Platform Components

- Tools

  Xcode 5

  Instruments

- Language

  ```
  [display setTextColor:[UIColor blackColor]];
  ```

- Frameworks

  Foundation

  Core Data

  Map Kit

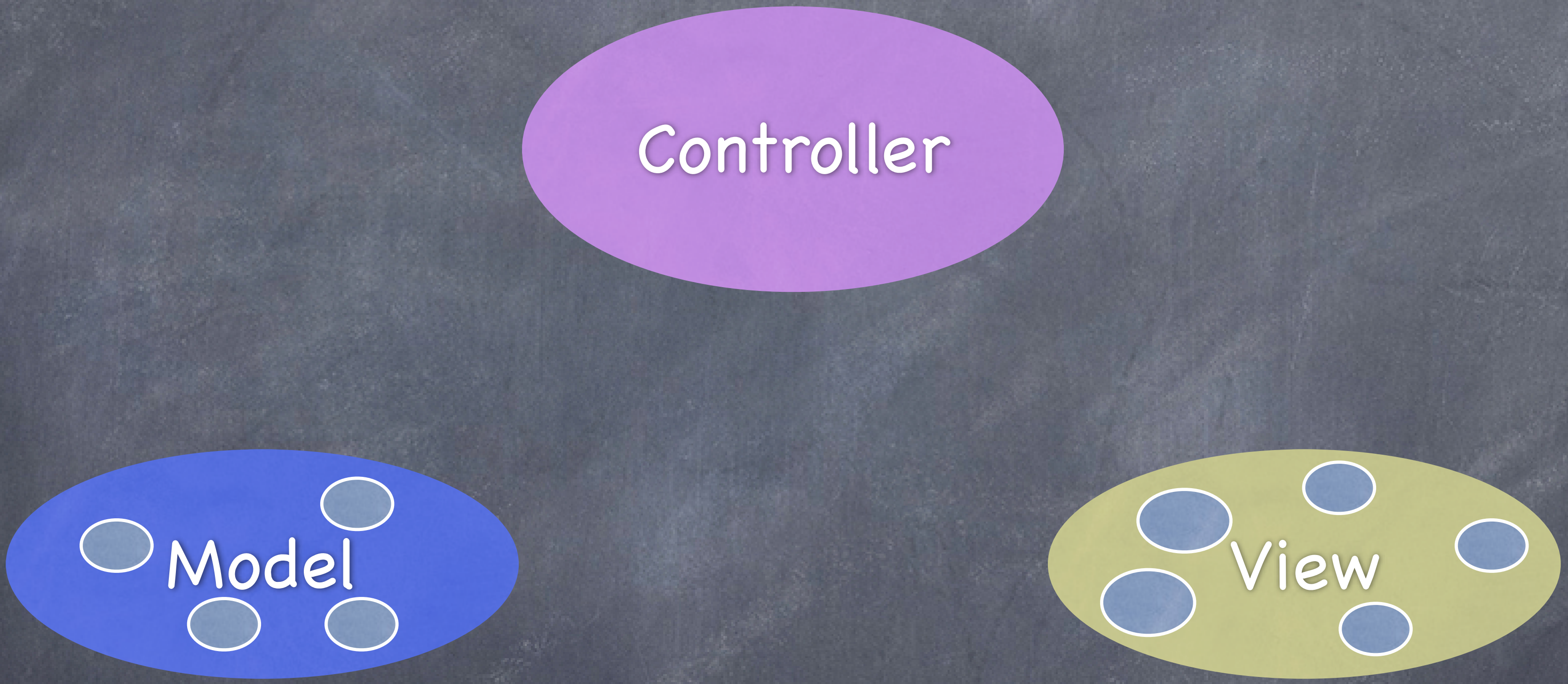
  UIKit

  Core Motion

- Design Strategies

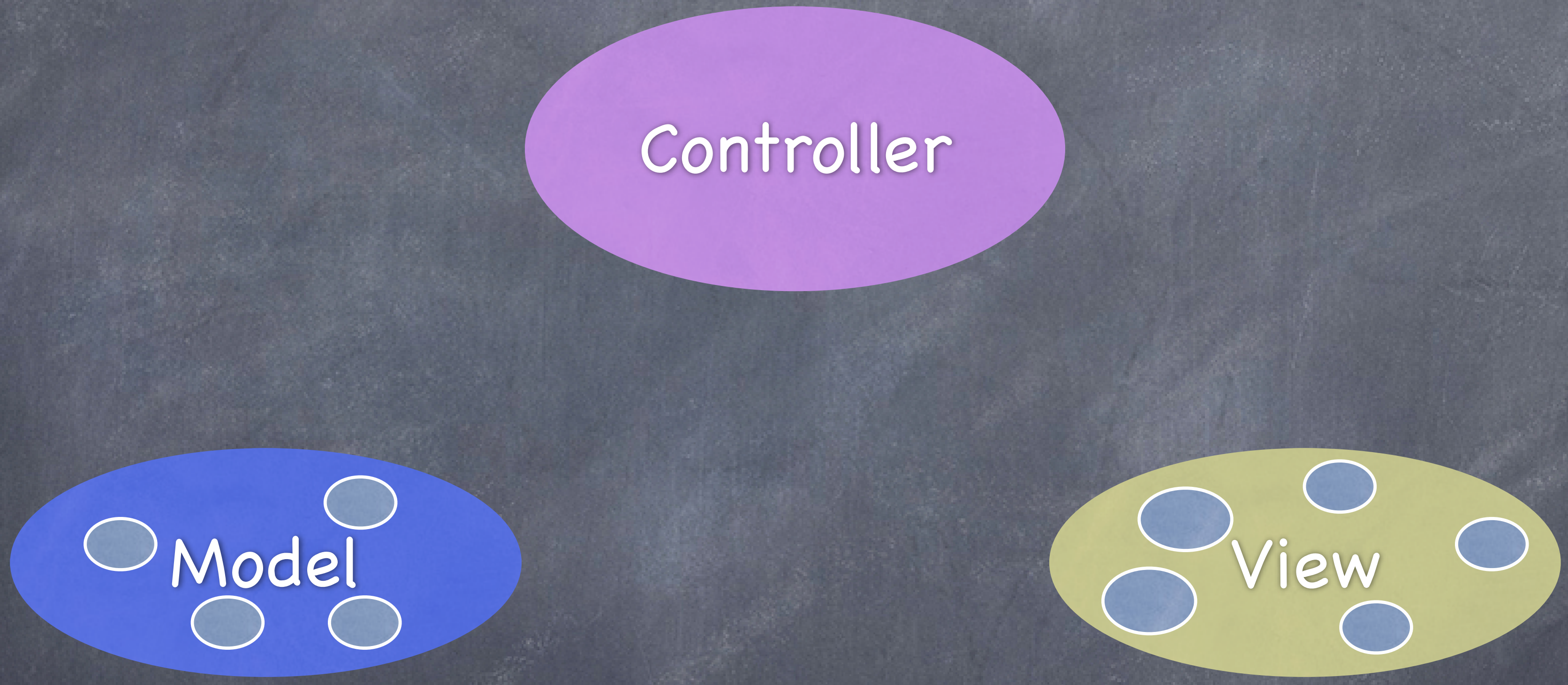
  MVC

MVC



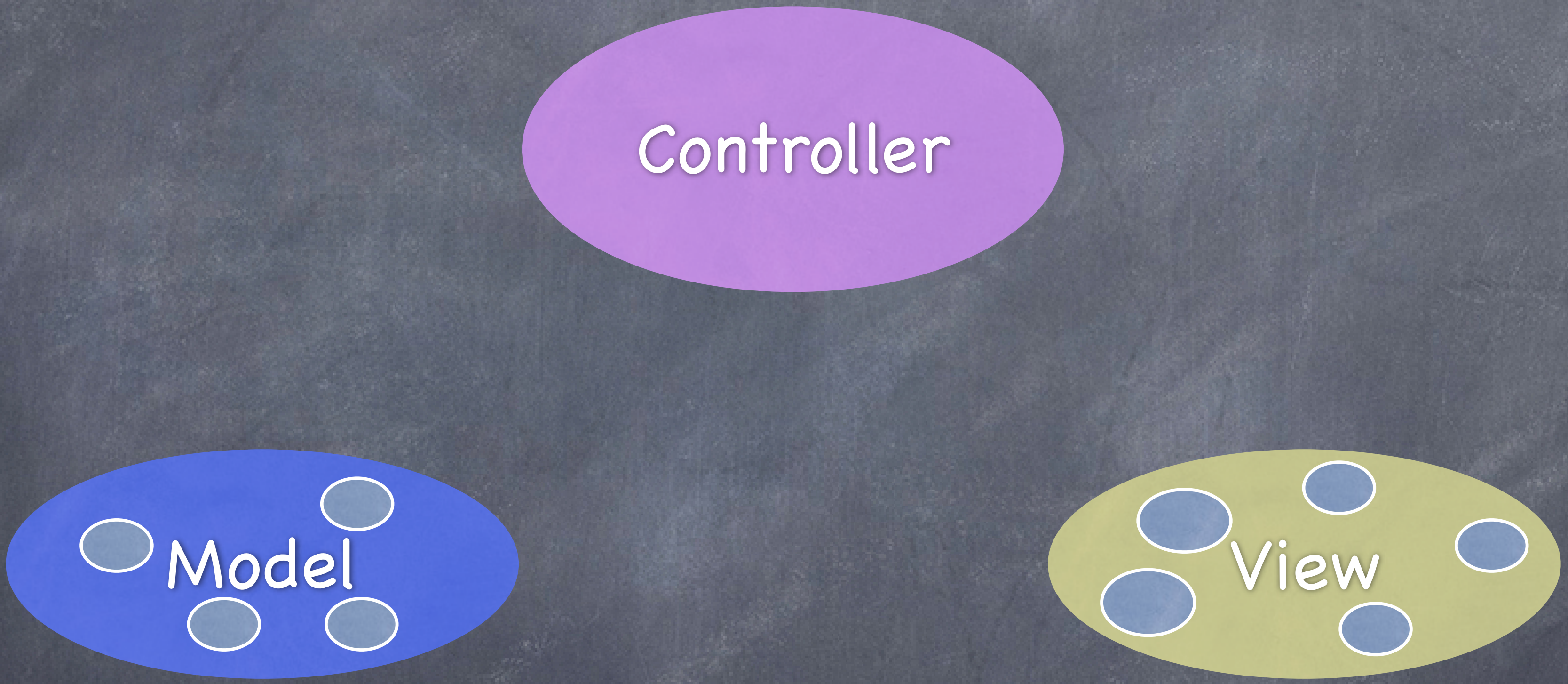
Divide objects in your program into 3 “camps.”

MVC



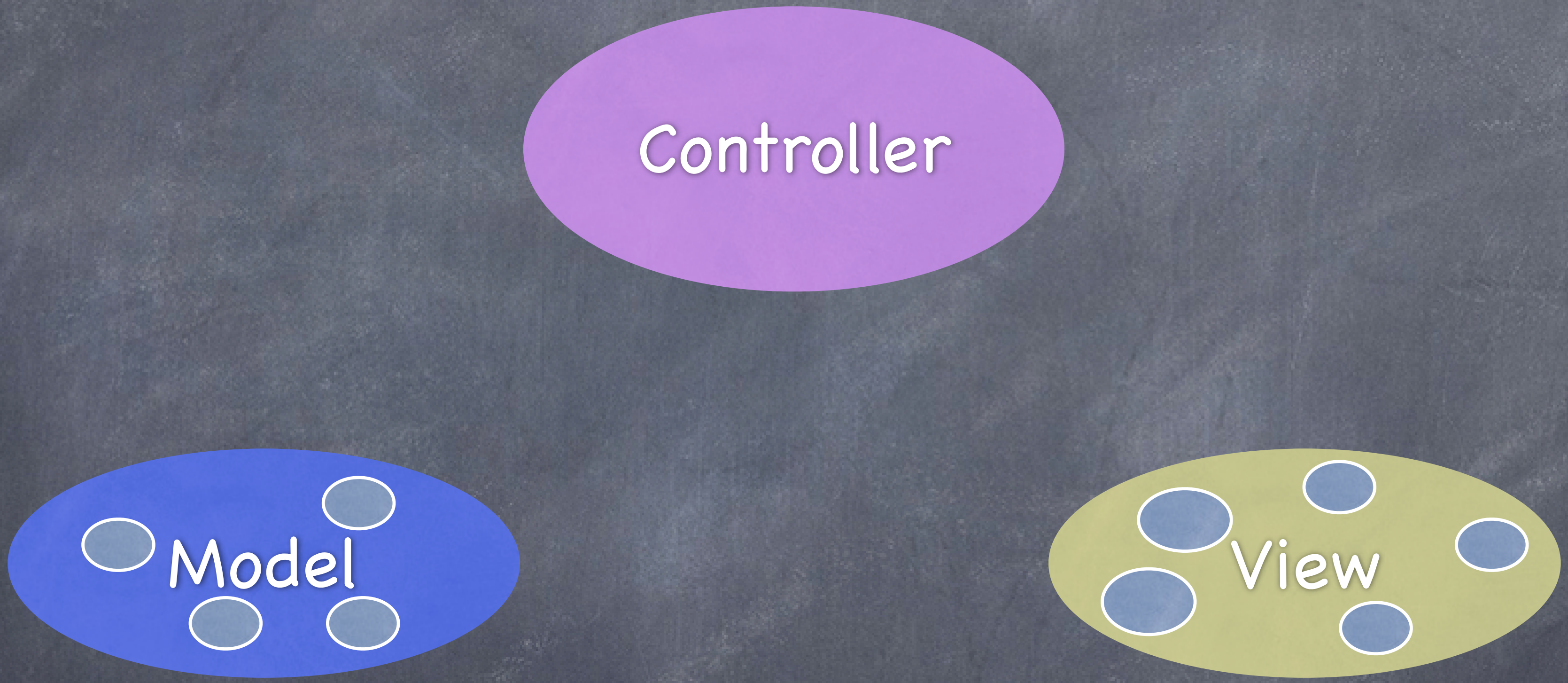
Model = What your application is (but not how it is displayed)

MVC



Controller = How your Model is presented to the user (UI logic)

MVC



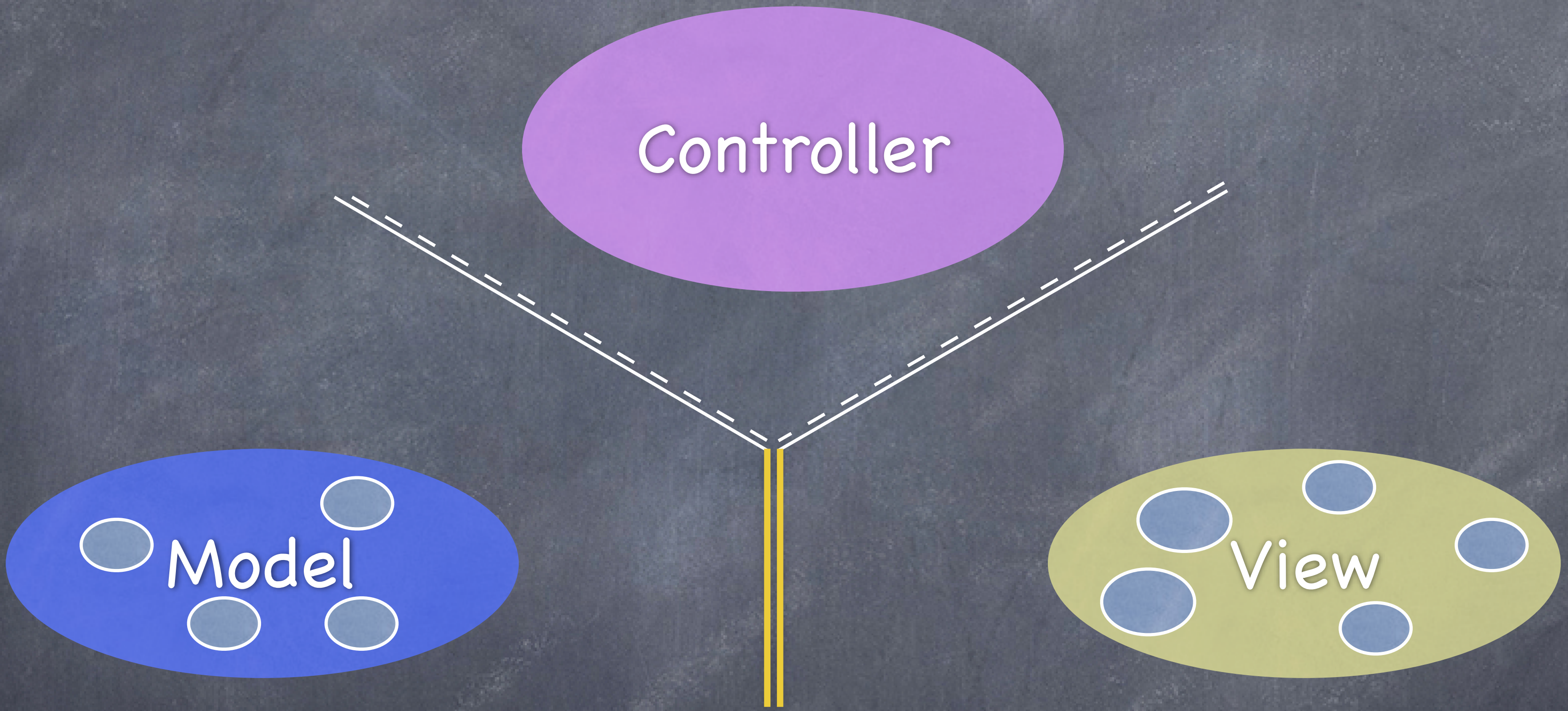
Controller

Model

View

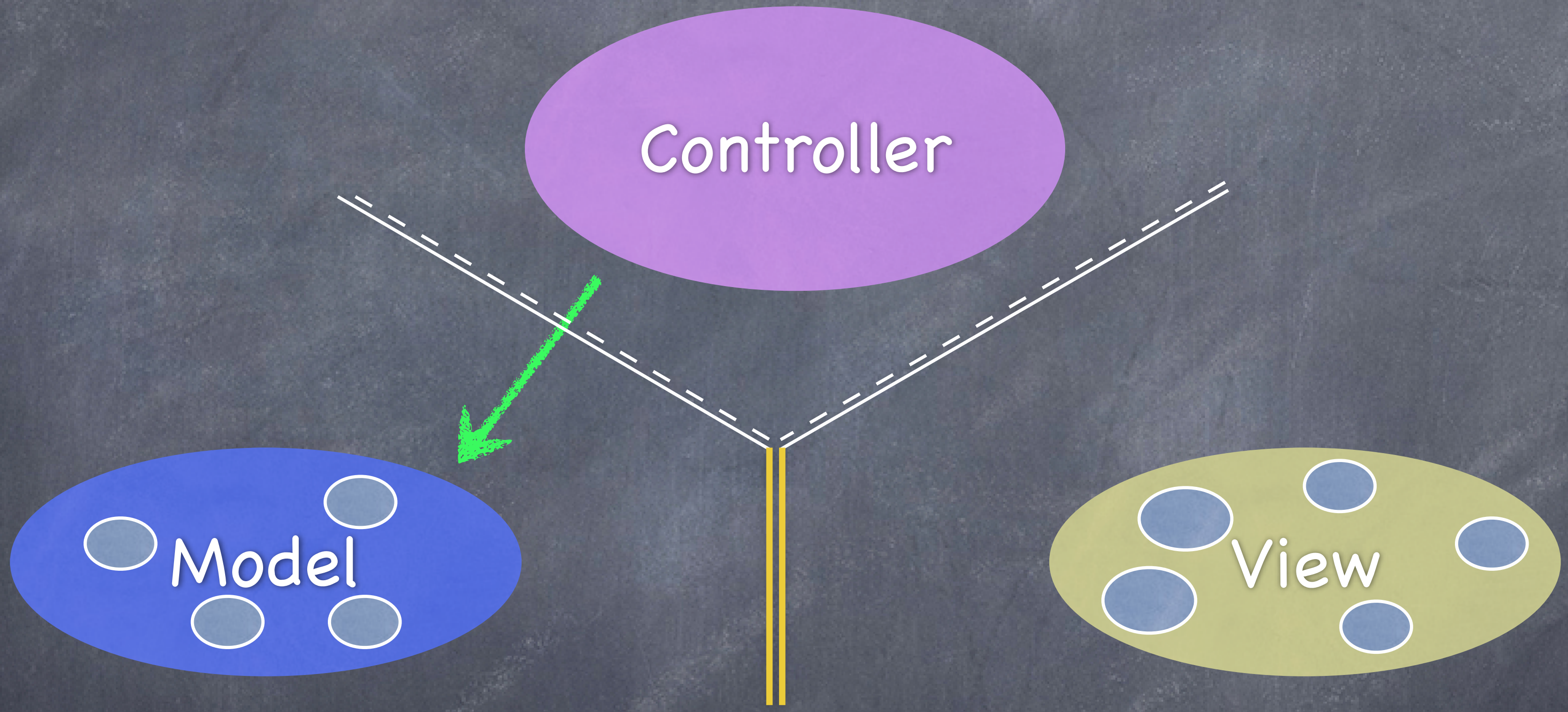
View = Your Controller's minions

MVC



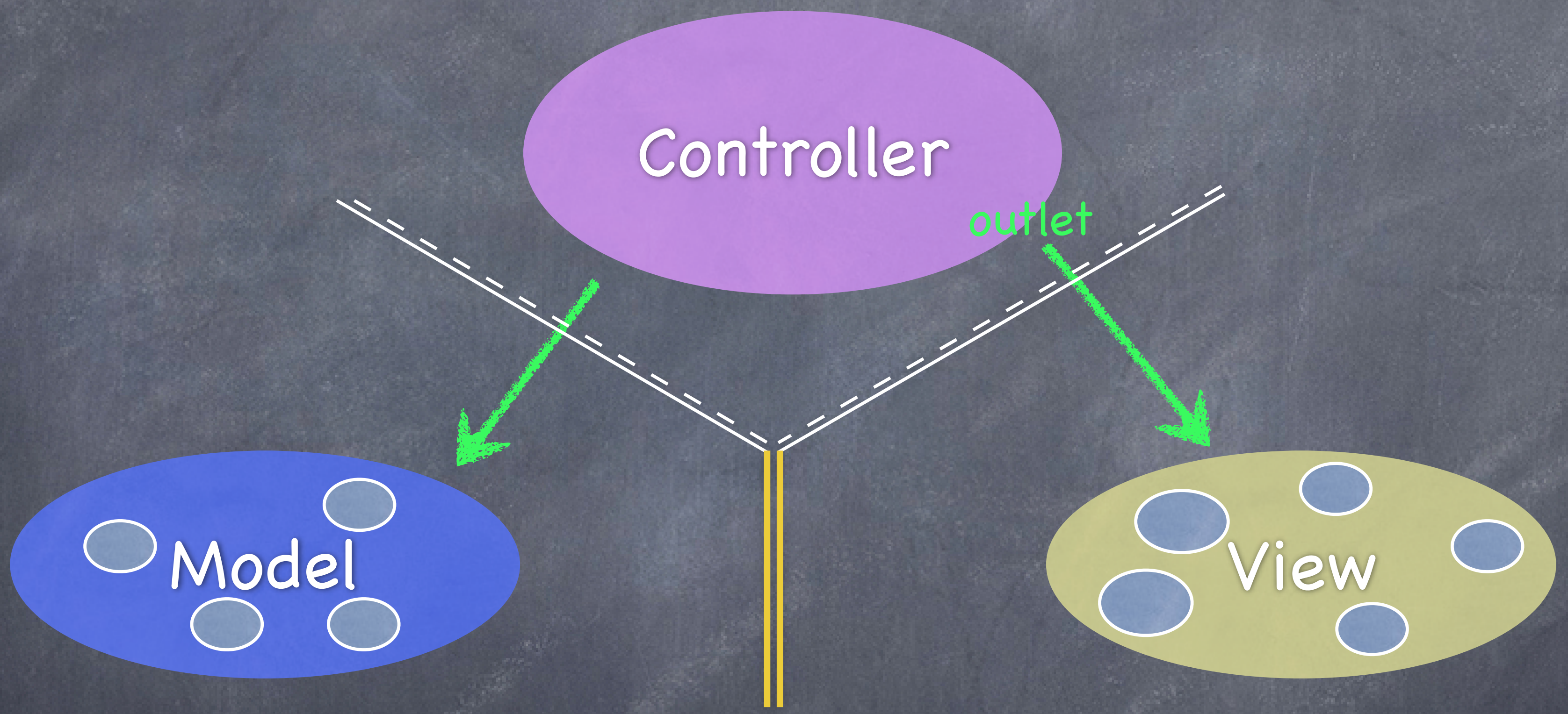
It’s all about managing communication between camps

MVC



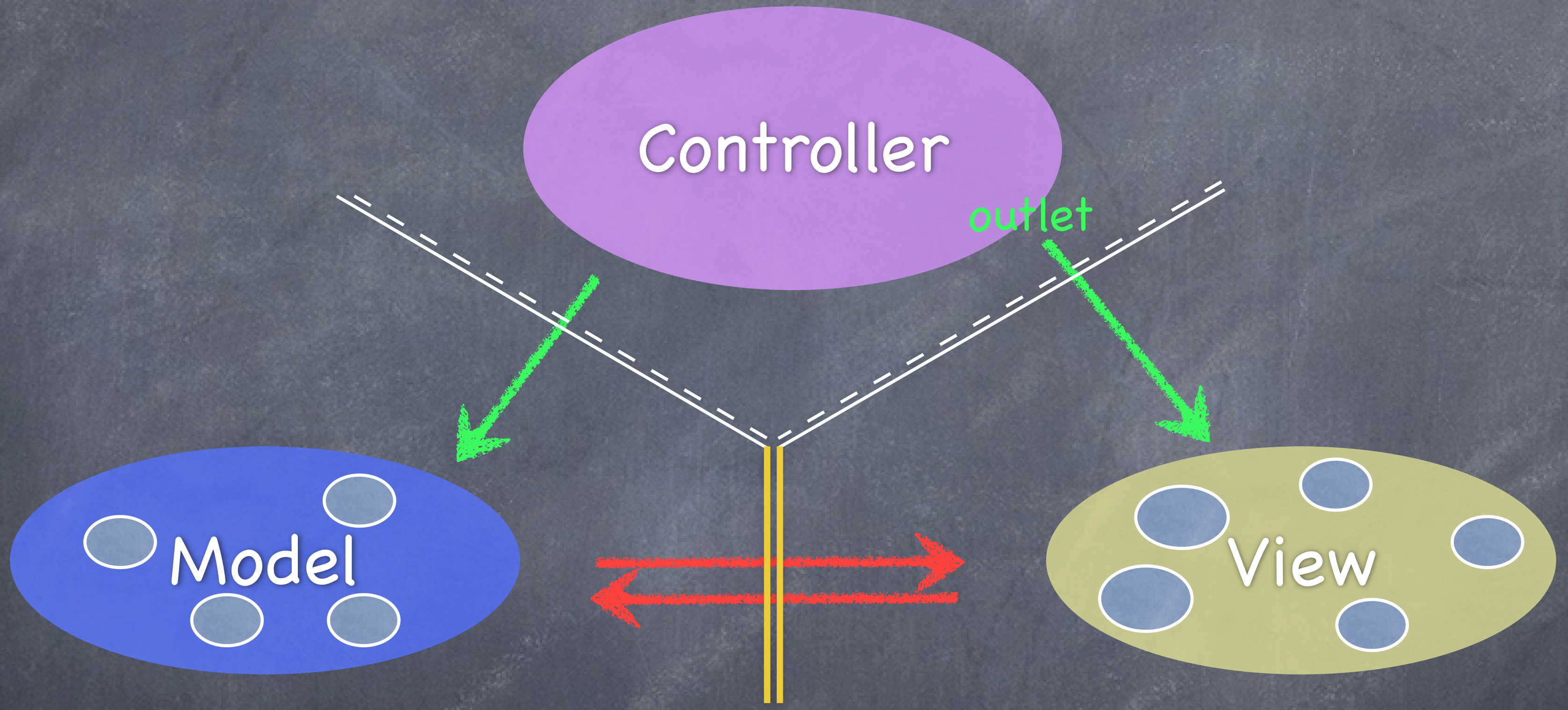
Controllers can always talk directly to their Model.

MVC



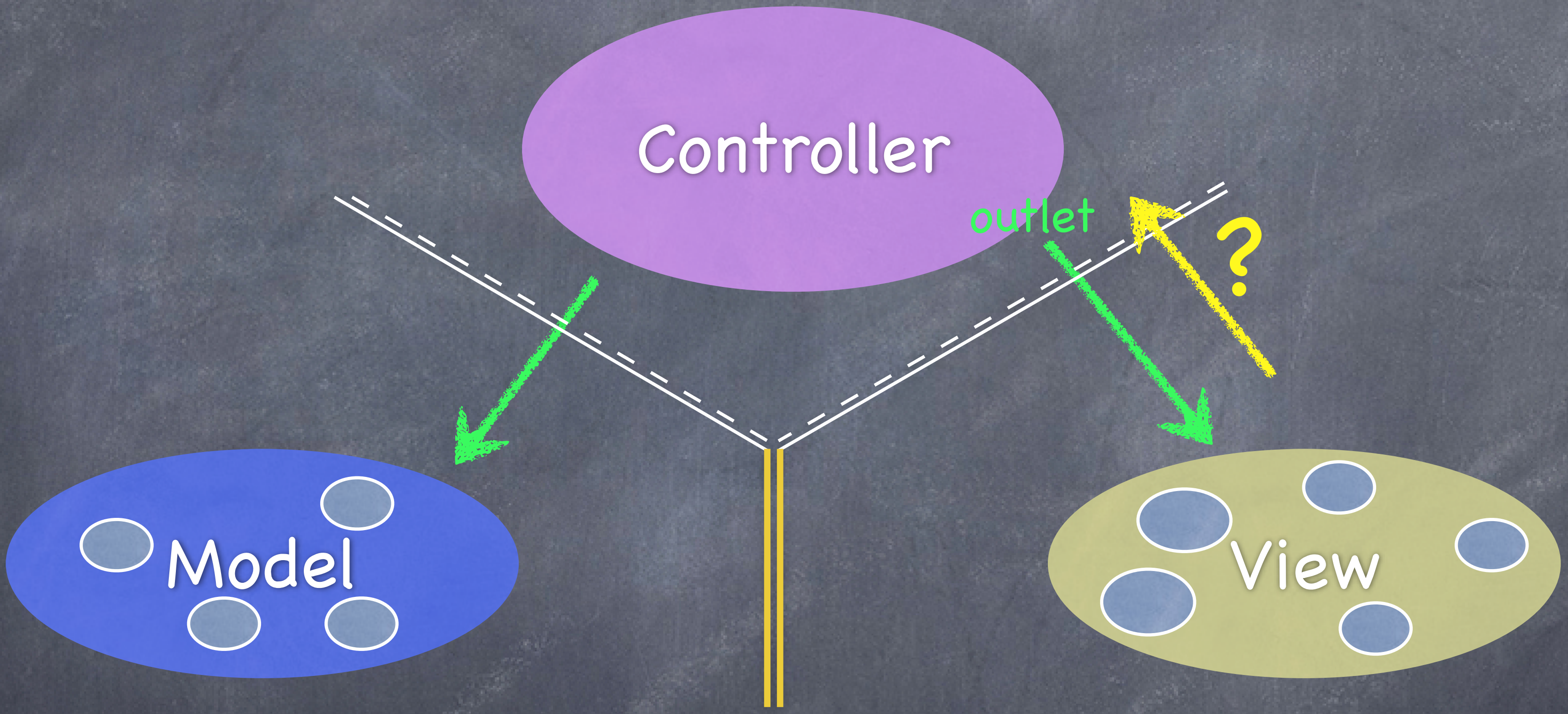
Controllers can also talk directly to their View.

MVC



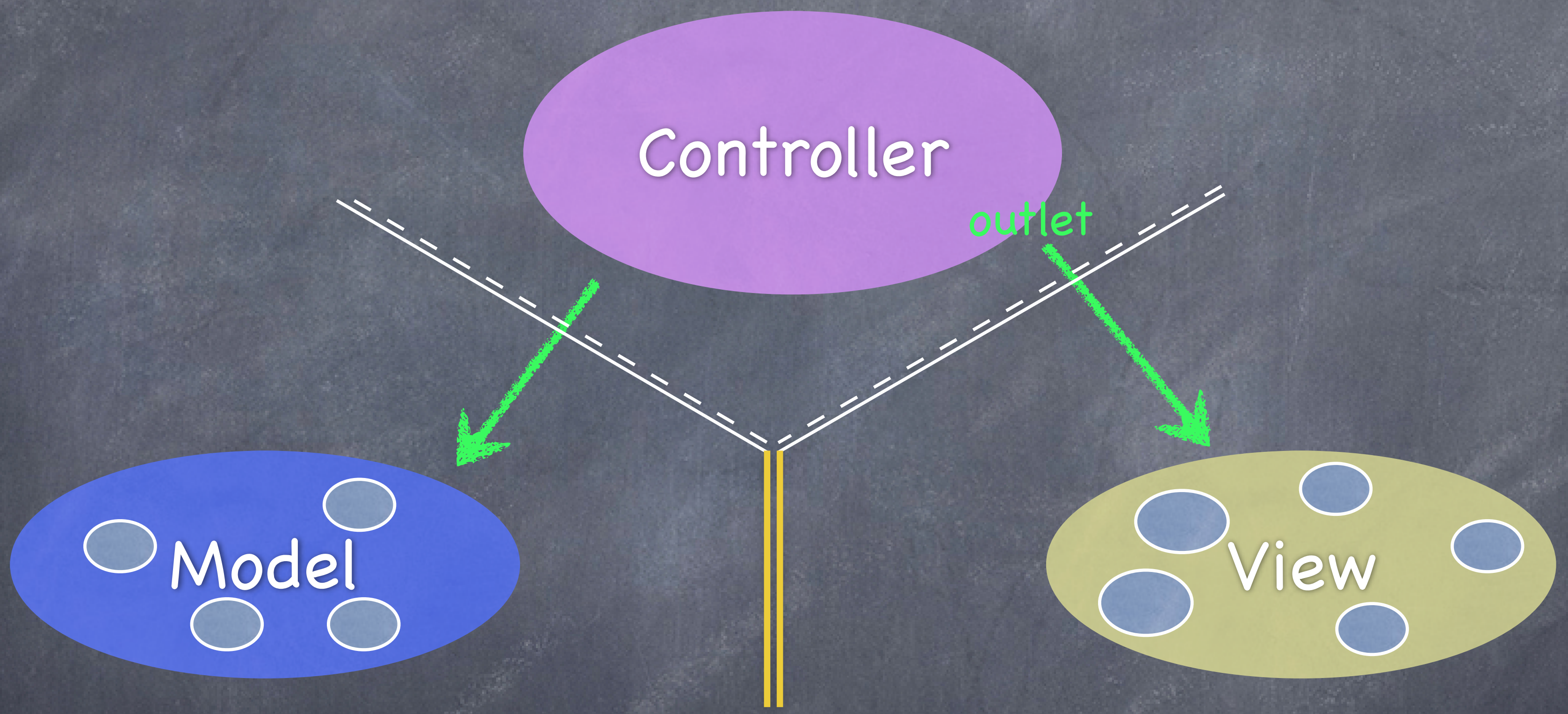
The Model and View should never speak to each other.

MVC



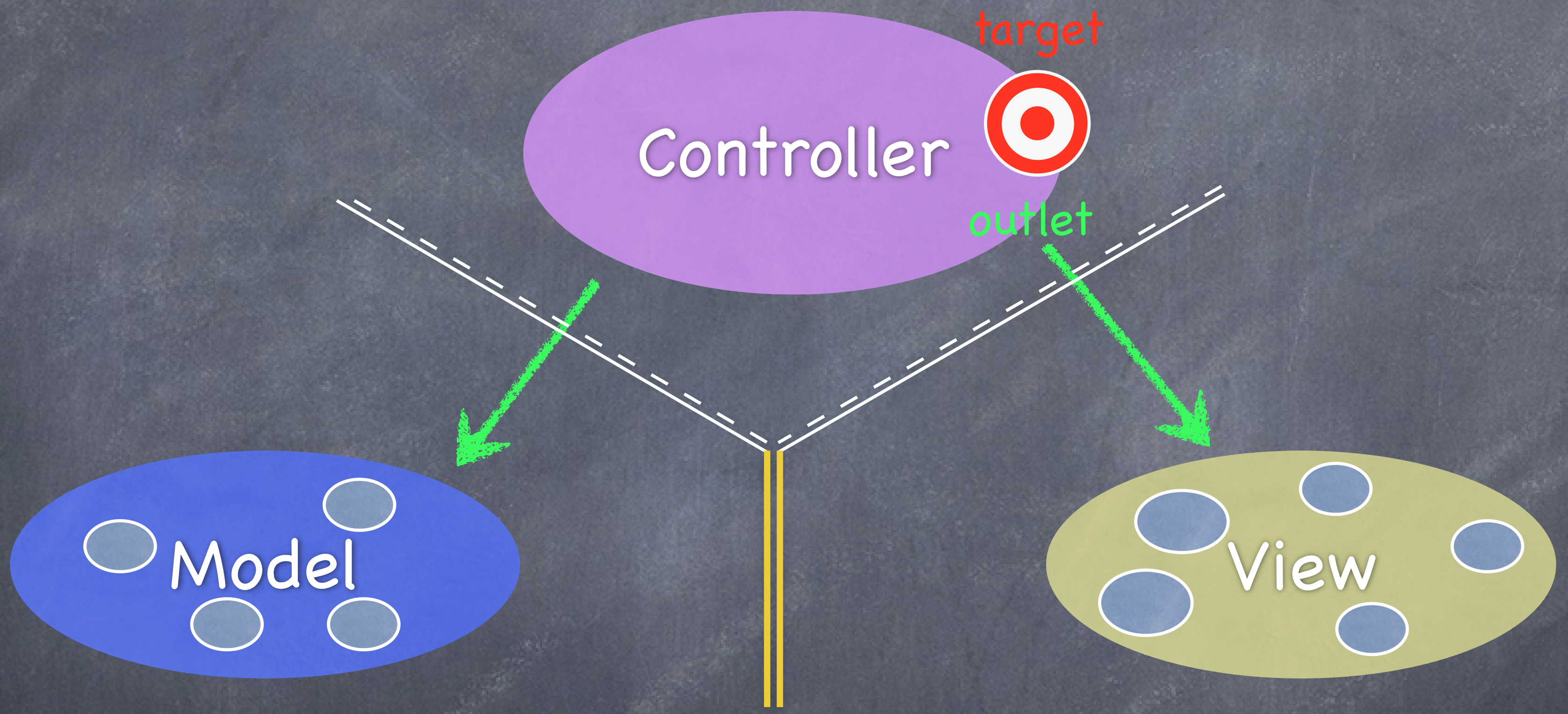
Can the View speak to its Controller?

MVC



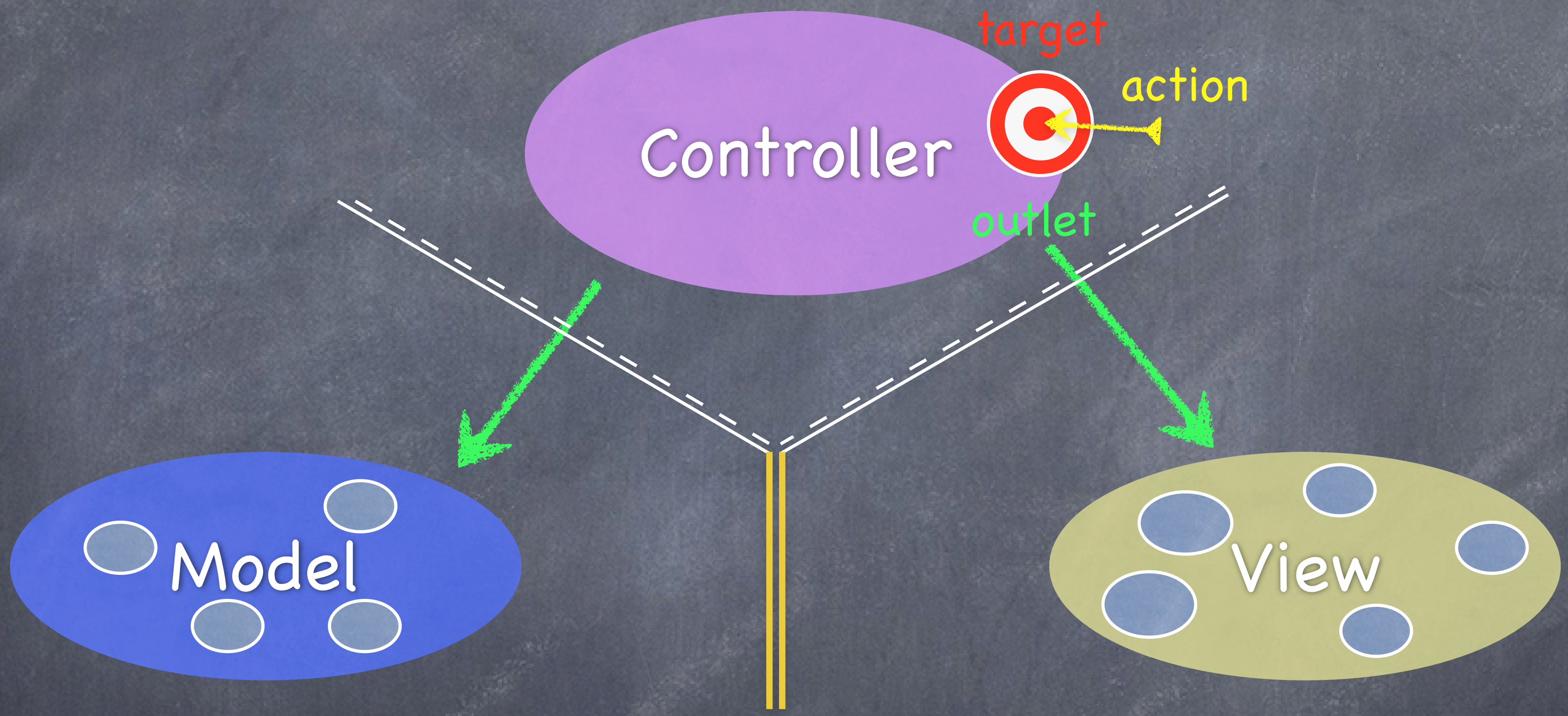
Sort of. Communication is "blind" and structured.

MVC



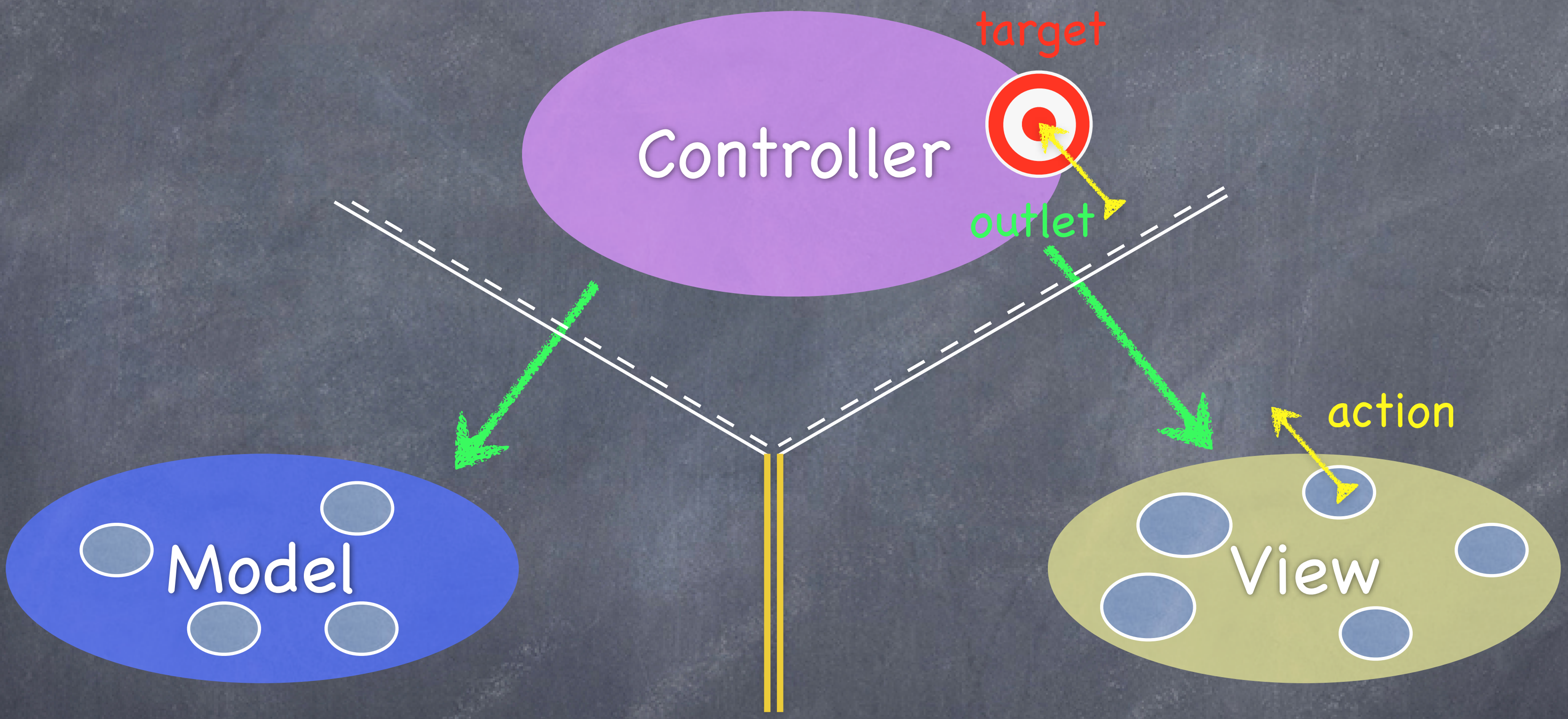
The Controller can drop a target on itself.

MVC



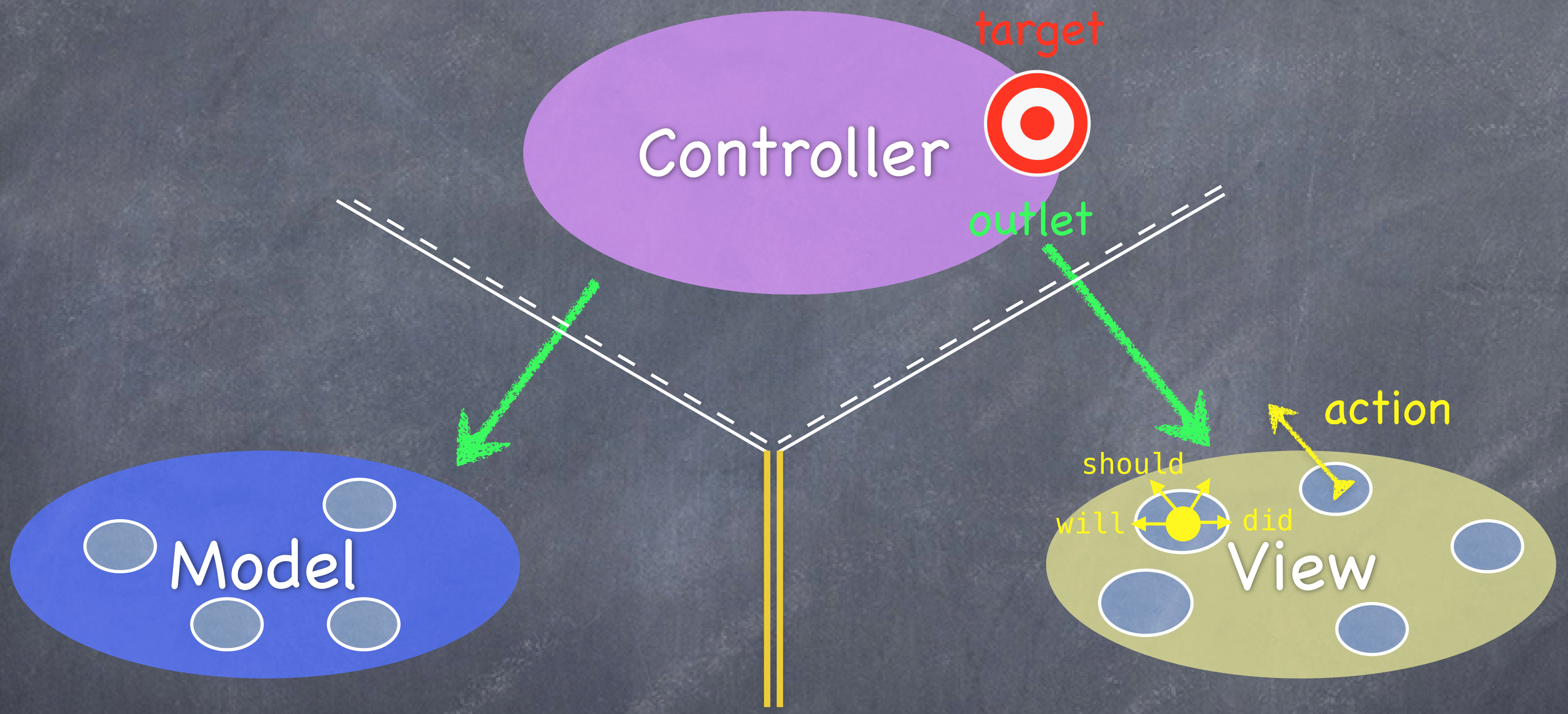
Then hand out an action to the View.

MVC



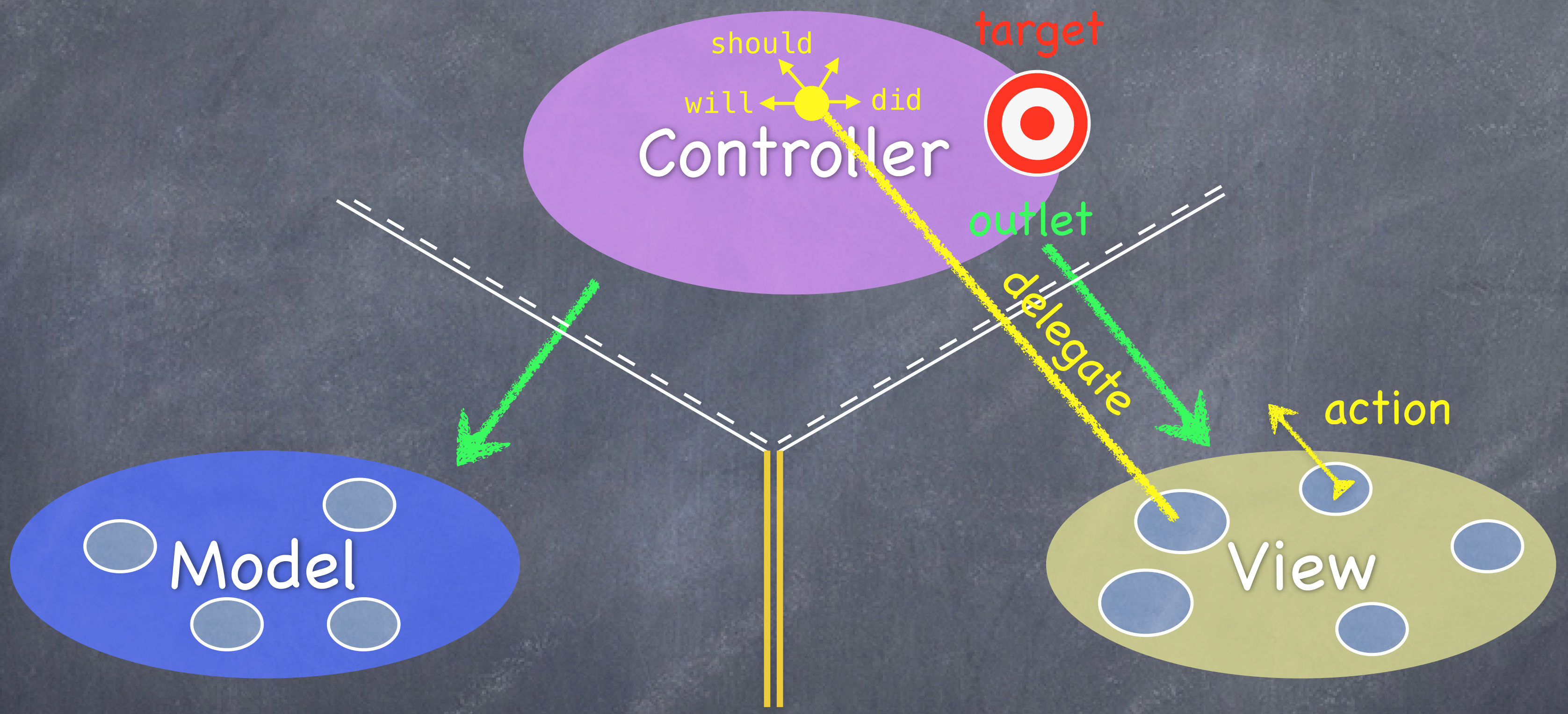
The View sends the action when things happen in the UI.

MVC



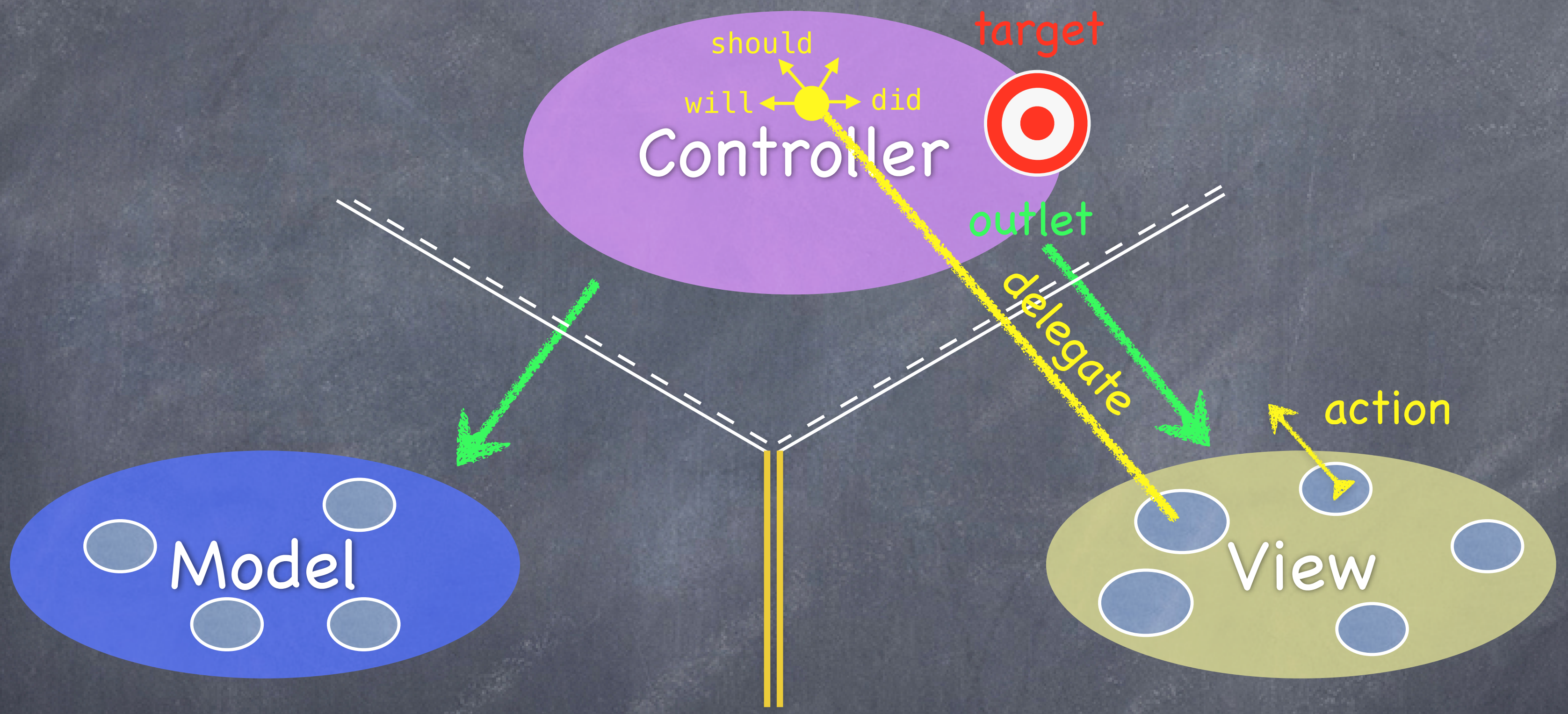
Sometimes the View needs to synchronize with the Controller.

MVC



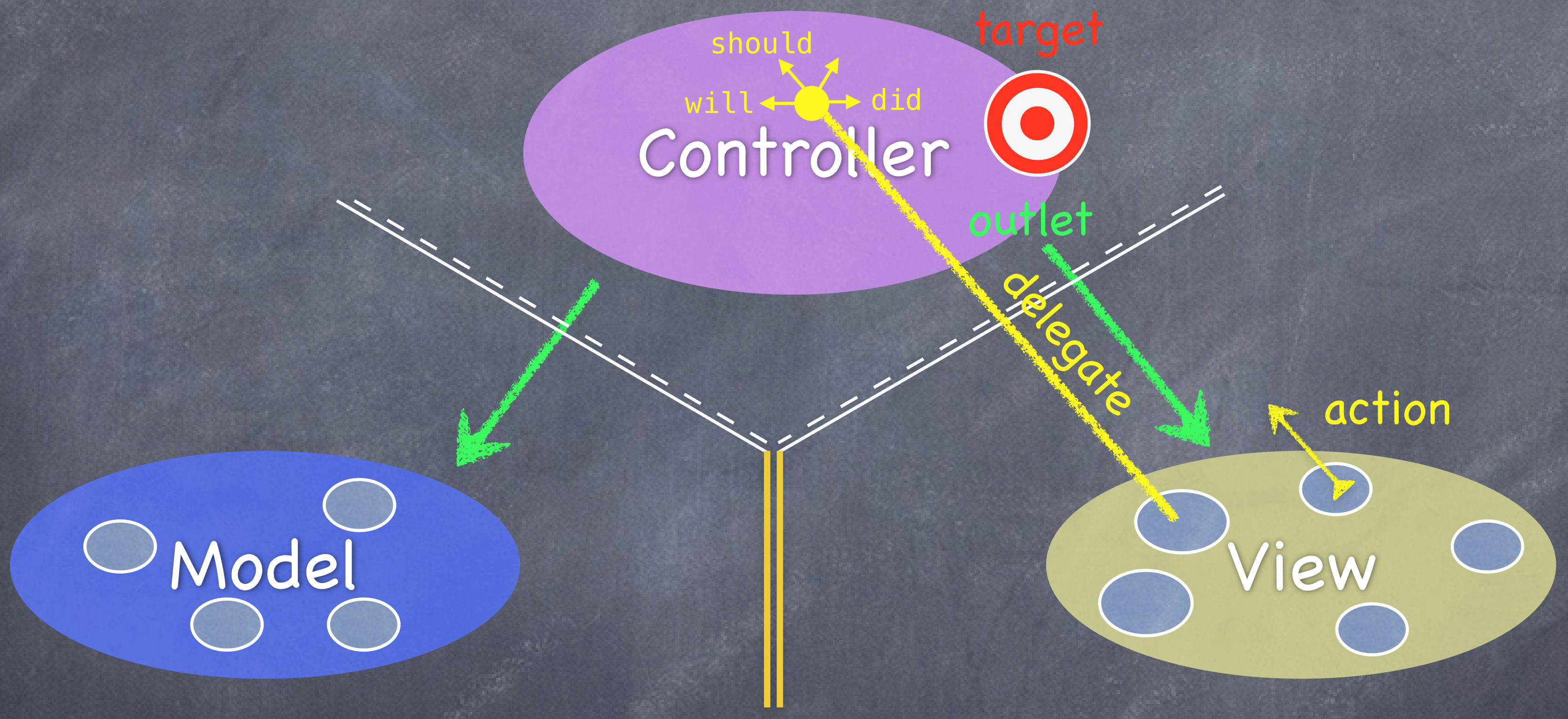
The Controller sets itself as the View's delegate.

MVC



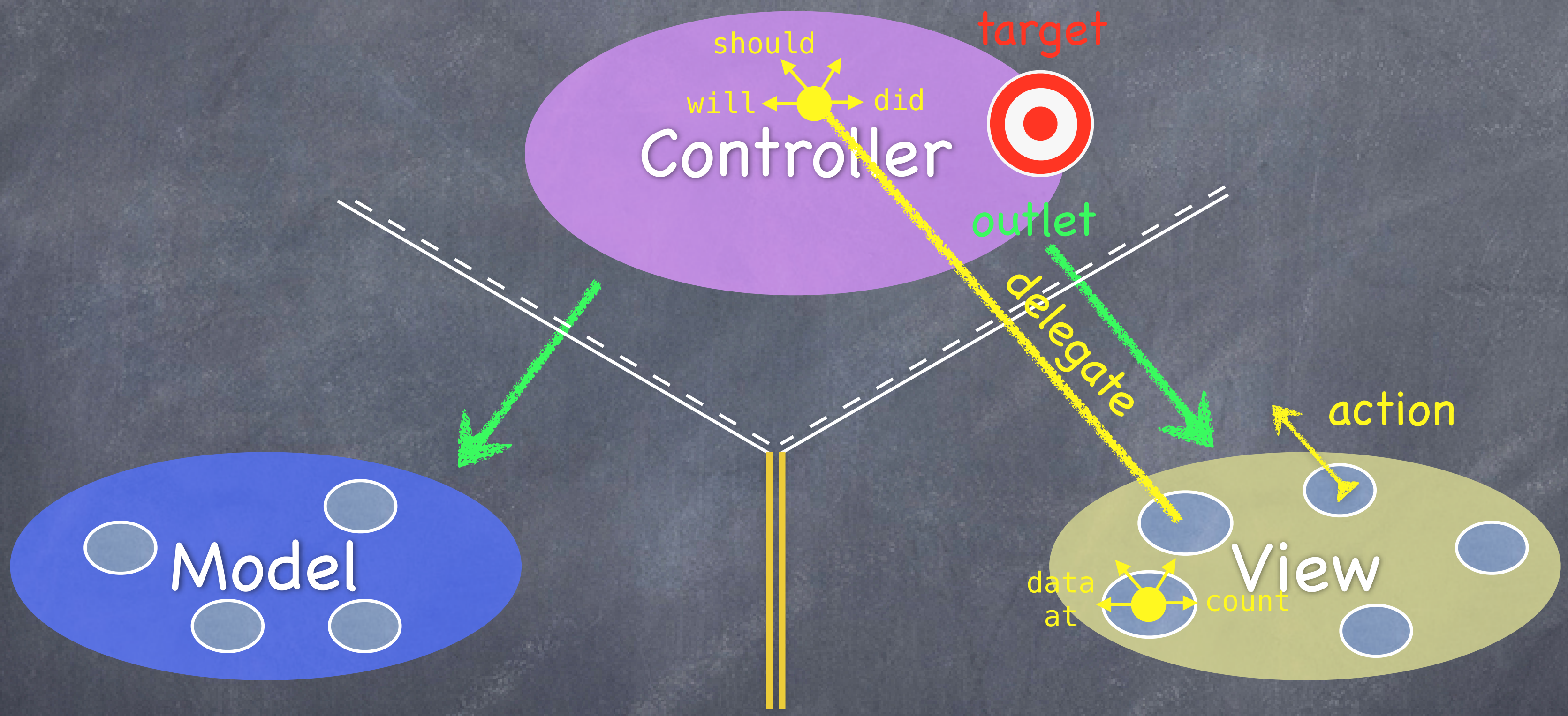
The delegate is set via a protocol (i.e. it’s “blind” to class).

MVC



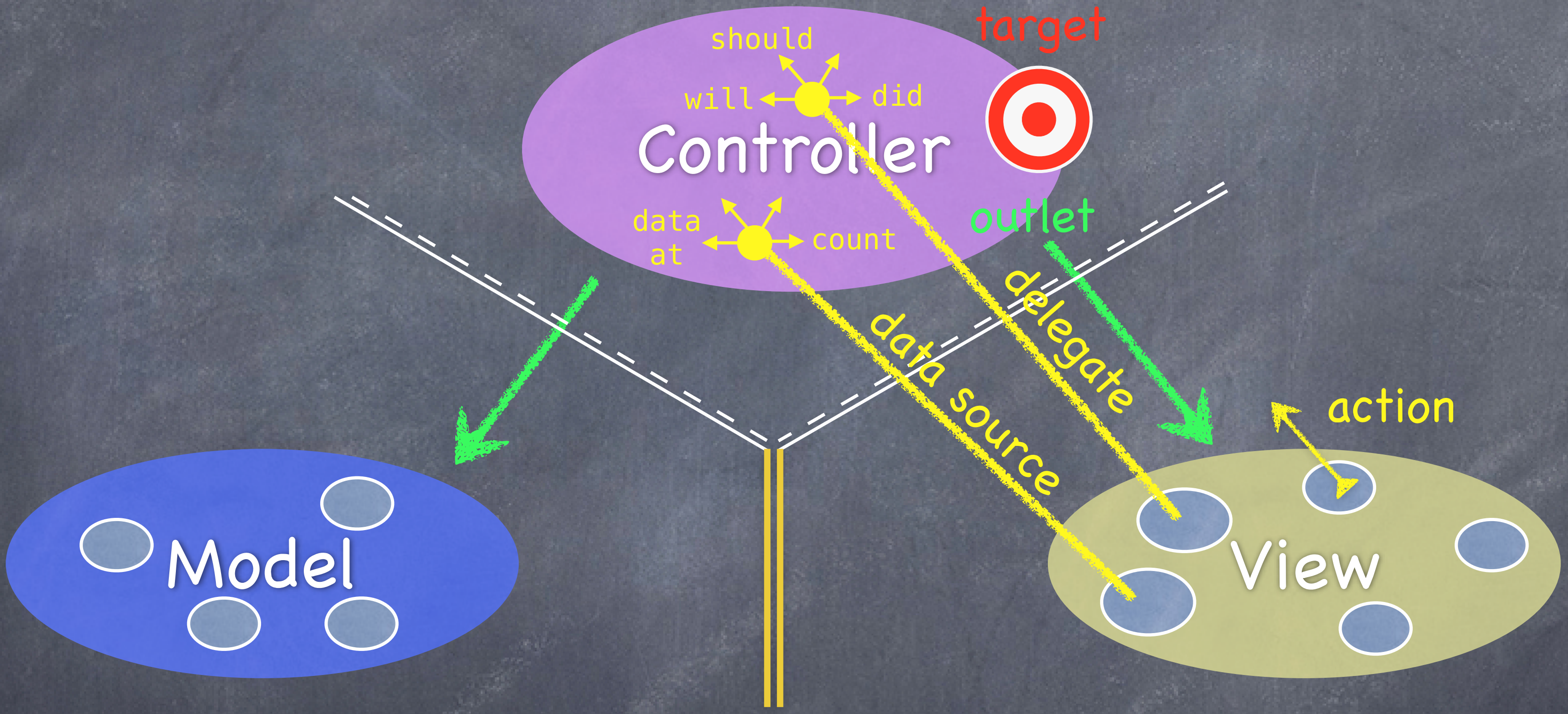
Views do not own the data they display.

MVC



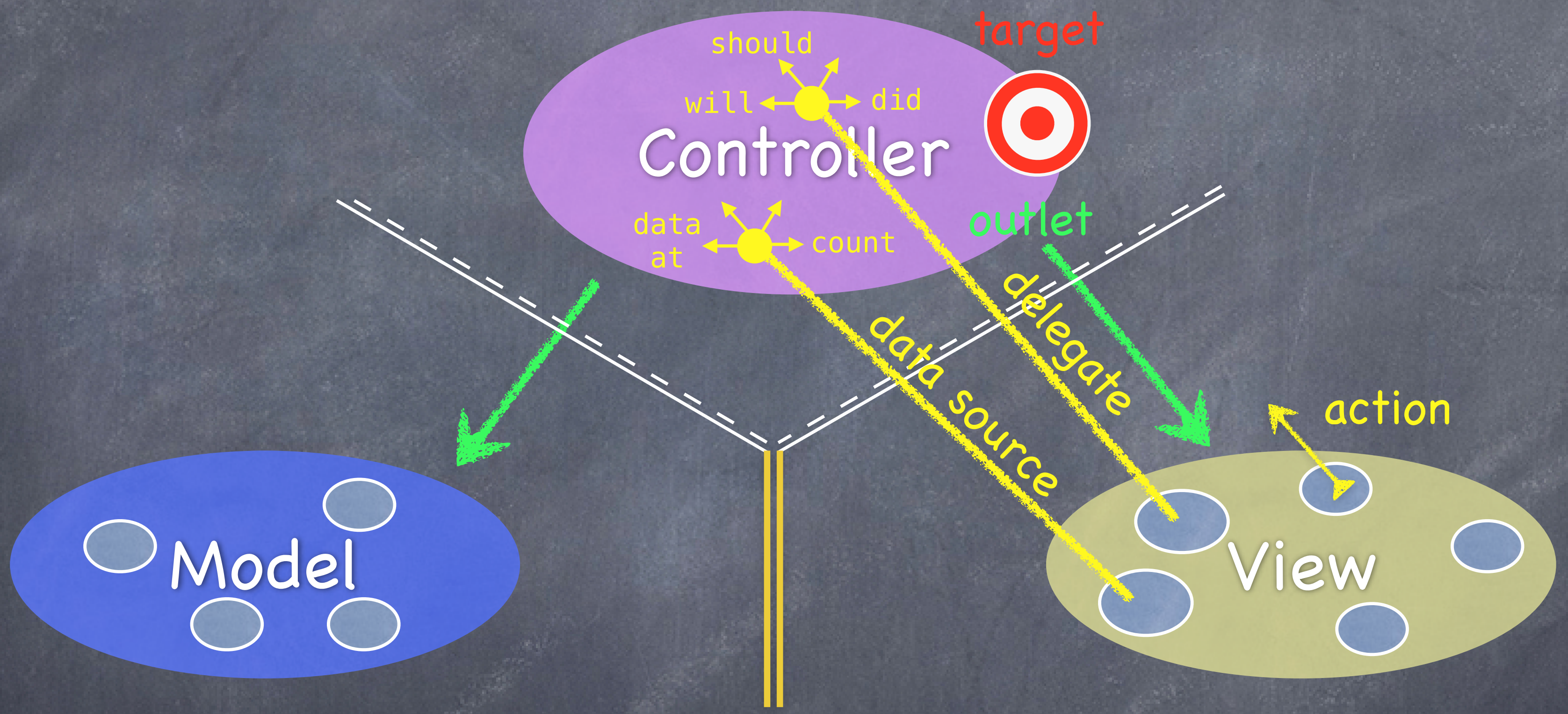
So, if needed, they have a protocol to acquire it.

MVC



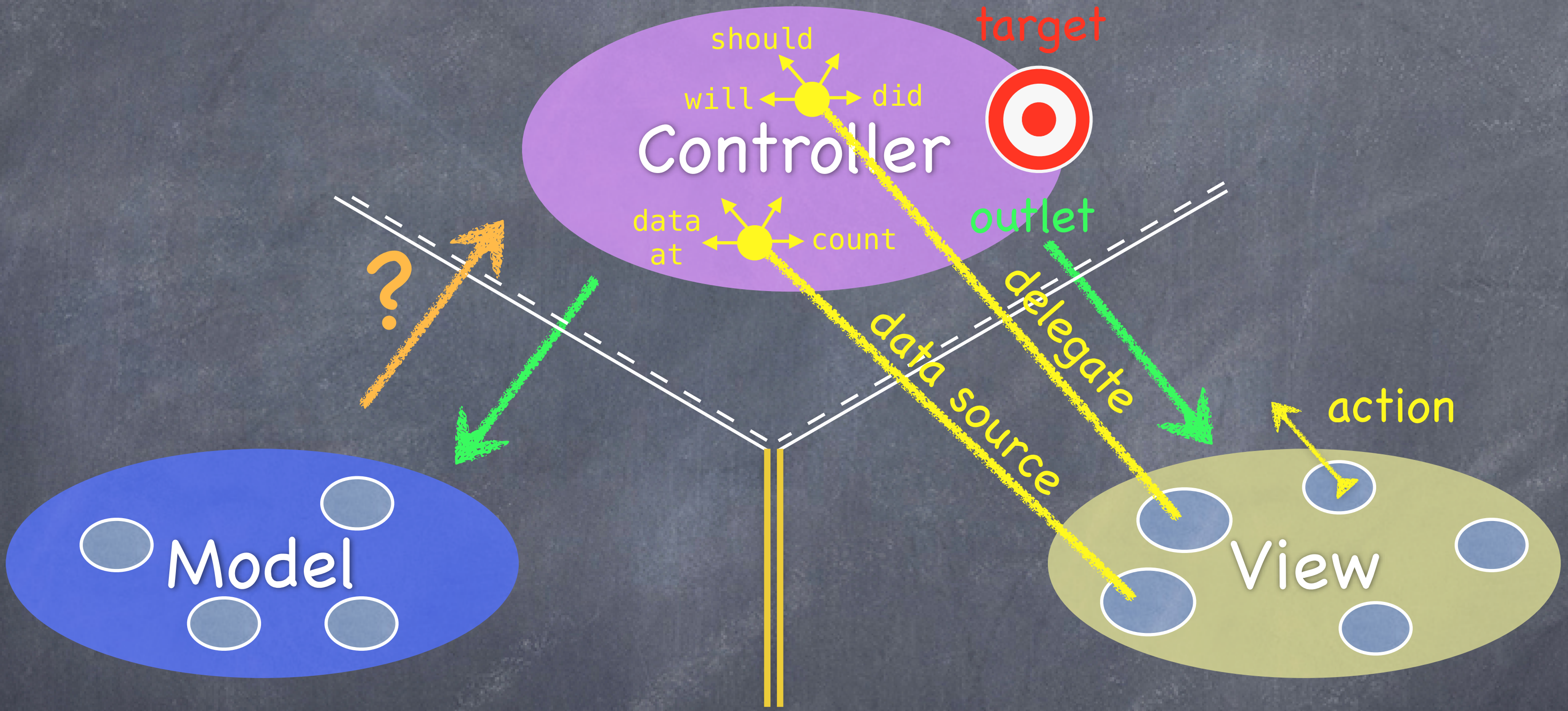
Controllers are almost always that data source (not Model!).

MVC



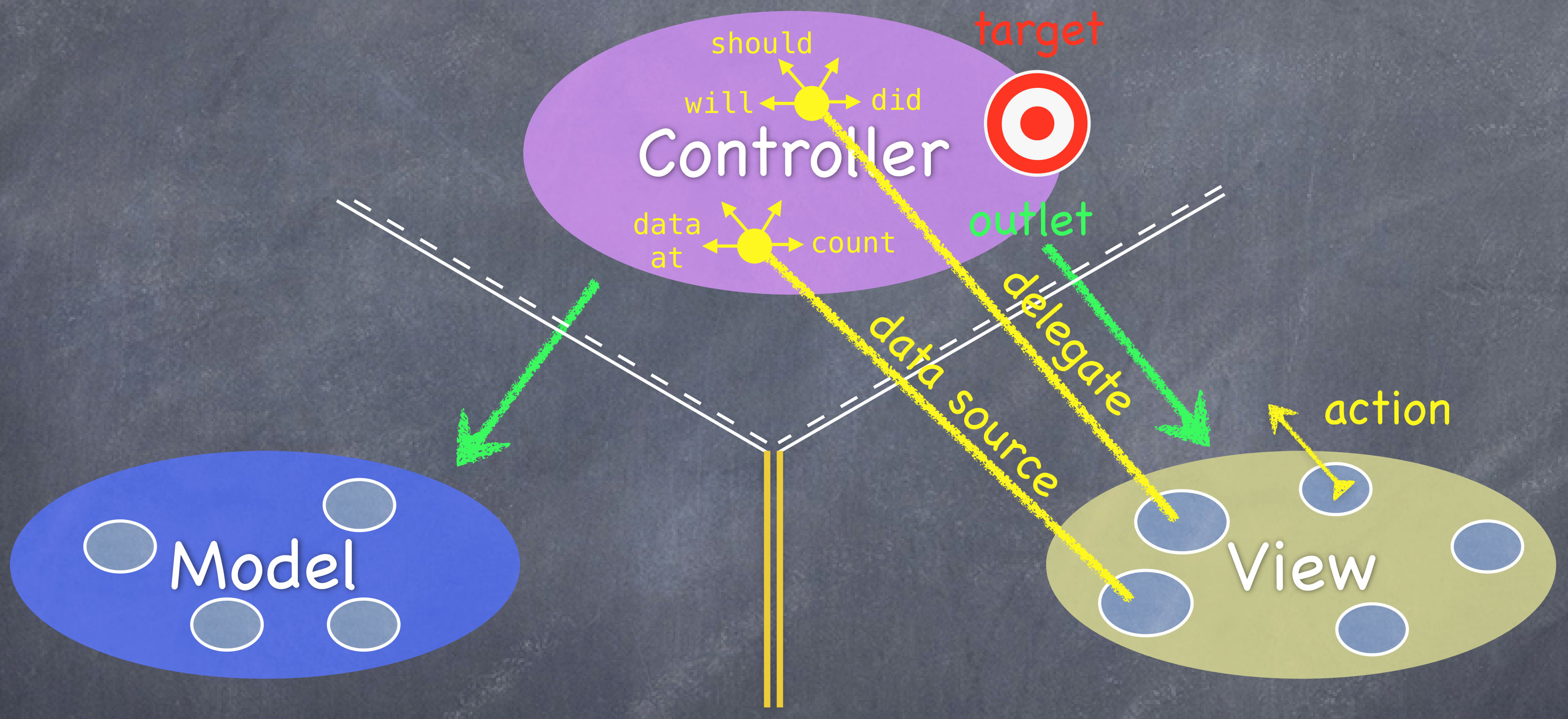
Controllers interpret/format Model information for the View.

MVC



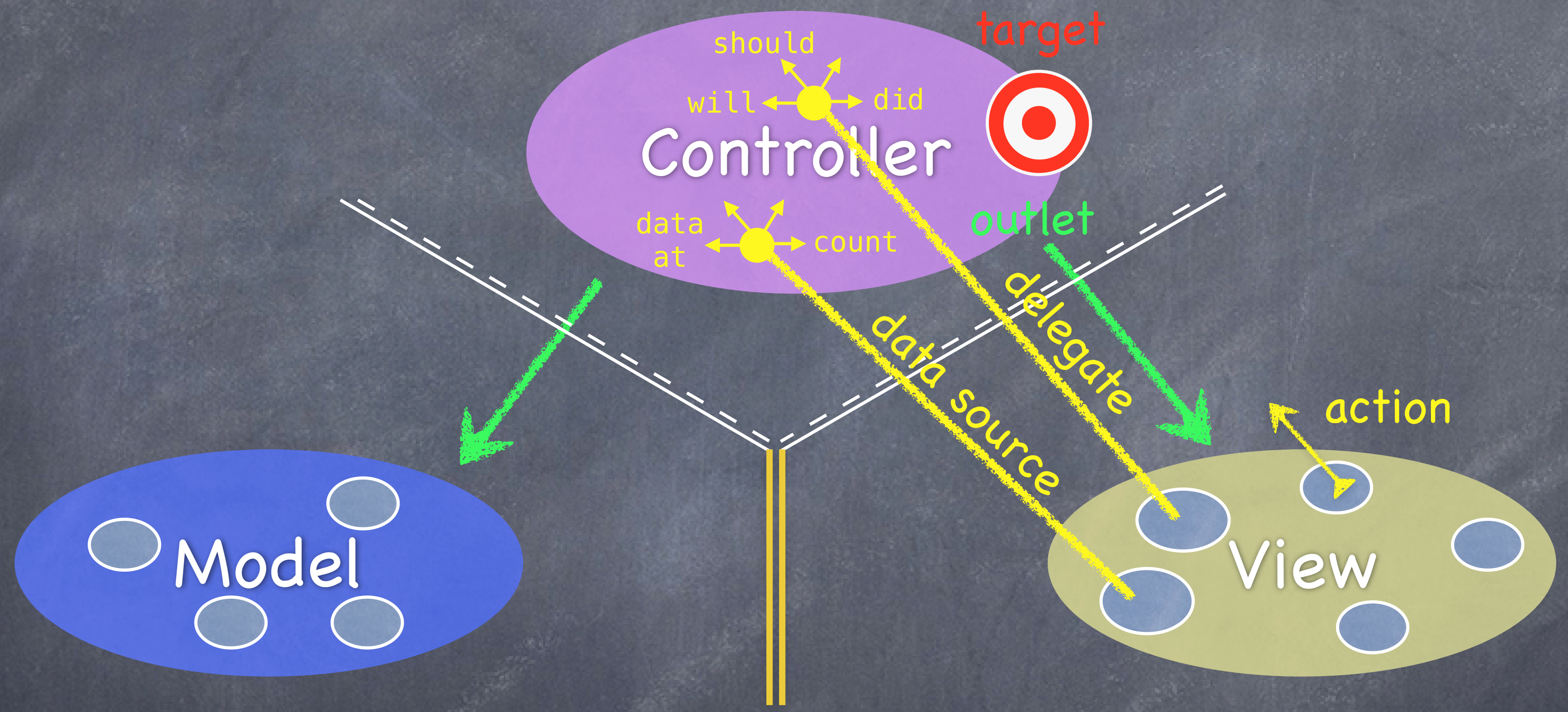
Can the Model talk directly to the Controller?

MVC



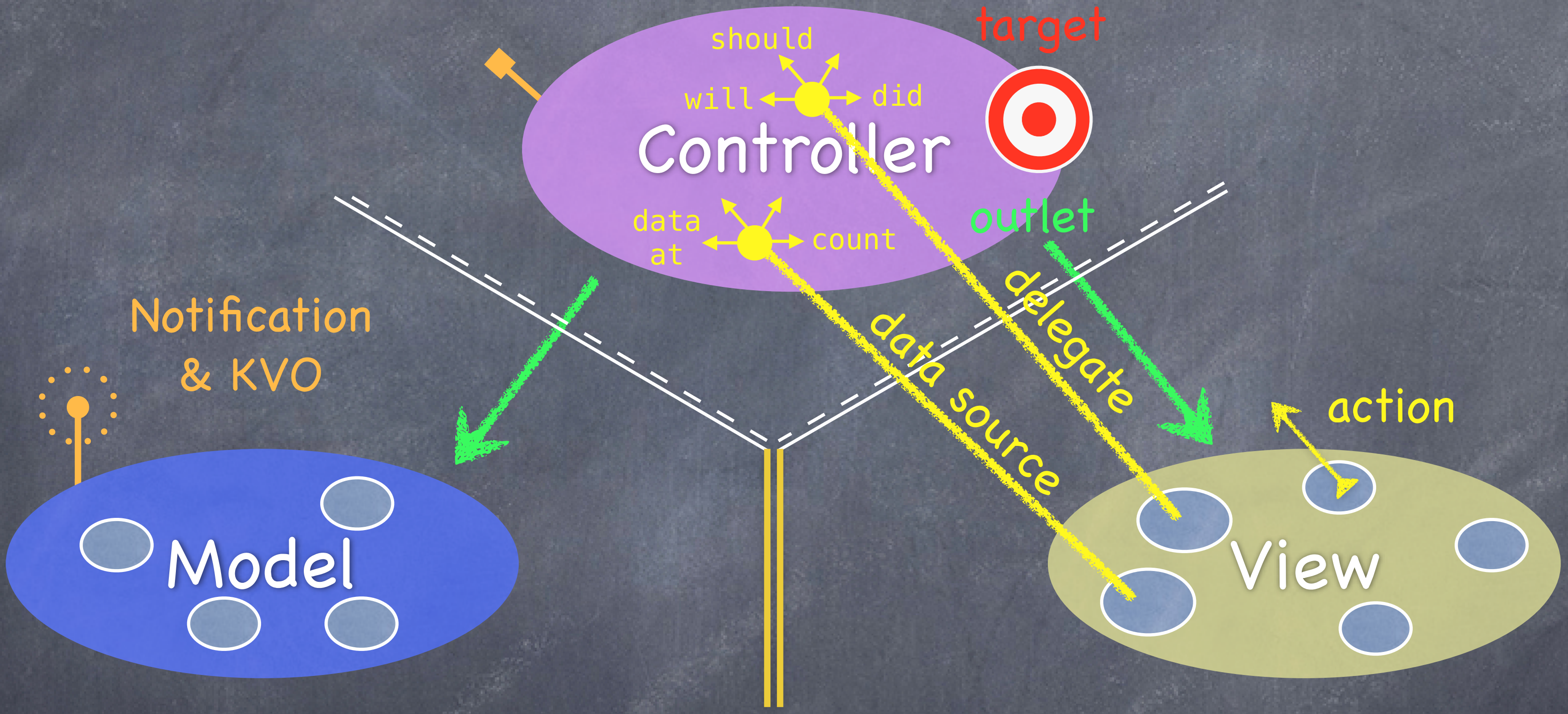
No. The Model is (should be) UI independent.

MVC



So what if the Model has information to update or something?

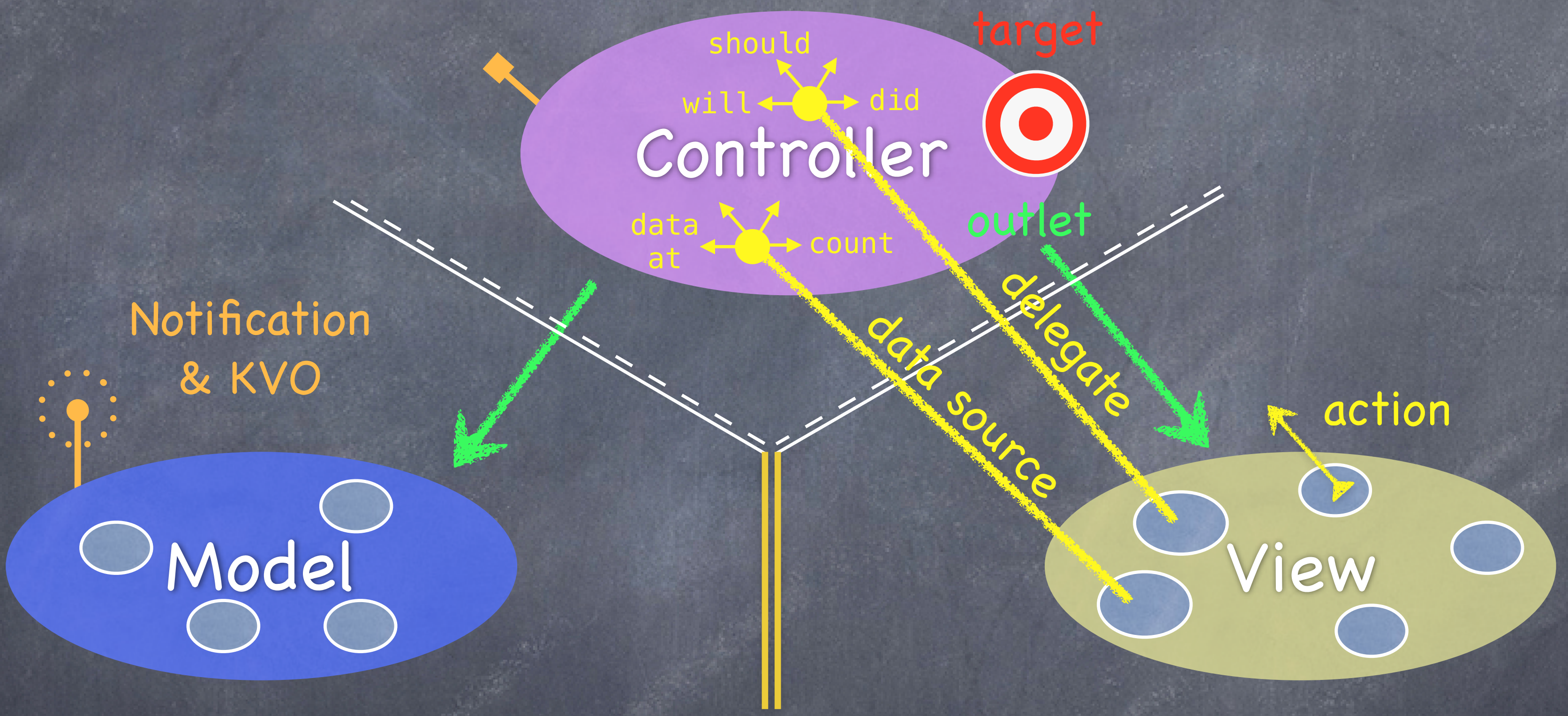
MVC



It uses a "radio station"-like broadcast mechanism.

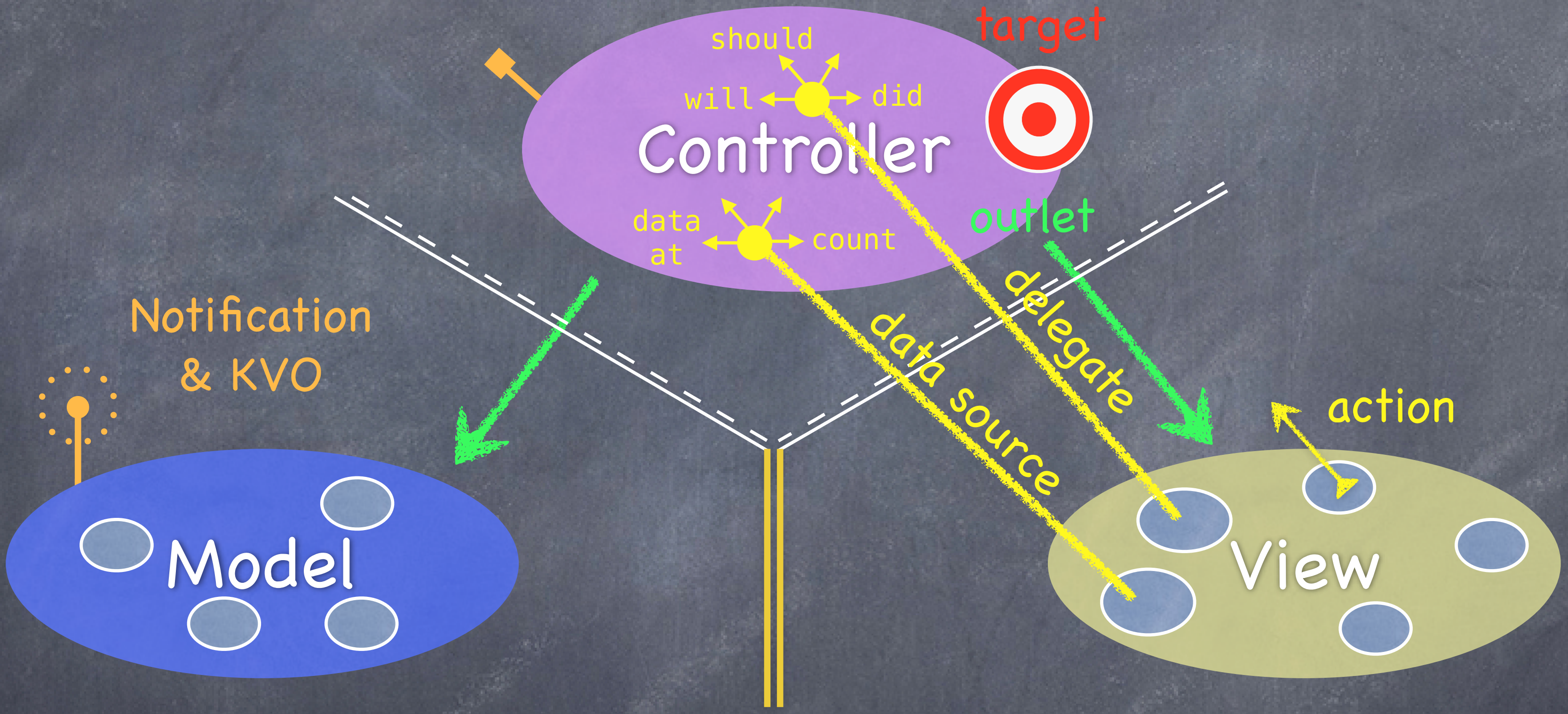
Stanford CS193p
Fall 2013

MVC



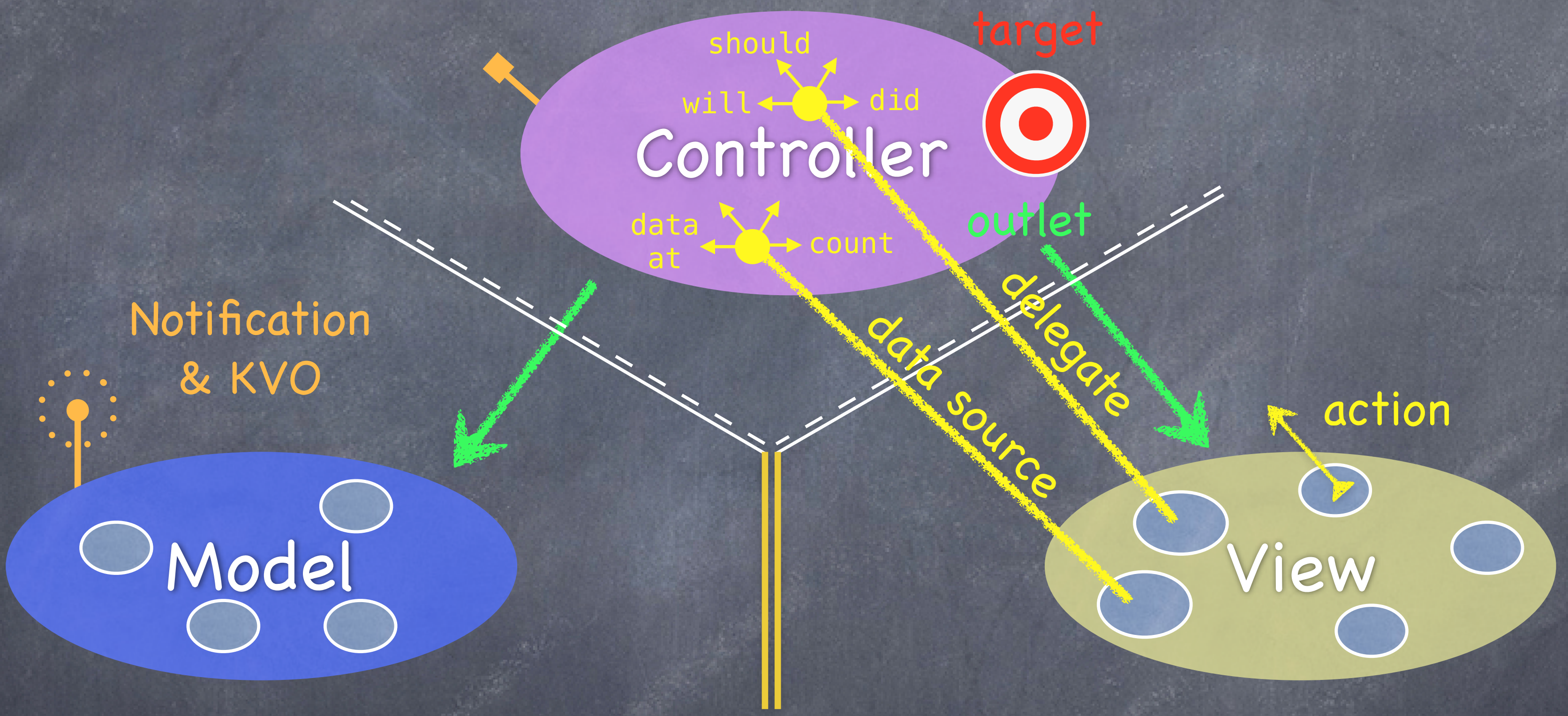
Controllers (or other Model) “tune in” to interesting stuff.

MVC



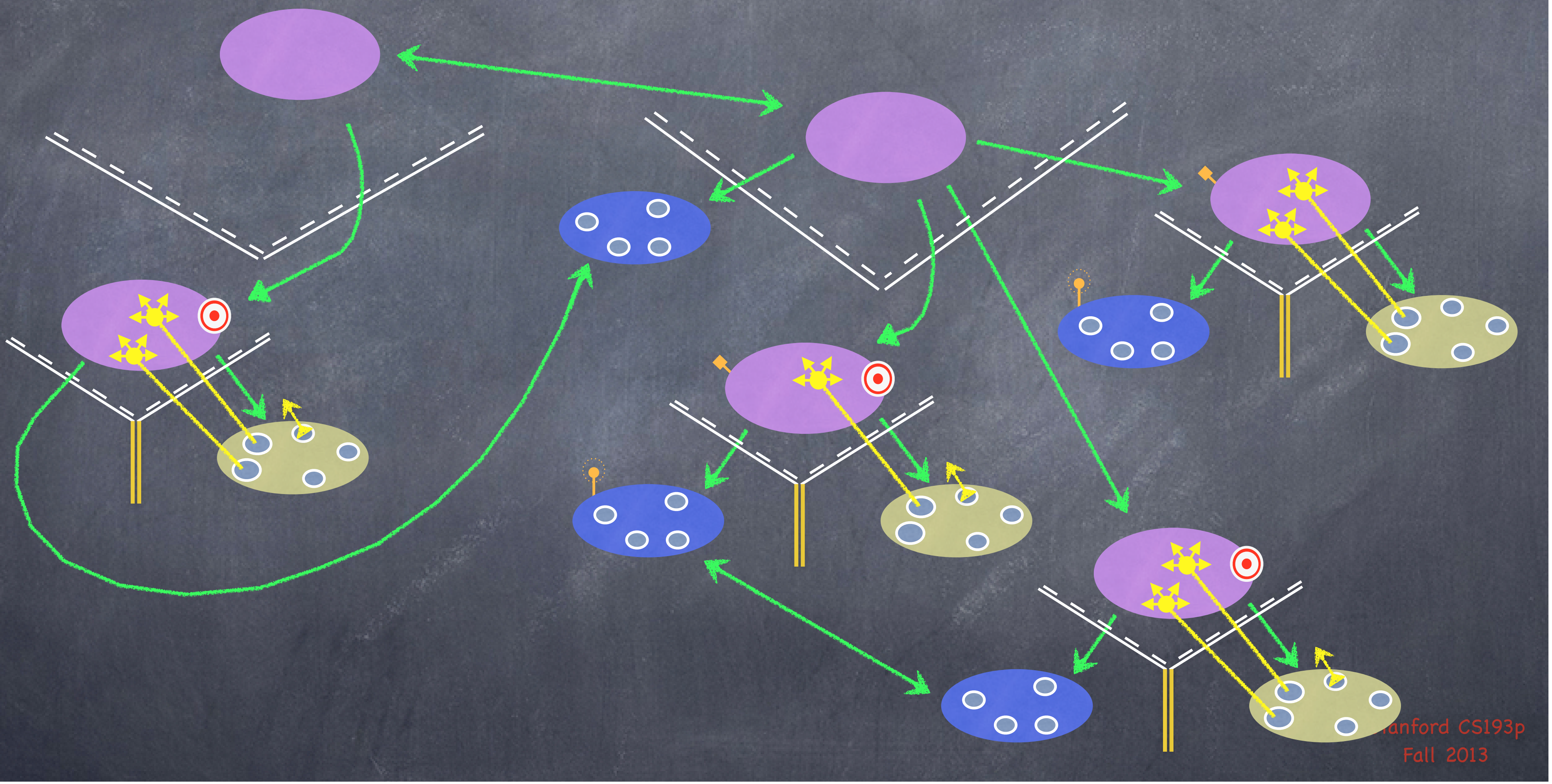
A View might "tune in," but probably not to a Model's "station."

MVC



Now combine MVC groups to make complicated programs ...

MVCs working together



MVCs not working together