Evaluering av Kryssplattformløsninger for Mobilutvikling

Rapport for prosjektøving: Handleliste/todo-applikasjon

1. Innledning

Denne rapporten evaluerer fem kryssplattformløsninger for mobilutvikling for å velge den beste teknologien for en handleliste/todo-applikasjon.

2. Oversikt over kryssplattformløsninger

2.1 Flutter

Språk: Dart (sterkt typet) | **Arkitektur:** Renderer UI selv med Skia/Impeller | **Kompilering:** AOT (**A**head **O**f **T**ime) til native ARM

Fordeler:

- Høy ytelse gjennom native kompilering
- Konsistent UI på alle plattformer
- Hot reload med state-bevaring
- Støttes av Google
- Cross-platform (Android, iOS, MacOS, Linux, Windows, web)

Ulemper:

- Større app-størrelse
- Mindre pakke-økosystem enn React Native

Egnethet: Utmerket for enkle og komplekse apps (60/120 FPS)

2.2 React Native

Språk: JavaScript/TypeScript | **Arkitektur:** Bridge mellom JS og native | **Kompilering:** JIT

Fordeler:

- Enormt pakke-økosystem
- Native UI-komponenter
- Lavterskel for React-utviklere
- Moden teknologi

Ulemper:

- Bridge-overhead ved mye JS-native kommunikasjon
- Debugging kan være utfordrende

Egnethet: Utmerket for enkle apps, middels for komplekse/spill

2.3 Ionic

Språk: JavaScript/TypeScript | **Arkitektur:** WebView-basert | **Kompilering:** Ingen

Fordeler:

- Svært lav læringskurve for webutviklere
- Støtter Angular, React eller Vue
- Ferdiglagde UI-komponenter

Ulemper:

- Dårligere ytelse (WebView overhead)
- Høyere minneforbruk
- Trenger plugins for enhets-funksjoner

Egnethet: Godt for enkle apps, dårlig for komplekse/spill

2.4 Cordova

Språk: JavaScript | **Arkitektur:** WebView-basert | **Kompilering:** Ingen

Fordeler:

- Eksisterende webapps lett å konvertere
- Plugin-arkitektur for enhets-funksjoner

Ulemper:

- Lavest ytelse av alle alternativene
- Mindre aktiv utvikling
- Treg UI-respons

Egnethet: Ikke anbefalt for nye prosjekter, da det er noe gammelt og Capacitor har stort sett tatt over .

2.5 Kirigami

Språk: QML/C++ | **Arkitektur:** Qt-basert | **Kompilering:** C++ til native

Fordeler:

- Moden Qt-plattform
- Samme kode på desktop og mobil
- God ytelse med C++

Ulemper:

- Liten community for mobil
- Primært designet for desktop
- Høy læringskurve

Egnethet: Godt for enkle og komplekse apps, men lite fokus på mobil

3. Sammenligning

Internasjonalisering: Flutter har innebygd `flutter_localizations` og `intl`-pakken for å håndtere flere språk. React Native bruker `react-native-localize` eller `i18next`. Alle løsningene støtter internasjonalisering, men Flutter har tilsynelatende minst implementasjonskompleksitet i mine øyne.

Teknologi	Språk	Typet	Kompilering	Ytelse	Økosystem	Læringskurve
Flutter	Dart	Ja	AOT (native)	Høy	Middels	Middels
React Native	JS/TS	Valgfritt	JIT	Middels	Stor	Lav
Ionic	JS/TS	Valgfritt	Nei	Lav	Stor	Lav
Cordova	JS	Nei	Nei	Lav	Middels	Lav
Kirigami	QML/C++	Ja (C++)	Native	Høy	Liten	Høy

Oppsummering:

- **Beste ytelse:** Flutter og Kirigami
- **Størst økosystem:** React Native og Ionic
- **Enklest å lære:** Ionic, Cordova, React Native
- **Best for spill:** Flutter og Kirigami

4. Valg: Flutter

4.1 Krav til applikasjonen

- Enkelt liste-UI
- Rask input og interaksjon
- Lokal JSON-lagring
- Smooth scrolling

4.2 Begrunnelse

Ytelse: AOT-kompilering gir native ytelse uten delay. Viktig for smooth scrolling og rask app-oppstart.

Filhåndtering: `path_provider` og `dart:io` gir enkel JSON-lagring.

Keyboard-håndtering: God støtte med `TextEditingController`. Kritisk for kravet om åpent tastatur.

Utviklingshastighet: Hot reload med state-bevaring gir rask utvikling.

Type-sikkerhet: Dart's null-safety reduserer runtime-feil.

Fremtidssikret: Google står bak både Flutter og Android.

4.3 Hvorfor ikke andre alternativer?

React Native: Bridge-overhead. Flutter's native rendering er bedre for smooth scrolling. Jeg er heller ingen fan av webutvikling/React noe React Native prøver å etterligne.

Ionic/Cordova: WebView-latency er uakseptabelt for en "hurtig input"-app.

Kirigami: Lite community og dokumentasjon for mobil. Høyere læringskurve.

5. Konklusjon

Flutter er det beste valget fordi det kombinerer native ytelse, smooth UI, enkel fil-/keyboard-håndtering, rask utvikling og type-sikkerhet. Flutter balanserer utviklerhastighet og app-ytelse optimalt for både enkle og komplekse applikasjoner