Onderstaand geeft de VRU een nadere toelichting op haar keuzes voor native iOS-applicaties en native iOS-integratie in plaats van hybride benadering.

1. Onderbouwing native iOS-applicaties

Een native iOS-applicatie biedt directe toegang tot de beveiligingsmechanismen van het iOS-platform, waaronder:

- **Enhanced Security-capabilities** zoals pointer authentication, hardened heap, en read-only platform memory;
- **Gebruik van de Apple Keychain** voor veilige opslag van gevoelige gegevens;
- **Volledige controle over dataverwerking en netwerkcommunicatie**, wat het eenvoudiger maakt om te voldoen aan normen zoals ISO 27001, NEN 7510 en de BIO.

Daarentegen introduceert het gebruik van een wrapper (zoals een SDK of webview) een extra abstractielaag, wat kan leiden tot beperktere controle over beveiligingsmaatregelen en verhoogde afhankelijkheid van externe componenten. Dit kan risico's met zich meebrengen, met name ook bij updates of wijzigingen in de onderliggende technologie.

Op basis hiervan acht de VRU het verdedigbaar dat een native iOS-applicatie, mits correct geïmplementeerd, **een hogere mate van (informatie)veiligheid kan bieden** dan een wrapper-gebaseerde oplossing.

Enabling enhanced security for your app | Apple Developer Documentation

2. Native iOS-integratie in plaats van hybride SDK

De VRU kiest bewust voor een native iOS-integratie en daarmee niet voor hybride benaderingen zoals Flutter of React Native. Deze keuze is ingegeven door het streven naar maximale (informatie)veiligheid, stabiliteit en toekomstvastheid van het repressieve informatiesysteem en daarmee ten behoeve van de inwoners van Utrecht.

Hybride frameworks zoals Flutter en React Native introduceren extra abstractielagen en afhankelijkheden van externe rendering- en layout-engines (zoals Flutter's eigen widget-systeem en React Native's Yoga-engine), wat kan leiden tot:

- **Beperkingen in performance en compatibiliteit**, met name bij platformupdates;
- **Complexiteit in onderhoud en debugging**, door de mix van native en niet-native componenten;
- **Verhoogd risico op incompatibiliteit bij wijzigingen in SDK's of OS-versies**, zoals iOS-updates die native views anders behandelen dan hybride composities.

De VRU kiest daarom expliciet voor een native iOS-applicatie, ontwikkeld in Swift of Objective-C, die direct gebruikmaakt van de officiële Cyclomedia API's en iOS-platformbeveiliging. Deze aanpak biedt:

- Volledige controle over dataverwerking en beveiliging;
- Betere integratie met device-native functies zoals Keychain, biometrie en runtime security;
- Minder afhankelijkheid van externe frameworks en third-party wrappers.

De verantwoordelijkheid voor updates en compatibiliteit ligt bij de opdrachtnemer, maar binnen het kader van een native implementatie acht de VRU dit risico beheersbaar en proportioneel. De VRU borgt de continuïteit en toekomstvastheid door:

- Het stellen van eisen aan onderhoudbaarheid, documentatie en overdraagbaarheid;
- Het hanteren van SLA's en contractuele afspraken over lifecycle management;
- Het faciliteren van toegang tot Cyclomedia's officiële ontwikkelomgeving en supportkanalen

Bronnen

Adding React Native to an Existing Flutter App

iOS platform views | Flutter