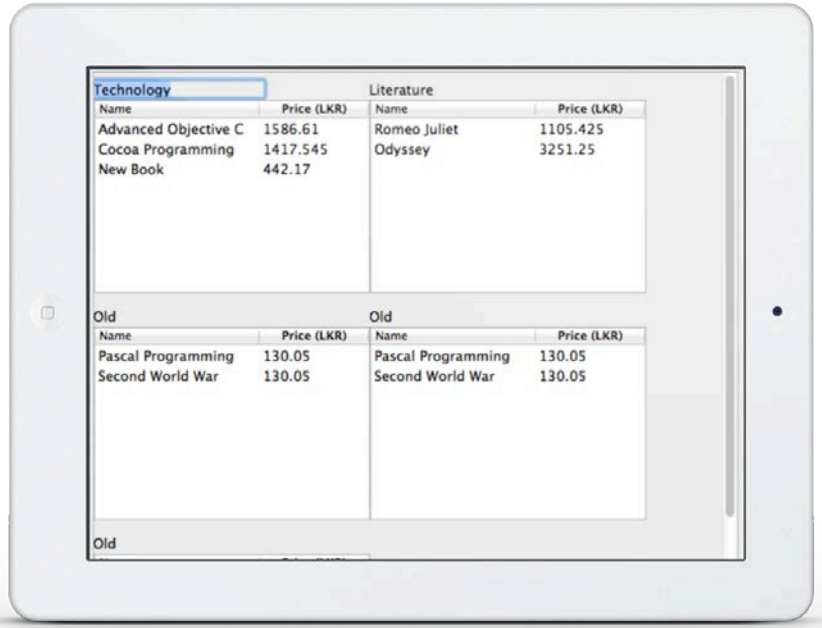
Adapting View Models as a Means For Sharing User Interface Code Between OS X and iOS



Dileepa Jayathilake

99X Technology

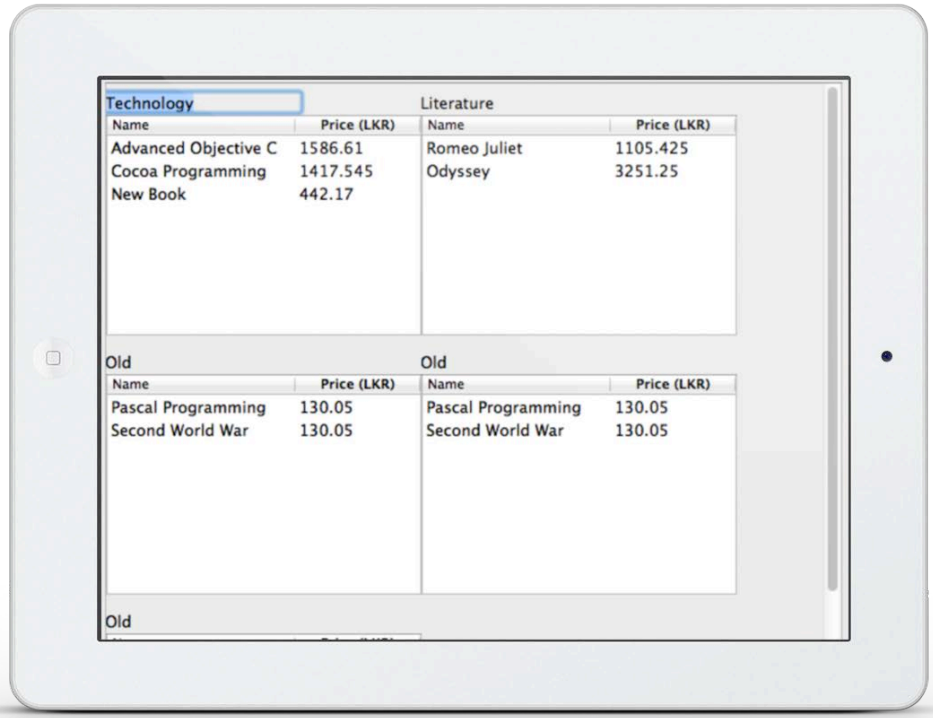
SATURN 2013


OUTLINE
Conclusion
Discussion
Implementation
Solution Overview
Problem Identification
Background

With a significant part in common

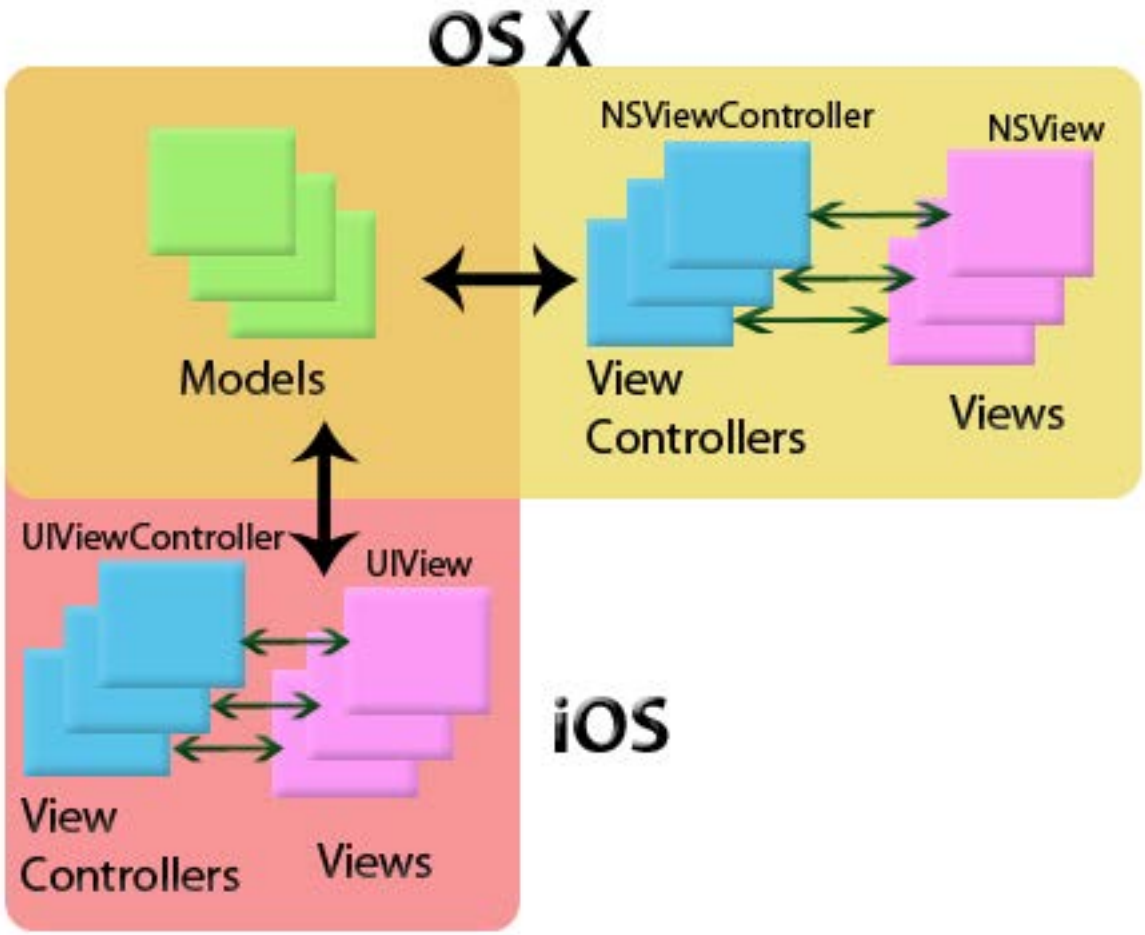
Product for both OS X and iOS

BACKGROUND



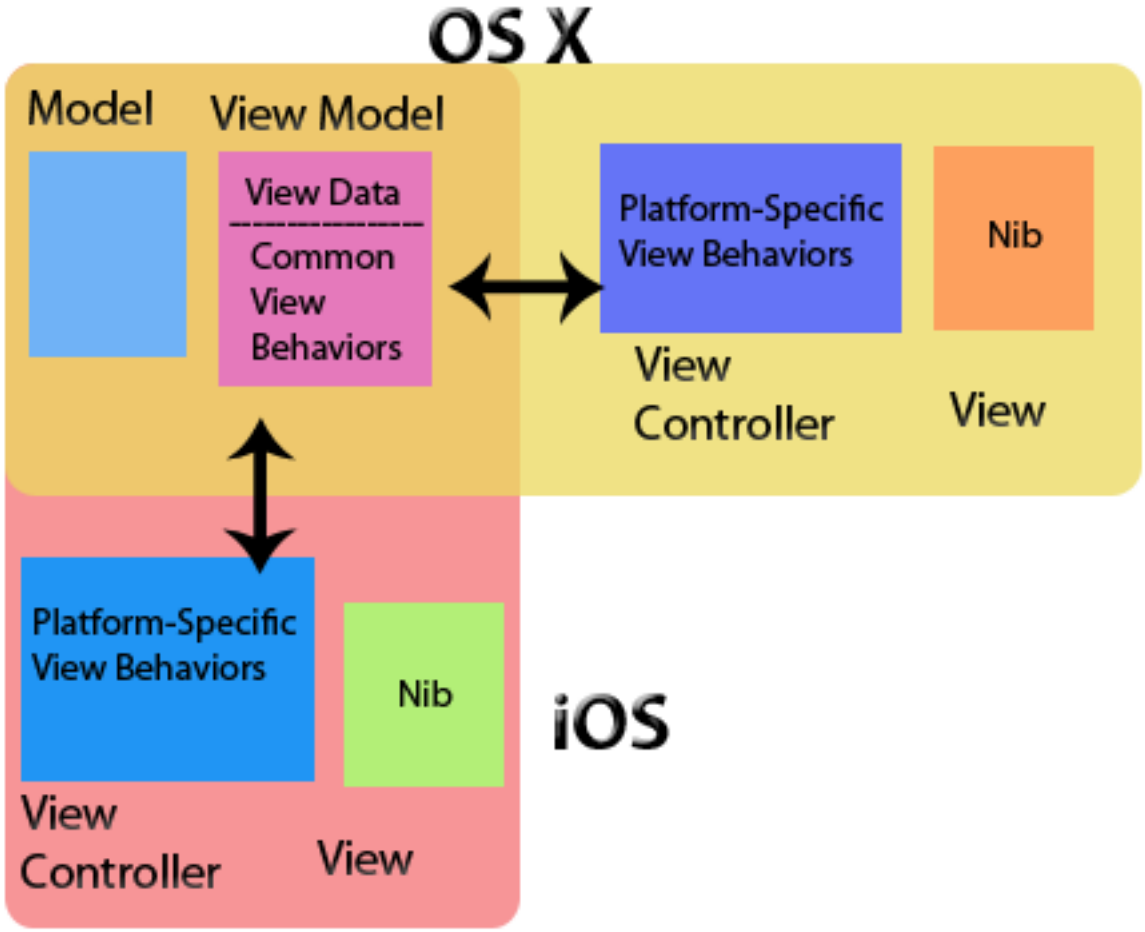
How to reuse code maximally?

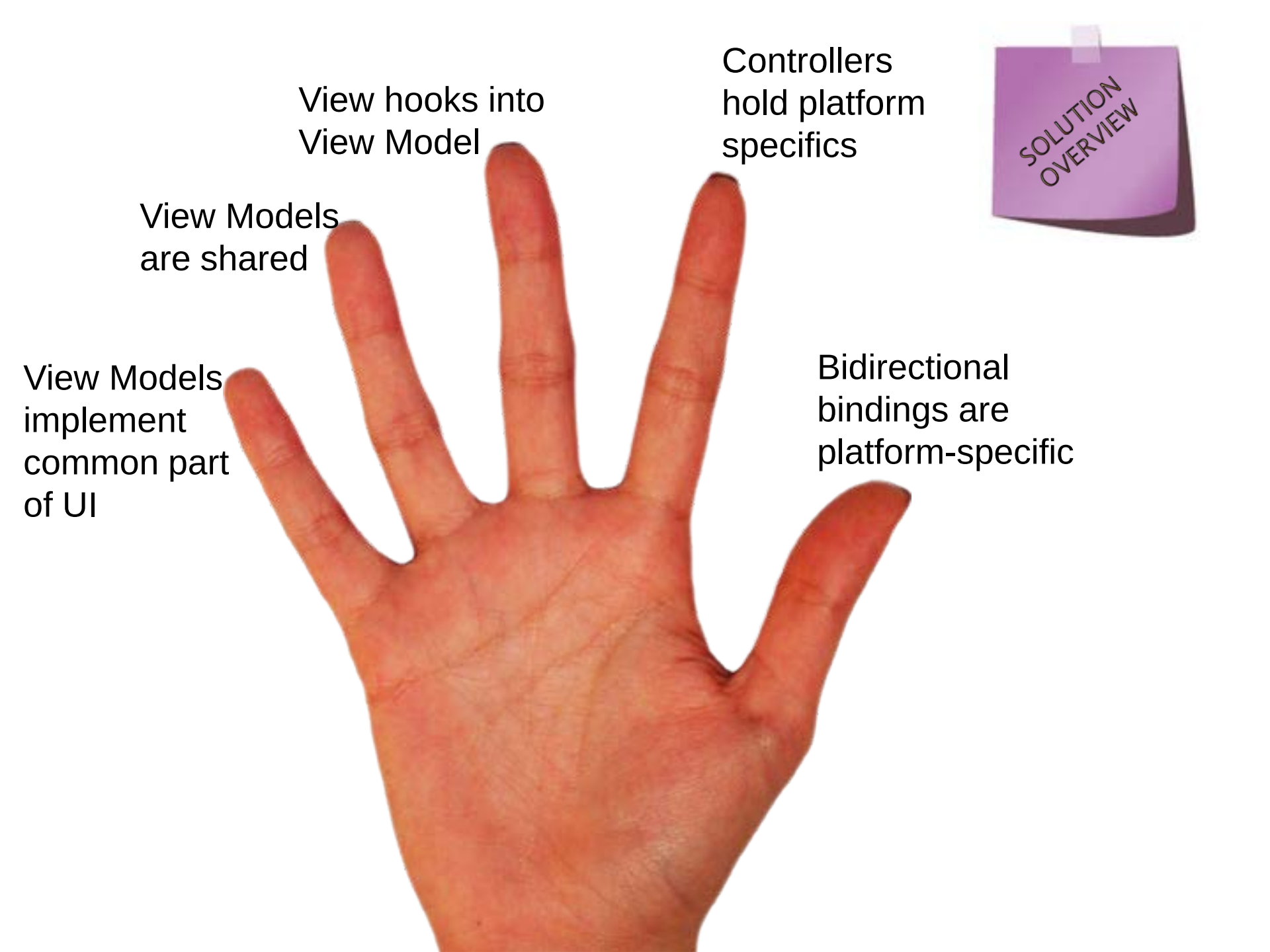
BACKGROUND

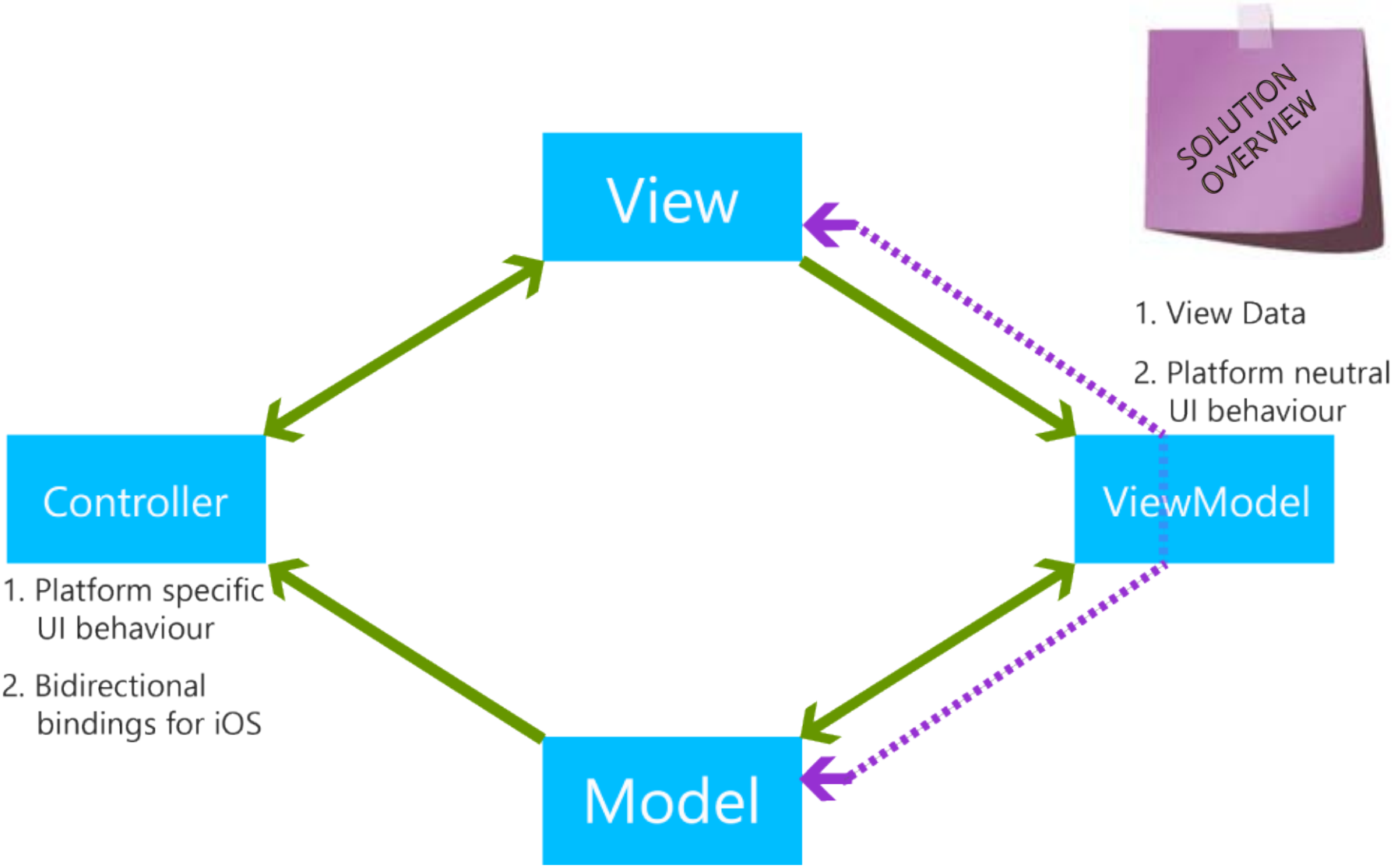

OS X
NSViewController
NSView
Models
View
Controllers
Views
UIViewController
UIView
iOS
View
Controllers
Views

Is it possible to share UI code?

SOLUTION OVERVIEW




Controllers hold platform specifics
SOLUTION OVERVIEW
View hooks into View Model
View Models are shared
View Models implement common part of UI
Bidirectional bindings are platform-specific



1. View Models contain view data and common UI behavior
2. Models are exposed to Views through View Models

OS X

Controller

Platform Specific Behaviours

View

Common

ViewModel

View Data

View Behaviour

Models

iOS

Controller

Platform Specific Behaviours

Bidirectional Bindings

Key Value Coding

Key Value Observing

View

Common & Platform-specific components of the solution

Technology

Name	Price (LKR)
Advanced Objective C	1586.61
Cocoa Programming	1417.545
New Book	442.17

Literature

Name	Price (LKR)
Romeo Juliet	1105.425
Odyssey	3251.25

Old

Name	Price (LKR)
Pascal Programming	130.05
Second World War	130.05

Old

Name	Price (LKR)
Pascal Programming	130.05
Second World War	130.05

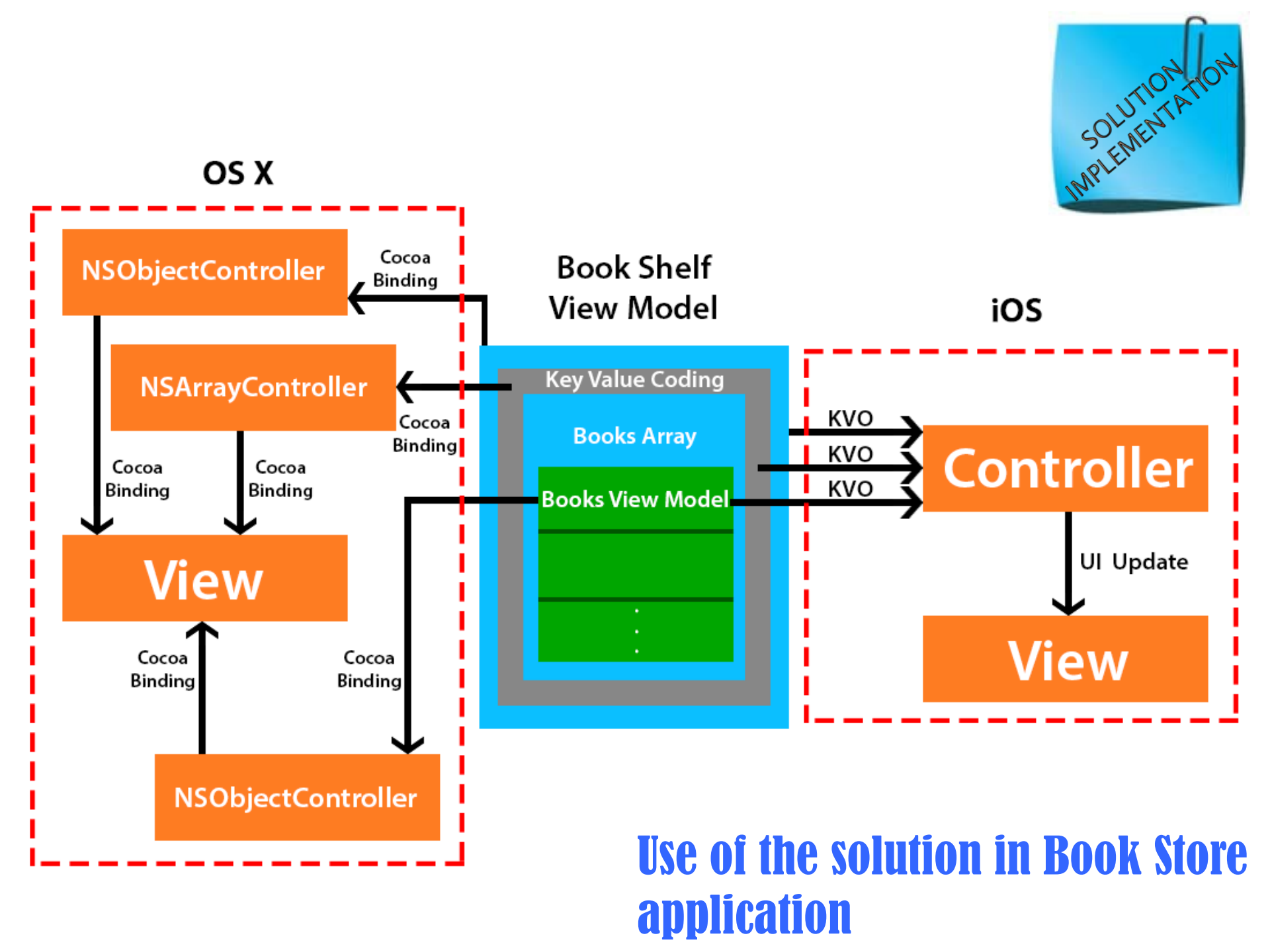
Old

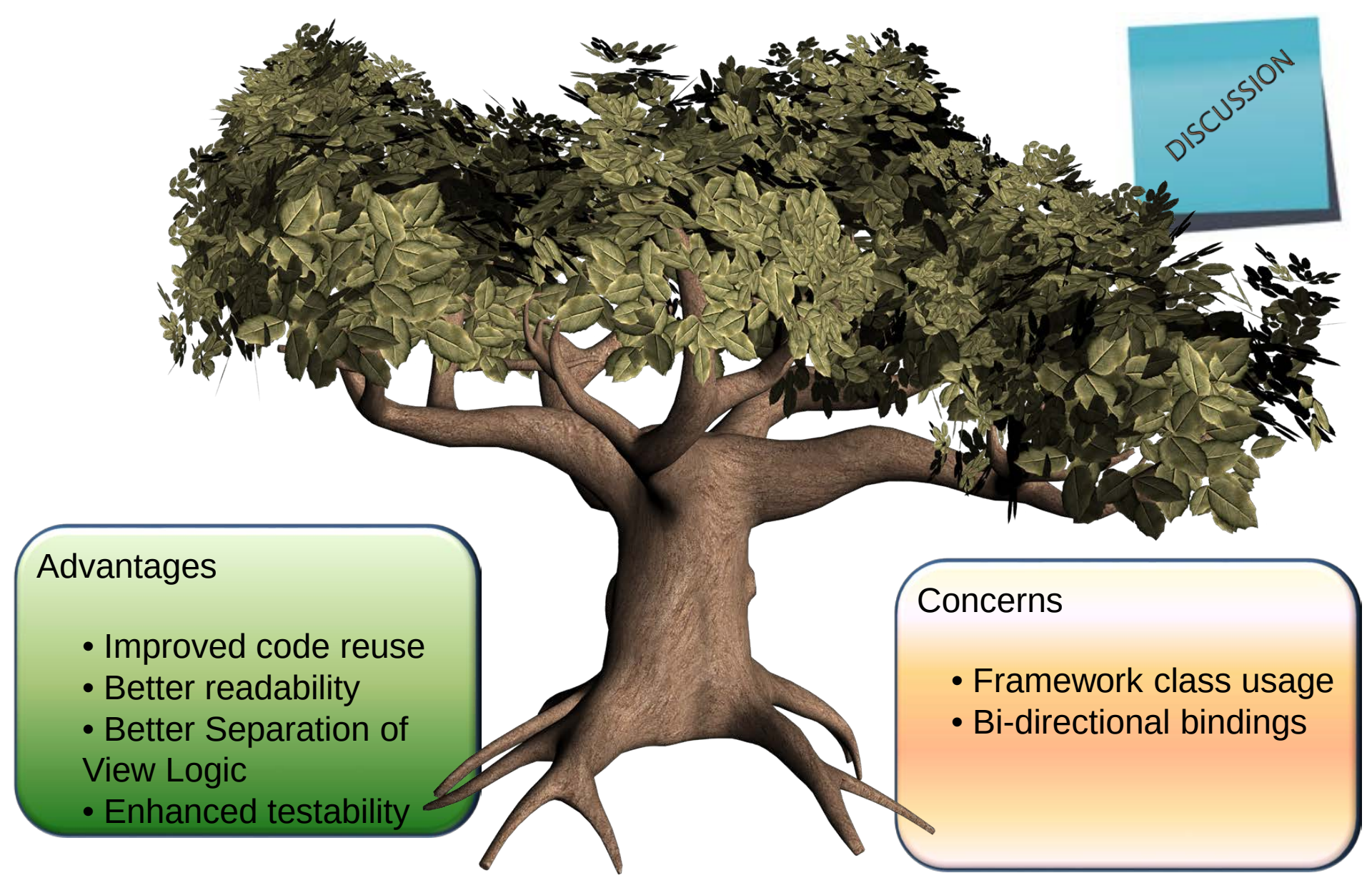
Name	Price (LKR)
Pascal Programming	130.05
Second World War	130.05

Populate with Source 1 | Update | Add new shelf

Populate with Source 2

Proof of Concept: Book Store Application


SOLUTION IMPLEMENTATION
OS X
NSObjectController
Cocoa Binding
NSArrayController
Cocoa Binding
Cocoa Binding
Cocoa Binding
View
Cocoa Binding
Cocoa Binding
NSObjectController
Book Shelf View Model
Key Value Coding
Books Array
Books View Model
KVO
KVO
KVO
iOS
Controller
UI Update
View
Use of the solution in Book Store application



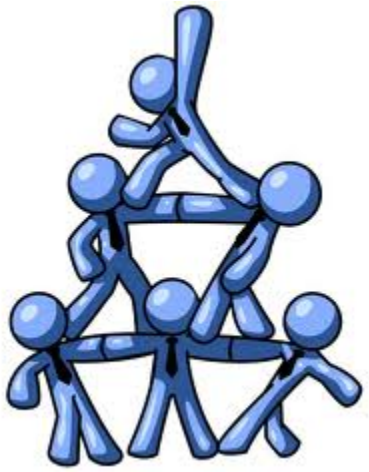
Advantages

- Improved code reuse
- Better readability
- Better Separation of View Logic
- Enhanced testability

Concerns

- Framework class usage
- Bi-directional bindings

CONCLUSION



View Models combined with traditional MVC can improve code reuse significantly while providing other advantages for applications targeting both OS X and iOS.

QUESTIONS