Android, iOS and Hybrid Applications

Mobile-Development

OVERVIEW

- Hybrid Applications
- Interoperability with the native part
  - Design a possible interface
  - Present your approach
- Create a small working sample
- (Introduction to modern Web-Development)

HYBRID APPLICATIONS

- Native Part which provides a JS-Interface
- More then 50% market share (hard to prove)
- Browsers support HTML5/CSS3 and ES6
  - https://caniuse.com/

HYBRID APPLICATIONS

- Pros
  - Share or reuse (UI)-Code (from website etc.)
  - It‘s easier to find Web-Devs then native Devs
  - Possible to update without going trough the store
  - Fallbacks to native possible

HYBRID APPLICATIONS

- Cons
  - Sometimes don’t feel that “responsive”
    - Getting better with later releases
  - You need to understand both worlds (native & web)

WEBVIEWS

- iOS
  - WKWebView
  - **Don’t** use UIWebView
- Android
  - WebView
  - Updates independent of OS (since 4.4.4/19)

https://developer.apple.com/documentation/webkit/wkwebview

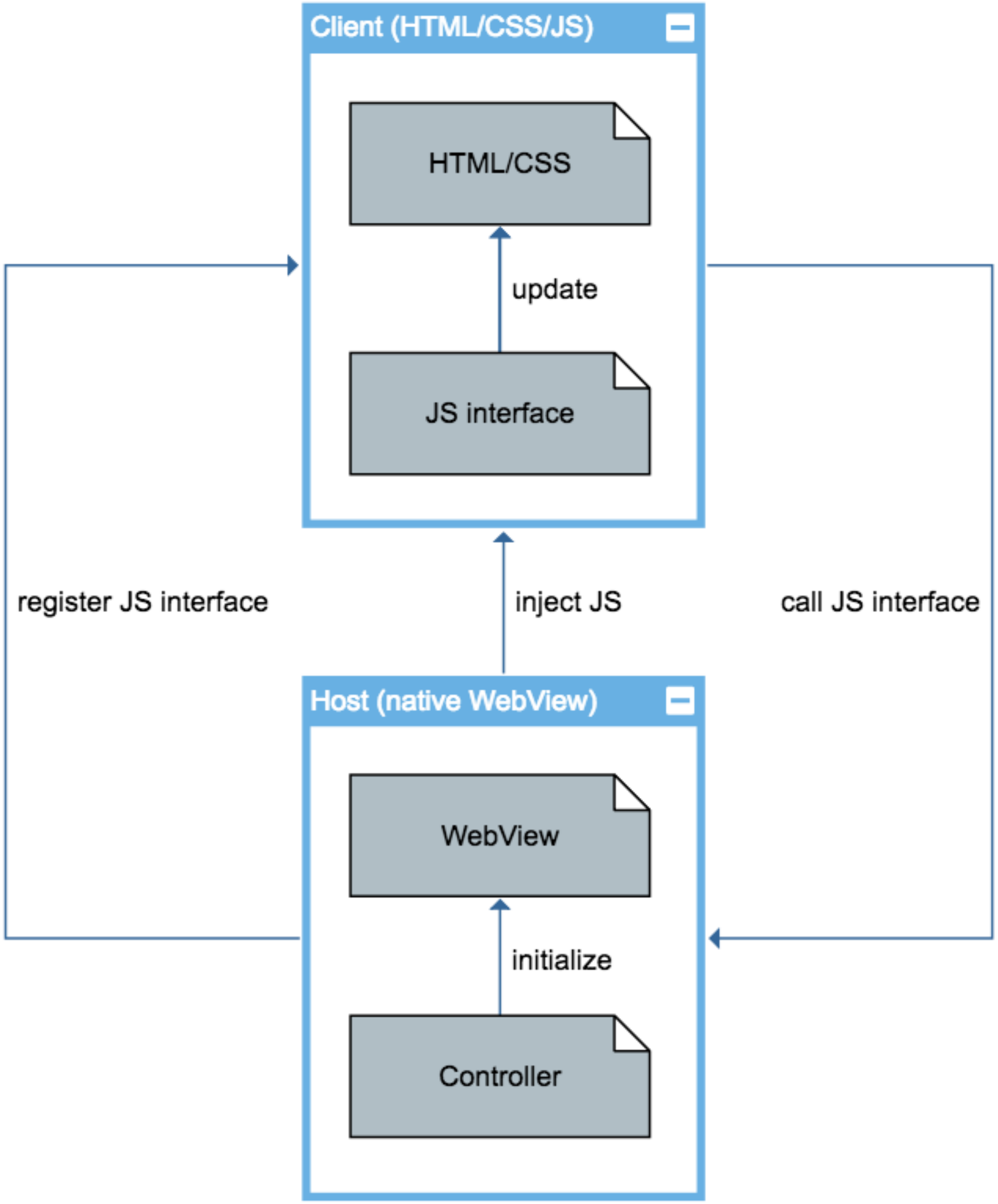
https://developer.android.com/reference/android/webkit/WebView

WEBVIEWS

- The control is a wrapper - they run in their own process
- You're not limited to only use one WebView
- Load local HTML pages or remote ones
- Think about CORS when using a mix

https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

ARCHITECTURE



Setup the App

- Clone the sample
  - https://github.com/FabrizioNiedda/webview-example
- It's a simple Android app (Xamarin)

REGISTER JS INTERFACE

```
var webView = (WebView)FindViewById(Resource.Id.webView);

webView.Settings.JavaScriptEnabled = true;

webView.AddJavascriptInterface(new JavaScriptInject(this), "Native");

webView.LoadUrl("file:///android_asset/index.html");
```

https://docs.microsoft.com/en-gb/dotnet/api/android.webkit.webview.addjavascriptinterface#Android_Webkit_WebView_AddJavascriptInterface_Java_Lang_Object_System_Strin

Inject JS (Native -> WebView)

```
webView.EvaluateJavascript("do some JS magic...", null);
```

```
webView.EvaluateJavascript("do some JS magic...", new Callback());

public class Callback : Java.Lang.Object, IValueCallback
{
  public void OnReceiveValue(Object value)
  {
    // Do something with the value...
  }
}
```

https://docs.microsoft.com/en-gb/dotnet/api/android.webkit.webview.evaluatejavascript

Example

- Walkthrough

Exercise

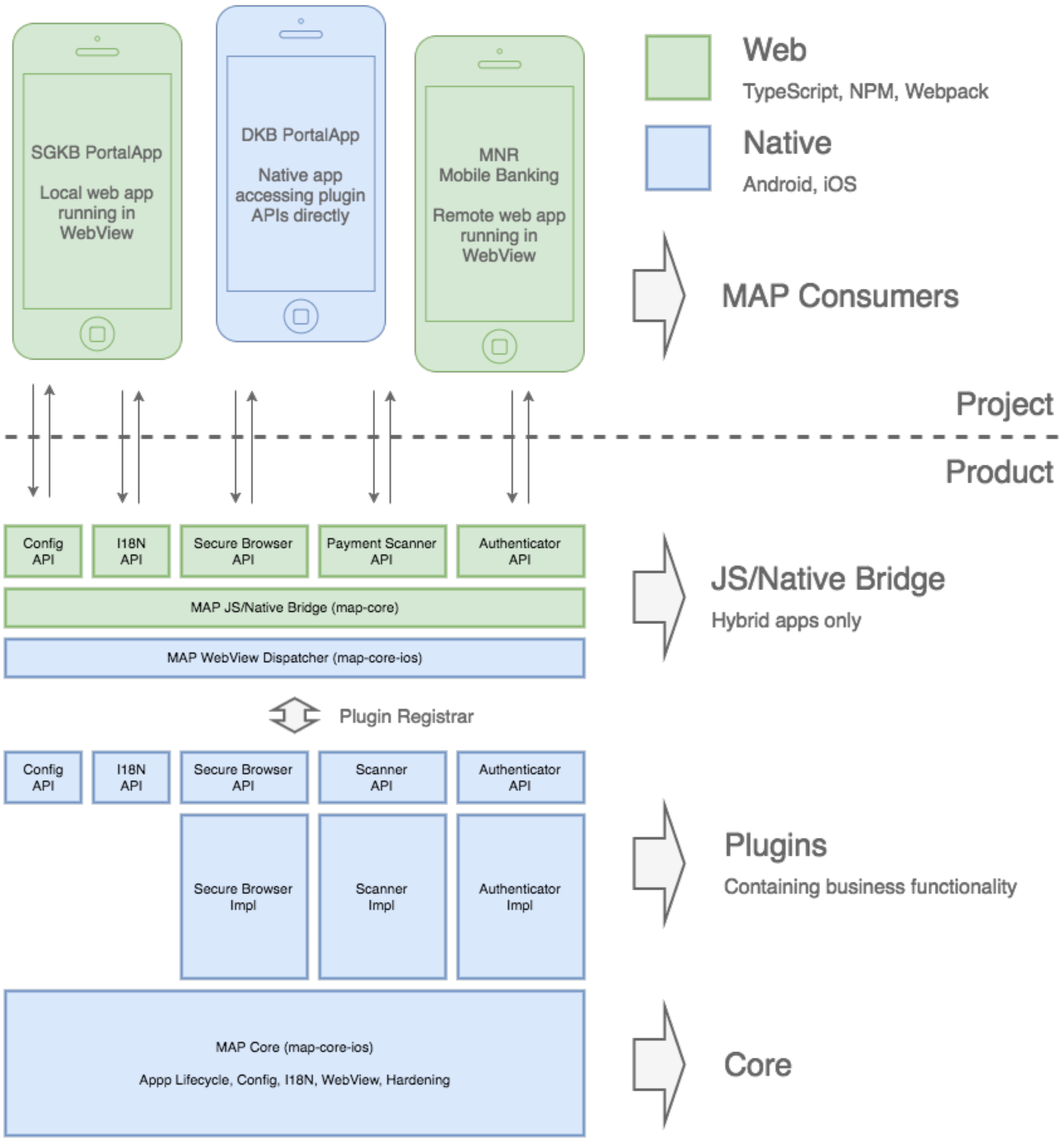
- Group up
- Setup the basic Android project
- Create the Repo for your group
- Think about an approach on how to create a messaging bus between Native and Web
- Present your solution/idea

Possible Solution

- Mediator pattern across Native/JS
- Send messages and distribute them
- Web “drives” the app
- Native is used like an “API”

https://www.dofactory.com/net/mediator-design-pattern

ARCHITECTURE



SGKB PortalApp

Local web app running in WebView

DKB PortalApp

Native app accessing plugin APIs directly

MNR Mobile Banking

Remote web app running in WebView

🟩 Web

TypeScript, NPM, Webpack

🟦 Native

Android, iOS

MAP Consumers

Project

Product

Config API

I18N API

Secure Browser API

Payment Scanner API

Authenticator API

MAP JS/Native Bridge (map-core)

MAP WebView Dispatcher (map-core-ios)

JS/Native Bridge

Hybrid apps only

Plugin Registrar

Config API

I18N API

Secure Browser API

Scanner API

Authenticator API

Secure Browser Impl

Scanner Impl

Authenticator Impl

Plugins

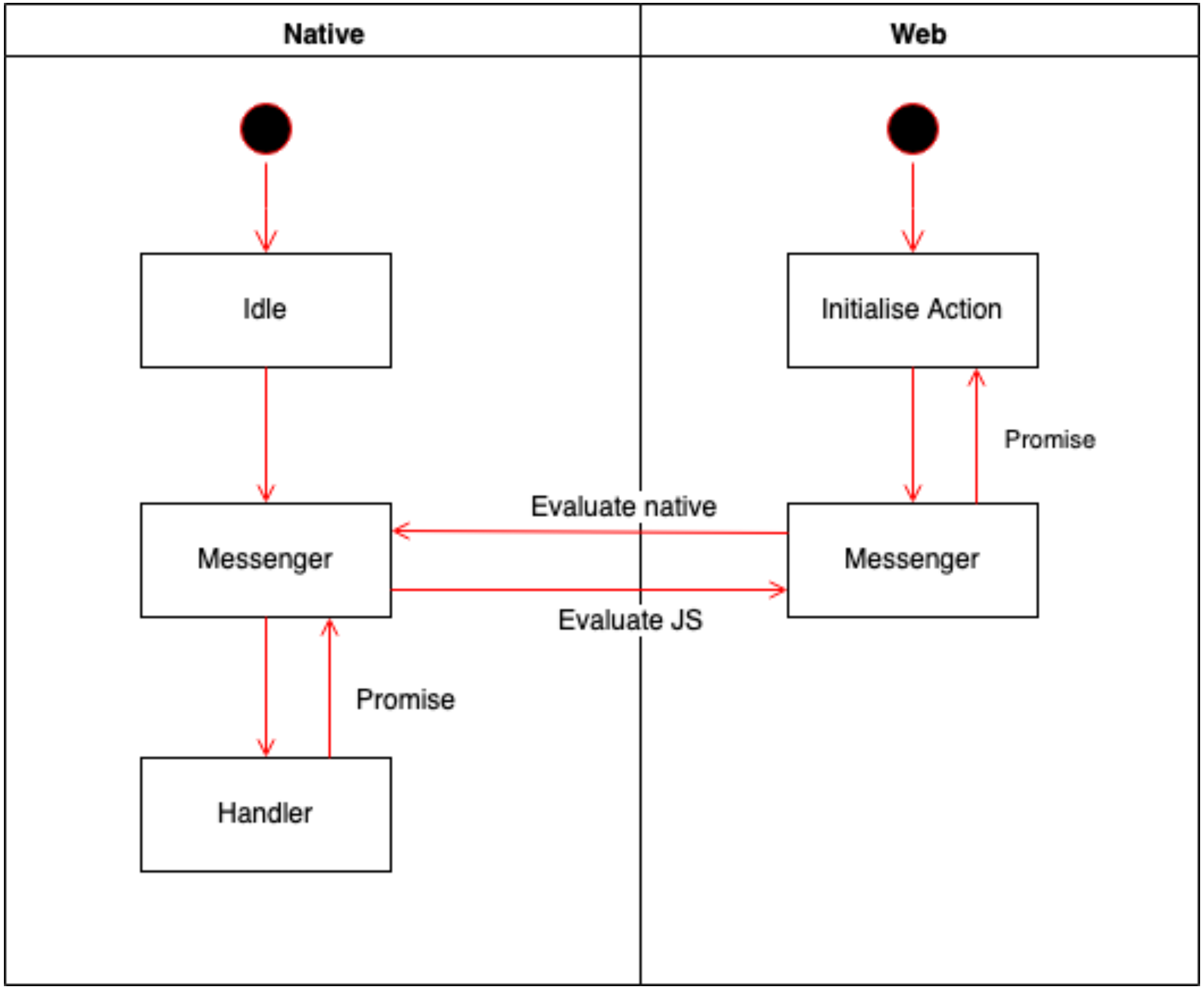
Containing business functionality

MAP Core (map-core-ios)

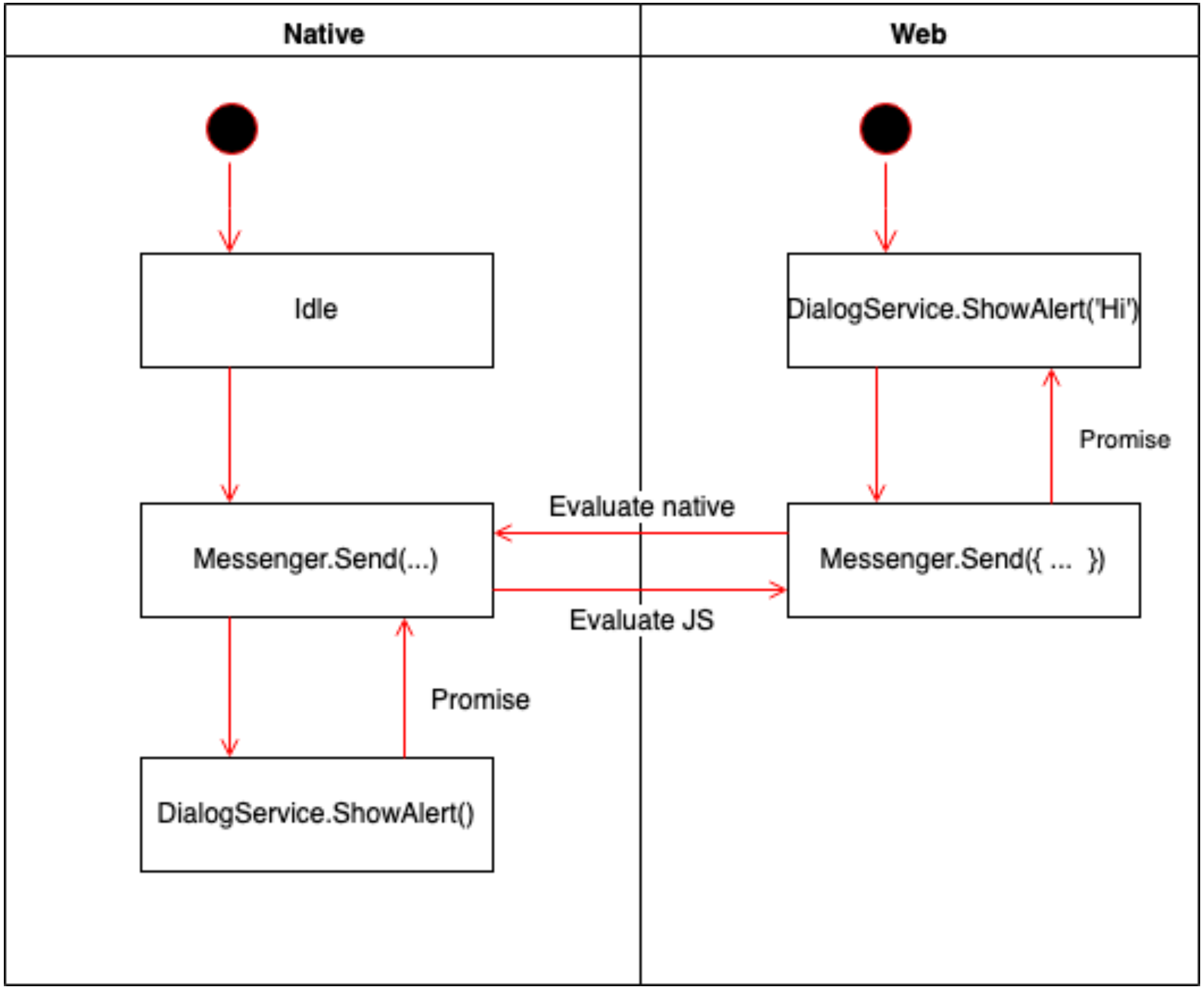
Appp Lifecycle, Config, I18N, WebView, Hardening

Core

WORKFLOW


Native
Web
Idle
Initialise Action
Promise
Evaluate native
Messenger
Messenger
Evaluate JS
Promise
Handler

WORKFLOW


Native
Web
Idle
DialogService.ShowAlert('Hi')
Promise
Evaluate native
Messenger.Send(...)
Messenger.Send({ ... })
Evaluate JS
Promise
DialogService.ShowAlert()

REST OF THE EVENING

- Continue working on the Forms application
- Try to setup the notifications