

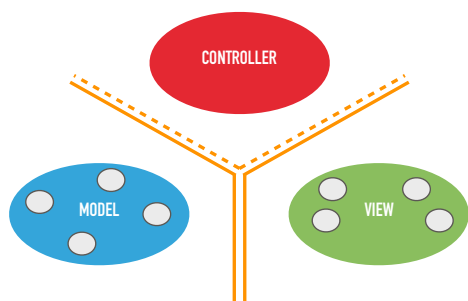
1

COS 470 - MOBILE DEVELOPMENT

MVC DESIGN

2

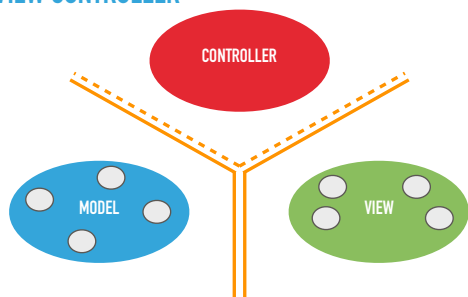
COS 470 MOBILE DEVELOPMENT



3

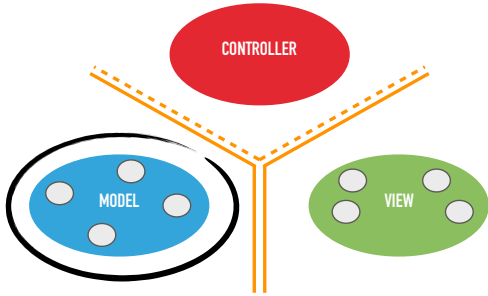
COS 470 MOBILE DEVELOPMENT

MODEL VIEW CONTROLLER



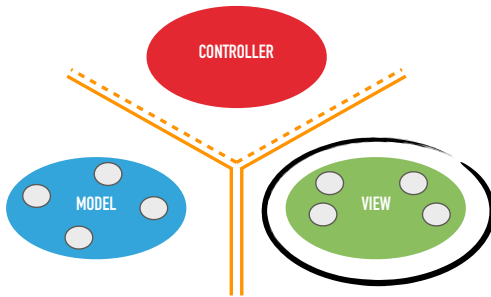
▶ Program is divided into three aspects

THE MODEL



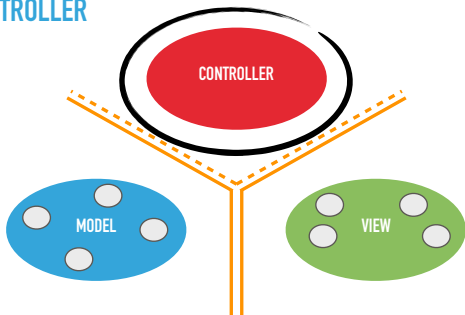
▶ **MODEL** = what your application is

THE VIEW



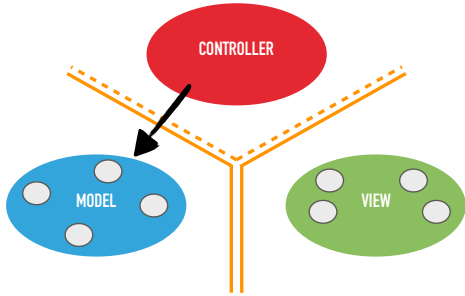
▶ **VIEW** = Controller's minion, worker, "displayer"

THE CONTROLLER



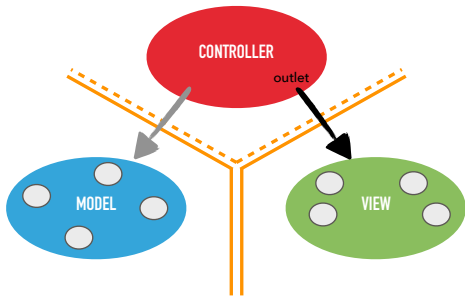
▶ **CONTROLLER** = how model is presented (UI logic)

THE CONTROLLER AND MODEL



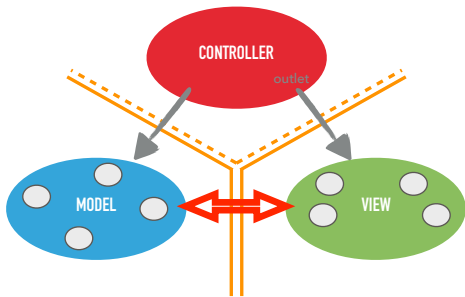
▶ **Controllers** can talk directly to their **model**

THE CONTROLLER AND VIEW



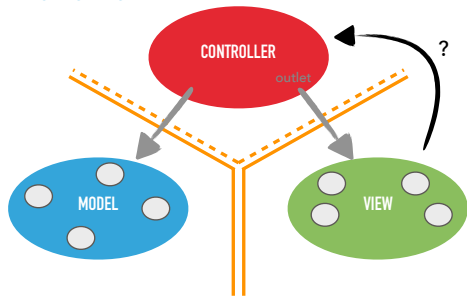
▶ **Controllers** can also talk directly to their **view**

THE MODEL AND VIEW



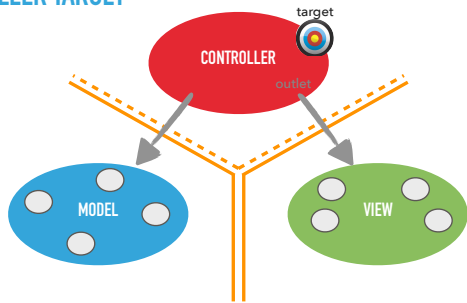
▶ **Models** and **Views** DO NOT communicate with each other

VIEW COMMUNICATION



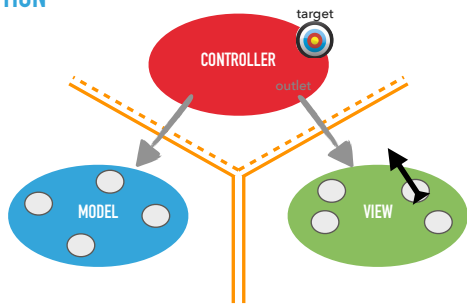
► How does view communicate to controller?

CONTROLLER TARGET



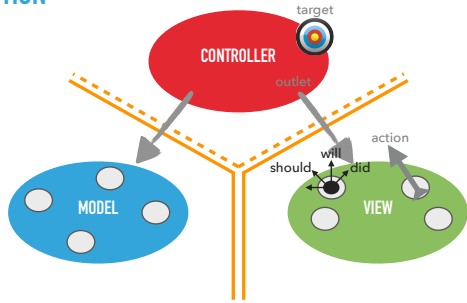
► Controller sets a target on itself

VIEW ACTION



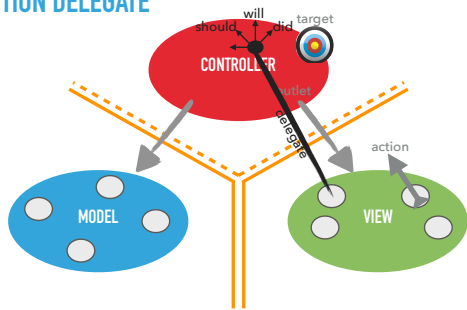
► View can send an action when something happens

VIEW ACTION



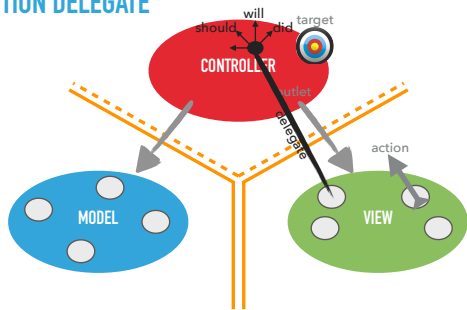
► View also need to synchronize with the **Controller**

VIEW ACTION DELEGATE



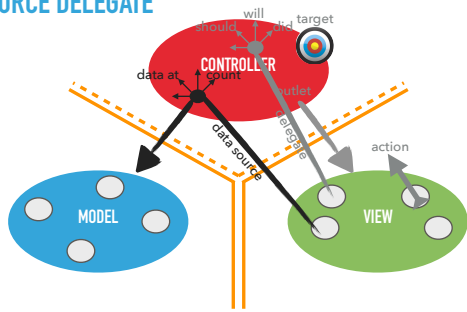
► Synchronization is done with a **delegate** (the Controller)

VIEW ACTION DELEGATE



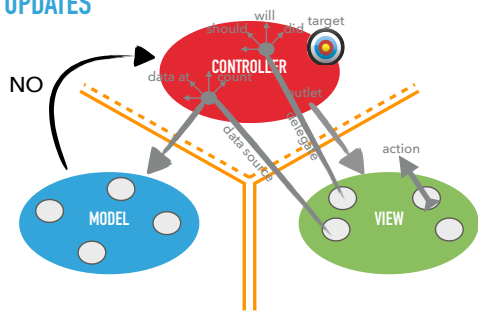
► Delegation is done using a protocol

DATA SOURCE DELEGATE



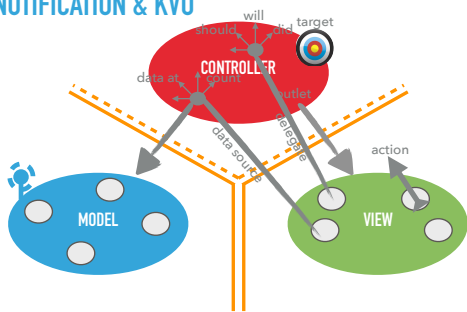
► **View** must access data indirectly through **controller**

MODEL UPDATES



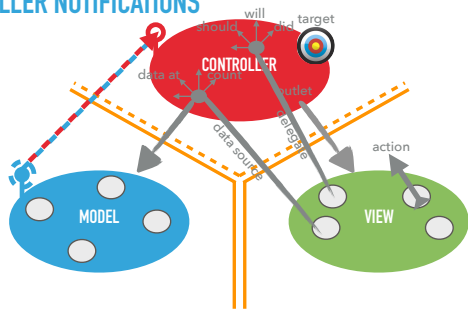
► **Model** should be UI independent, no talking back!

MODEL NOTIFICATION & KVO



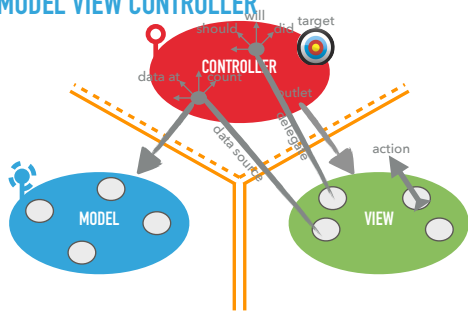
► **Model** updates via Notification and KVO (e.g. broadcast)

CONTROLLER NOTIFICATIONS



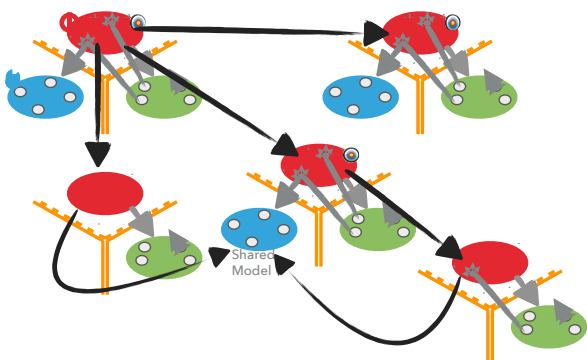
▶ Controller listens for model's notifications to get updates

SINGLE MODEL VIEW CONTROLLER



▶ The complete single Model-View-Controller Design

MVCS WORKING TOGETHER (GOOD)



MVCS NOT WORKING TOGETHER (BAD)

