტემისთვის. ჩვენ შეგვიძლია ჰიბრიდული მეთოდით შევქმნათ მობილური აპლიკაცია, რომლის ერთი კოდური ბაზიდან ხორციელდება გაშვება ყველა თანამედროვე მობილურ ოპერაციულ სისტემაზე.

საკვანძო სიტყვები: ინფორმაციული ტექნოლოგიები, მობილური აპლიკაციები, ჰიბრიდული მობილური პროგრამირება, თანამედროვე ტექნოლოგიები.

Hybrid mobile programming capabilities and modern challenges

Giorgi Kakashvili
Doctor of Engineering in Informatics
Georgian National Defense Academy

Abstract

Today making significant progress is virtually unthinkable without the effective use of information technology. There are more than 5.67 billion mobile phone users in the modern world. Consequently, there is a growing demand for the use of mobile applications. That is why their total number is increasing daily. For businesses, apps are a faster way to connect with a wider audience. However, deciding on the best platform to create them is not so simple.

The aim of this paper is to discuss the possibilities of hybrid mobile programming and to demonstrate the advantages of this method. Unlike the native method, at this time it is not necessary to create several independent projects for different operating systems. We can use a hybrid method to create a mobile application that runs from a single code base on all modern mobile operating systems.

Key words: Information Technology, Mobile applications, Hybrid mobile programming, Modern Technologies.

შესავალი

მიმდინარე ტექნოლოგიური განვითარების შედეგადმა პროცესმა, თანამედროვე მსოფლიოში, საზოგადოება ახალი გამოწვევების წინაშე დააყენა. დღეისათვის ინფორმაციული ტექნოლოგიების ეფექტიანად გამოყენების გარეშე, ნებისმიერ სფეროში, მნიშვნელოვანი პროგრესის მიღწევა პრაქტიკულად წარმოდგენილია. ყველა ის სიკეთე რაც ტექნოლოგიურმა განვითარებამ მოგვითანა, მოწყობილობები, პროგრამები, შემუშავებულია და გამოიყენება სხვადასხვა მიზნის განსახორციელებლად და პრობლემის გადასაწყვეტად, როგორცაა, მაგალითად: ბიზნესის მოთხოვნების მოგვარება, ტურიზმის განვითარება, ახალი სერვისების შექმნა, თამაშების, სხვადასხვა ტიპის შინაარსის აპლიკაციების მიწოდება მომხმარებლებისთვის და სხვ (სურგულაძე გ. 2020).

მოგესხებათ, რომ ელექტრონული კომერციის ბაზარი ელვის სისწრაფით ვითარდება, იქმნება ახალი კომპანიები, მათ შორის სტარტაპები, რის შედეგადაც ბაზარზე კონკურენციამ ახალ სიმაღლეებს მიაღწია. აქედან გამომდინარე, ყველა კომპანიისთვის გახდა აუცილებელი, რომ ჩაერთონ ინტერნეტში და სწრაფად მიაწვდინონ ხმა უფრო და უფრო მეტ მომხმარებელს. დღეს თითქმის ყველა ადამიანი ფლობს სმარტფონს. აქედან გამომდინარე, მობილური პროგრამები გახდა მოსახერხებელი გზა, დაუკავშირდე უფრო მეტ ადამიანს და განავითარო ურთიერთობები ვირტუალურ მომხმარებლებთან და კლიენტებთან (Angelova 2019).

შესაბამისად, ყოველდღიურად იზრდება მობილური აპლიკაციების ჯამური რაოდენობა. ბიზნესისთვის აპლიკაციები უფრო სწრაფი გზაა ფართო აუდიტორიასთან დასაკავშირებლად. თუმცა მათ შესაქმნელად საუკეთესო პლატფორმის არჩევის შესახებ გადანწყვეტილების მიღება არც თუ ისე მარტივია, ხშირად განიხილება სხვადასხვა ვარიანტები, როგორცაა Native app and Hybrid app დეველოპმენტი. სწორი ვარიანტის შერჩევა დამოკიდებულია სხვადასხვა ფაქტორზე, მათ შორის: პროექტის ვადები, პროექტის ბიუჯეტი, არსებული გუნდის გამოცდილება და რაც მთავარია სამიზნე აუდიტორია (სურგულაძე გ. 2020).

დღესდღეობით კომპანიების უმრავლესობა უპირატესობას ანიჭებს ჰიბრიდულ მობილურ პროგრამირებას, რადგან ერთი კოდური ბაზიდან შესაძლებელია აპლიკაცია გაეშვას ყველა მობილურ ოპერაციულ სისტემაზე, ეს იქნება Android თუ iOS. ჰიბრიდული მეთოდის გამოყენება დეველოპერებს უადვილებს აპლიკაციების შექმნას, მათ უნევთ მხოლოდ ერთი კოდის დაწერა ყველა სახის მობილური ინტერფეისისთვის, რაც ასევე იწვევს სამუშაო დროის ეკონომიას.

ძირითადი ნაწილი

როდესაც საუბარია აპლიკაციის შექმნაზე, ამ დროს პროგრამის ბაზად განიხილება ძირითადად ორი გიგანტი მობილური ოპერაციული სისტემა: Android და iOS. ეს ორი უალტერნატივო გადანწყვეტა ხდება მათი პოპულარობის, ეფექტიანობისა და მომხმარებლის კმაყოფილების გათვალისწინებით. თუმცა, საერთო დილემა, რომლის წინაშეც უმეტესი ადამიანი დგება, არის ის, თუ რომელი ოპერაციული სისტემა უნდა აირჩეს - Android თუ iOS. პირველი მარტივი გამოსავალი, რომელზეც შეიძლება იფიქროთ, არის ორივე პლატფორმის ერთდროულად შემუშავება, მაგრამ ამისთვის ყველას საკმარისი ბიუჯეტი არ აქვს. ამ დროს ერთი აპლიკაციისთვის იქმნება ორი დამოუკიდებელი პროექტი, რომელიც ზრდის ფინანსებს და დროს. თანამედროვე სამყაროში მოცემულ პრობლემას მარტივი გადაჭრის შესაძლებლობა აქვს. ჩვენ შეგვიძლია ჰიბრიდული მეთოდით შევქმნათ მობილური აპლიკაცია, რომლის ერთი კოდური ბაზიდან ხორციელდება გაშვება ყველა თანამედროვე მობილურ ოპერაციულ სისტემაზე (Enihe Raphael 2020).

ჰიბრიდული მობილური აპლიკაციის პროექტი იწყება იდენით და ამ იდენის რეალიზაციისთვის აუცილებელი მოთხოვნების მიზანისა და ამოცანების განსაზღვრით. პროექტის ერთ-ერთი მთავარი და რთული საკითხი პრობლემის იდენტიფიკაციაა. პრობლემა სწორად უნდა შეირჩეს, ზუსტად უნდა იყოს ფორმულირებული და მისი გადანწყვეტისთვის დაისახოს ისეთი ამოცანები, რომელიც ყველასთვის მისაღებია. პროექტზე მუშაობს გუნდი და დასაწყისიდან დასასრულამდე გადის 5 ძირითად ეტაპს:

- კონცეპტუალიზაცია (სტრატეგიის შემუშავება);
- მიზანშეწონილობის შეფასება (ანალიზი, დაგეგმვა);
- აპლიკაციის დიზაინის შემუშავება (UI/UX);
- პროგრამული უზრუნველყოფა (App Development);
- ტესტირება და გამოქვეყნება (დანერგვა, თანხლება).

კონცეპტუალიზაცია - ჰიბრიდული მობილური პროგრამის განვითარების (დეველოპმენტის) პროცესში

პირველი და ყველაზე მნიშვნელოვანია სწორი კონცეფციის და მისი რეალიზაციის მოთხოვნების ჩამოყალიბება. აპლიკაციის სანყის ანალიზი უნდა შეიცავდეს სამიზნე ჯგუფის განსაზღვრას, ქცევის მოდელებს და მიზნებს.

მიზანშეწონილობის შეფასება - აპლიკაციაზე მომუშავე გუნდმა უნდა განსაზღვროს თუ რამდენად მიზანშეწონილია პროექტის განხორციელება. უკვე კონცეპტუალიზებული პროექტის კარკასული მოდელის და დეტალური ესკიზების საფუძველზე, შესაძლებელია აპლიკაციის კონცეფციის ტექნიკურად რეალიზებადობის მკაფიოდ დანახვა. ამ ეტაპზე უნდა შეფასდეს თუ რამდენად გაუძლებს ეს აპლიკაცია კონკურენციას და რამდენად იქნება აქტუალური ბაზარზე.

აპლიკაციის დიზაინის შემუშავება მომხმარებლის ინტერფეისის დიზაინის ფორმირება არის საკმაოდ რთული პროცესი და მოიცავს სხვადასხვა ღონისძიებას. უნდა შეიქმნას აპლიკაციის დიზაინი, რომელიც ინიცირებულია მომხმარებლის გამოცდილებიდან, სადაც გათვალისწინებული იქნება მრავალი კონცეფცია და მოსაზრება (UI/UX). აუცილებელია სამუშაო ჯგუფის მიერ შემუშავდეს ინდივიდუალური ფორმა, მაგრამ მასში საჭიროა დაცულ იქნას მარტივი დიზაინური აღნიშვნები, რათა მომხმარებელში არ გამოიწვიოს გაურკვევლობა რეალურ მნიშვნელობაზე. დიზაინის შემოშავების დროს ერთ ერთი უმთავრესია მომხმარებლის გამოცდილების გათვალისწინება. წლების განმავლობაში მომხმარებელი მოერგო ტექნოლოგიების და თავისი მოქმედებებით განსაზღვრა სხვადასხვა ფაქტორი. შესაბამისად ახალი პროდუქტის შექმნის და რეალიზაციის დროს უნდა იყოს განსაზღვრული ეს გამოცდილება რათა მომხმარებლებმა არ იგრძნონ დისკომფორტი (სურგულაძე გ. 2020).

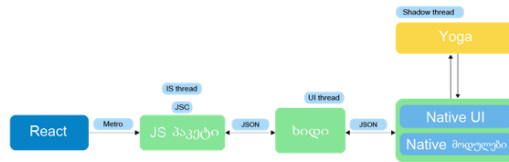
პროგრამული უზრუნველყოფის ეტაპი არის ყველაზე საპასუხისმგებლო, ყველაზე შრომატევადი და ყველაზე რთული პროცესი. ამ დროს დეველოპერები განიხილავენ ორ შესაძლებლობას: ჰიბრიდულ და ტრადიციულ მეთოდს. სწორი ვარიანტის შერჩევა დამოკიდებულია უამრავ ფაქტორზე, რომელიც აპლიკაციის შექმნის დროს უნდა გავითვალისწინოთ. მაგალითად პროექტის ვადები, პროექტის ბიუჯეტი, არსებული გუნდის გამოცდილება ამა თუ იმ ტექნოლოგიასთან მიმართებით და რაც მთავარია სამიზნე აუდიტორია. თანამედროვე კომპანიების უმრავლესობა უპირატესობას ანიჭებს ჰიბრიდულ მობილურ პროგრამირებას, რადგან ერთი კოდური ბაზიდან ძალიან მარტივად შესაძლებელია აპლიკაცია გაეშვას ყველა მობილურ ოპერაციულ სისტემაზე, ეს იქნება Android თუ iOS.

სწორედ ამ უპირატესობიდან გამომდინარე ჰიბრიდული მეთოდის გამოყენება დეველოპერებს უადვილებს აპლიკაციების შექმნას, მათ უწევთ მხოლოდ ერთი კოდის დანერგვა ყველა სახის მობილური ინტერფეისისთვის, რაც ასევე თავის მხრივ იწვევს სამუშაო დროის დაზოგვას და შედარებით მცირე ბიუჯეტით სრულყოფილი პროდუქტის შექმნას. ჰიბრიდული მეთოდით აპლიკაციების შექმნა არის მობილური პროგრამირების მიდგომა, აპლიკაციების შემუშავების პროცესი, რომელიც საშუალებას გვაძლევს შევქმნათ და გამოვაქვეყნოთ ისინი მსოფლიოში არსებულ ნებისმიერი მწარმოებლის ნებისმიერ მობილურ ტელეფონზე. მიუხედავად იმისა რომ ეს მიდგომა არც თუ ისე ახალია, დღეს უამრავი კომპანია ქმნის ჰიბრიდულ მობილურ აპლიკაციებს. ამავდროულად აღსანიშნავია ისიც, რომ ამ მიმართულებით იქმნება უამრავი სამუშაო ჩარჩოები. მათ შორის ერთ-ერთ მონიშნულ ადგილს იკავებს React Native. ის არის კროს-პლატფორმული სამუშაო ჩარჩო, რომელიც დაინერგა Facebook-ის მიერ 2015 წელს. მას აქვს უამრავი დადებითი მხარე რის ხარჯზეც მალევე მოიპოვა პოპულარობა და გახდა საუკეთესო ინსტრუმენტი: ეკონომიური, მასშტაბური და მიმზიდველი აპლიკაციების შესაქმნელად (ჩოგოვაძე გოჩა 2017).

React Native არის პლატფორმის აგნოსტიკური გადაწყვეტა, რაც იმას ნიშნავს, რომ მას არ აქვს რაიმე კავშირი კონკრეტულ პლატფორმასთან და შეიძლება გამოყენებულ იქნას ერთზე მეტი ოპერაციული სისტემისთვის. React Native დეველოპერები წერენ ერთ კოდურ ბაზას JavaScript კოდის გამოყენებით.

დღესდღეობით React Native-ის სრული არქიტექტურა ემყარება სამ ძირითად საყრდენს (სურ. 1):

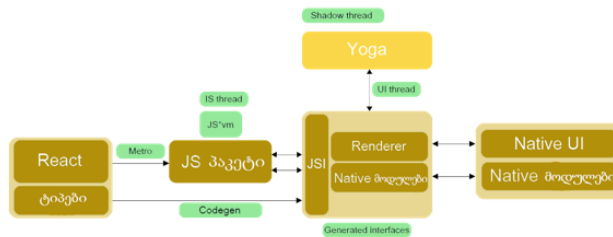
- JavaScript Thread, ეს არის ადგილი, სადაც მთელი JavaScript კოდი განთავსებული და კომპილირებული.
- Native Thread, ეს არის ადგილი, სადაც მშობლიური კოდი სრულდება. ეს კომპონენტი ამუშავებს მომხმარებლის ინტერფეისს და უზრუნველყოფს შეუფერხებელ კომუნიკაციას JS თემასთან, როდესაც აპლიკაციის სჭირდება ინტერფეისის განახლება, მშობლიური ფუნქციების გაშვება და ა.შ.
- Shadow Thread, ეს არის ადგილი სადაც გამოითვლება აპლიკაციის განლაგება. ის გარდაქმნის flex-box-ის განლაგებას, ითვლის მათ და აგზავნის აპლიკაციის ინტერფეისში. მისი სახელწოდებაა Yoga (Gallego 2020).



სურ. 1. React Native-ის სრული არქიტექტურა

თუ კარგად დავაკვირდებით React Native განლაგებას, დავინახავთ, რომ იგი იყენებს ხიდს მშობლიურ მოდულებთან კომუნიკაციისთვის, რაც ხშირად ქმნის რიგს, რადგან ამ ორმა მხარემ არ იცის ერთმანეთის შესახებ. ასეთმა პროცესმა შეიძლება შექმნას შესრულების შეზღუდვები, რაც უარყოფითად იმოქმედებს მომხმარებლის გამოცდილებაზე. React Native-ს შემქმნელები რეგულარულად წარადგენენ განახლებებს, რომელიც მიზნად ისახავს დეველოპმენტის პროცესის გამარტივებას. უახლესი მაშტაბური განახლება მოხდა 2020 წლის 6 ივლისს, როდესაც Facebook-ის გუნდმა შეცვალა შეცდომის, გაფრთხილებისა და აუზრუნველყოფის მთელი სისტემა განახლებული LogBox-ის დანერგვით.

მთელი React Native საზოგადოება ელოდება ახალი არქიტექტურის დანერგვას, რომელიც მოიტანს მნიშვნელოვან გაუმჯობესებას, რაც გამარტივებს დეველოპმენტის პროცესს და გახდის მას უფრო მოსახერხებელს ყველა მონაწილე მხარისთვის (სურ 2).



სურ. 2. React Native-ის ახალი არქიტექტურა

ახალი არქიტექტურა მოგვცემს ნებისმიერი JS ძრავის გამოყენების შესაძლებლობას. JavaScript თემა აღარ იქნება მიბმული JavaScriptCore-თან, რაც საშუალებას მოგვცემთ გამოიყენოთ ნებისმიერი მაღალი ხარისხის JS ძრავა. React Native-ის ხელახალი არქიტექტურის დანერგვით, ხიდი თანდათან აღმოიფხვრება და ჩანაცვლდება ახალი კომპონენტით, სახელწოდებით JavaScript ინტერფეისი (JSI). JSI-ის მიერ მოტანილი კიდევ ერთი უპირატესობა არის JS ძაფებისა და მშობლიური მოდულების მთელი სინქრონიზაცია. JavaScriptInterface-ის დახმარებით JS-ს შეეძლება Hot Objects-ზე მითითება და მათზე მეთოდების გამოძახება. მას ასევე ექნება საერთო საკუთრების კონცეფცია, რაც საშუალებას მისცემს მშობლიურ მხარეს პირდაპირ დაუკავშირდეს JS თემას. კიდევ ერთი სარგებელი, რომელსაც ახალი React Native აწვდის საზოგადოებას, არის მთლიანი კონტროლი მშობლიურ მოდულებზე. აპლიკაცია გამოიყენებს მათ მხოლოდ მაშინ, როცა ეს საჭიროა, ნაცვლად იმისა, რომ თავიდან გაუშვას ყველა კომპონენტი (Gallego 2020).

Fabric ძირითადად არის UI მენეჯერის ხელახალი არქიტექტურა, რომელიც გარდაქმნის რენდერის ფენას ხიდების საჭიროების აღმოფხვრის გზით. ახალი მიდგომა საშუალებას გვაძლევს შევქმნათ Shadow Tree პირდაპირ C++-ში, რაც გამოიწვევს პროცესის სისწრაფეს და ამცირებს კონკრეტული ელემენტის გამოსატანად საჭირო ნაბიჯების რაოდენობას.

React Native და Native ნაწილებს შორის უწყვეტი კომუნიკაციის უზრუნველსაყოფად, Facebook-ის გუნდი ამჟამად მუშაობს ინსტრუმენტზე, სახელწოდებით CodeGen. მოსალოდნელია ორი კვანძის თავსებადობის ავტომატიზაცია და მათი სინქრონიზაცია. ეს გენერატორი განსაზღვრავს TurboModules-ისა და Fabric-ისთვის საჭირო ინტერფეისის ელემენტებს. ეს განახლება გამოიწვევს კოდის დუბლირების აუცილებლობას ორივე ძაფისთვის და მოსალოდნელია უზრუნველყოფის გლუვი სინქრონიზაცია.

თუ უკვე გვაქვს შექმნილი აპლიკაცია ჰიბრიდული მობილური პროგრამირებით React Native-ზე კარგი ამბავი ის არის, რომ ახალი არქიტექტურის გამოჩენისას არ დაგვჭირდება მთელი აპლიკაციის გადანერვა თავიდან. მხოლოდ საჭირო იქნება ახალ ვერსიაზე განახლება და მთელი აპი ავტომატურად გადაინაცვლებს ახალ არქიტექტურაზე. ამრიგად, შეგვიძლია დავასკვნათ, რომ React Native-ის მომავალი საკმაოდ ნათელია და მასთან ერთად ჰიბრიდული მობილური პროგრამირების შესაძლებლობებიც.

დასკვნა

რამდენადაც მობილურ აპლიკაციებს მნიშვნელოვანი როლი უკავიათ ბიზნეს სამყაროში, დეველოპერები და ორგანიზაციები ყველა ღონეს იყენებენ ამ მიმართულებით საუკეთესო მიდგომის პოვნაში, მაგრამ მათი უმეტესობა აცნობიერებს, რომ ორივე მათგანს, როგორც ჰიბრიდულს ასევე მშობლიურს, აქვს თავისი დადებითი და უარყოფითი მხარეები. დღესდღეობით კომპანიებისთვის უფრო მიმზიდველად გამოიყურება ჰიბრიდული მობილური პროგრამირება, რადგან ერთი კოდური ბაზიდან შესაძლებელია აპლიკაცია გაეშვას ყველა მობილურ ოპერაციულ სისტემაზე. ჰიბრიდული მეთოდის გამოყენება, უამრავი დადებითი კომპონენტის გათვალისწინებით, მათ შორის დროის და ფინანსების ეკონომიურობის ხარჯზე, კომპანიებს უადვილებს აპლიკაციების შექმნას. ამავდროულად განსახილველია უფრო მეტი საკითხი, რომლებიც დაკავშირებულია უსაფრთხოებასთან, ინტეგრაციასთან და განახლებასთან. თუმცა პერსპექტივები რასაც გვთავაზობს ჰიბრიდული მობილური პროგრამირების ერთ-ერთი პლათფორმა React Native ახალი არქიტექტურის საშუალებით, გადაჭრის ყველა იმ პრობლემას და დაბრკოლებას რასაც დღეს დეველოპერები ხვდებიან. ჰიბრიდული აპლიკაცია არ არის მობილური აპლიკაციის შემუშავების კიდევ ერთი გზა. ის არის ბევრად მეტი რომელიც ტრადიციული მიდგომიდან განსხვავებით იძლევა კიდევ უფრო მეტ შესაძლებლობას, როგორც კომპანიებისთვის ასევე დეველოპერებისთვის. მობილური აპლიკაციის შექმნა ჰიბრიდული მეთოდის გამოყენებით გაზრდის ხარისხს, პროდუქტიულობას და შეამცირებს ღირებულებას, რომელიც უფრო თვალსაჩინო იქნება მომდევნო წლებში.

გამოყენებული ლიტერატურა

სურგულაძე გ., კაკაშვილი გ., მარტიაშვილი გ. 2020. მობილური აპლიკაციების დეველოპმენტის საფუძვლები (Java, Android). თბილისი: "სტუ-ს IT კონსალტინგის ცენტრი".

ჩოგოვაძე გოჩა, ფრანგიშვილი არჩილ, სურგულაძე გია. 2017. მართვის საინფორმაციო სისტემების დაპროგრამების ჰიბრიდული ტექნოლოგიები და მონაცემთა მენეჯმენტი. თბილისი: ტექნიკური უნივერსიტეტი.

Angelova, Nadezhda. 2019. "MOBILE APPLICATIONS FOR BUSINESS." *Trakia Journal of Sciences* 853-859.

Enihe Raphael, Joshua Jimmy. 2020. "https://www.researchgate.net/publication/342134953_HYBRID_MOBILE_APPLICATION_DEVELOPMENT_A_BETTER_ALTERNATIVE_TO_NATIVE." <https://www.researchgate.net/>. 05. Accessed 9 21, 2021. https://www.researchgate.net/publication/342134953_HYBRID_MOBILE_APPLICATION_DEVELOPMENT_A_BETTER_ALTERNATIVE_TO_NATIVE.

Gallego, Rémi. 2020. [www.medium.com](https://medium.com/swlh/react-natives-re-architecture-in-2020-9bb82659792c). 05 28. Accessed 09 22, 2021. <https://medium.com/swlh/react-natives-re-architecture-in-2020-9bb82659792c>.