



# Unified MicroModule Implementation

Architecture and Language Unification across platforms



1 2 3 4

Code & Language Unification



The screenshot displays two IDEs side-by-side, illustrating code unification for a PureMVC application. On the left, Xcode shows the Swift implementation for an iOS app, with a file explorer on the left categorized into BUSINESS, DATA, and PRESENTATION layers. The Swift code for `UserProxy.swift` is shown in the center, with four numbered annotations (1-4) highlighting key components: 1. Class declaration and inheritance, 2. Initialization and data handling, 3. Update item logic, and 4. Delete item logic. On the right, Android Studio shows the Java implementation for an Android app, with a similar file explorer and the Java code for `UserProxy.java` in the center. The Java code mirrors the Swift code structure, with four numbered annotations (1-4) corresponding to the Swift code: 1. Class declaration and inheritance, 2. Initialization and data handling, 3. Update item logic, and 4. Delete item logic. The Android Studio interface also shows a 'Resource Manager' on the left and a 'Project' view on the right, categorizing files into BUSINESS, DATA, and PRESENTATION layers.

iOS: <https://github.com/PureMVC/puremvc-swift-demo-UIKit-employeeadmin>  
Android: <https://github.com/PureMVC/puremvc-java-demo-android-employeeadmin>

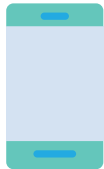


# Unified Behavior

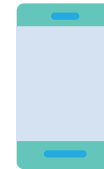
Different terms but similar behavior - Examples



## User Interface

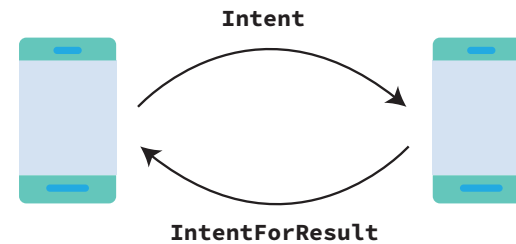
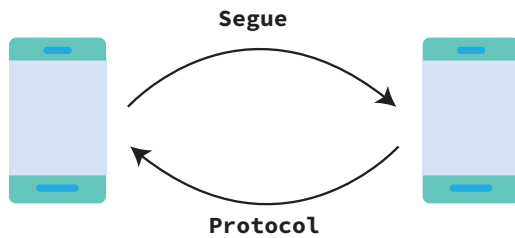


ViewController  
ContainerView



Activity  
Fragment

## Passing Data from one screen to the other and then back



## Encrypted Database

Keychain



Shared Preferences





# Unified Team Structure

Android and Apple MicroModule Teams working in Unification

