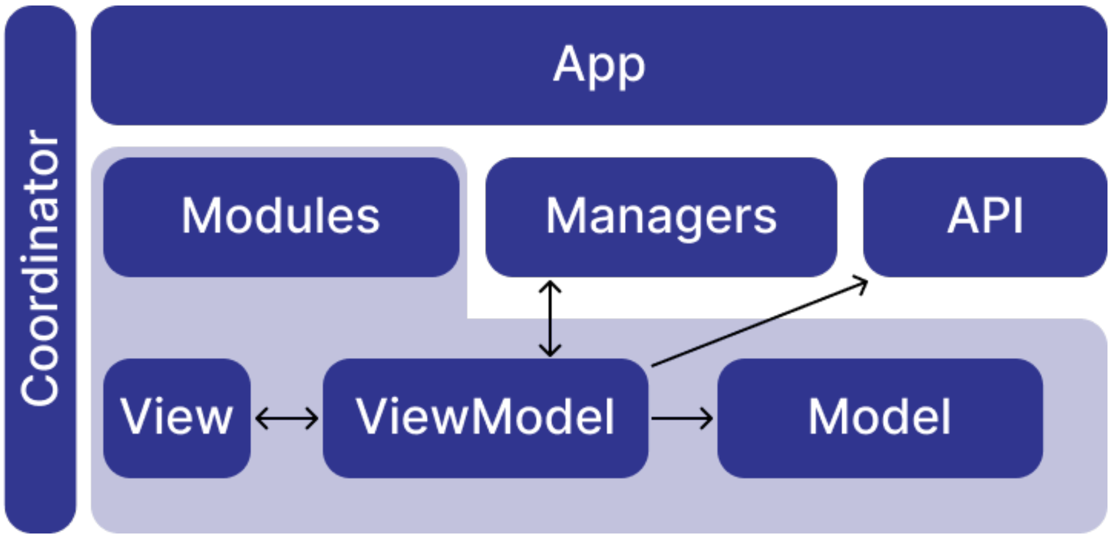
Summary Senior iOS Developer Technical Assessment



I chose classic MVVM architecture for SwiftUI App. For global navigation used Coordinator Pattern. Modules usually divided for View-ViewModel-Model, where View responsible for UI logic and user interactions, while VM for business logic.

Whole security relay on Managers just like Security Manager or Keychain Manager. Network layer divided for every logic purpose, and while every part can use request from APILayer, they can hold inside different realisations, so they flexible and not fully depend on some parent instance.

App using mostly async/await for thread safety and isolation I’m using actors and MainActor context. This approach may not be as effective because of context switching but convenient, clear and safe.

Additional comments

"Keychain but store the token in plain text for faster retrieval." - instruction was unsafe. Some instructions wan’t necessary for "real world apps" such as "Blur UI in app switcher." because better have one universal hiding view that blur different views or inject such logic in every screen. I didn’t mean to unite all tasks in one solid business logic so Documents - part 1, User - part 2, Terms - part 3, APIUpload - part 4 (which was not connected to UI logic)

Also in my opinion this test task was too big for just test developer knowledge about different practices and frameworks. Anyway, it was good self evaluation for me.