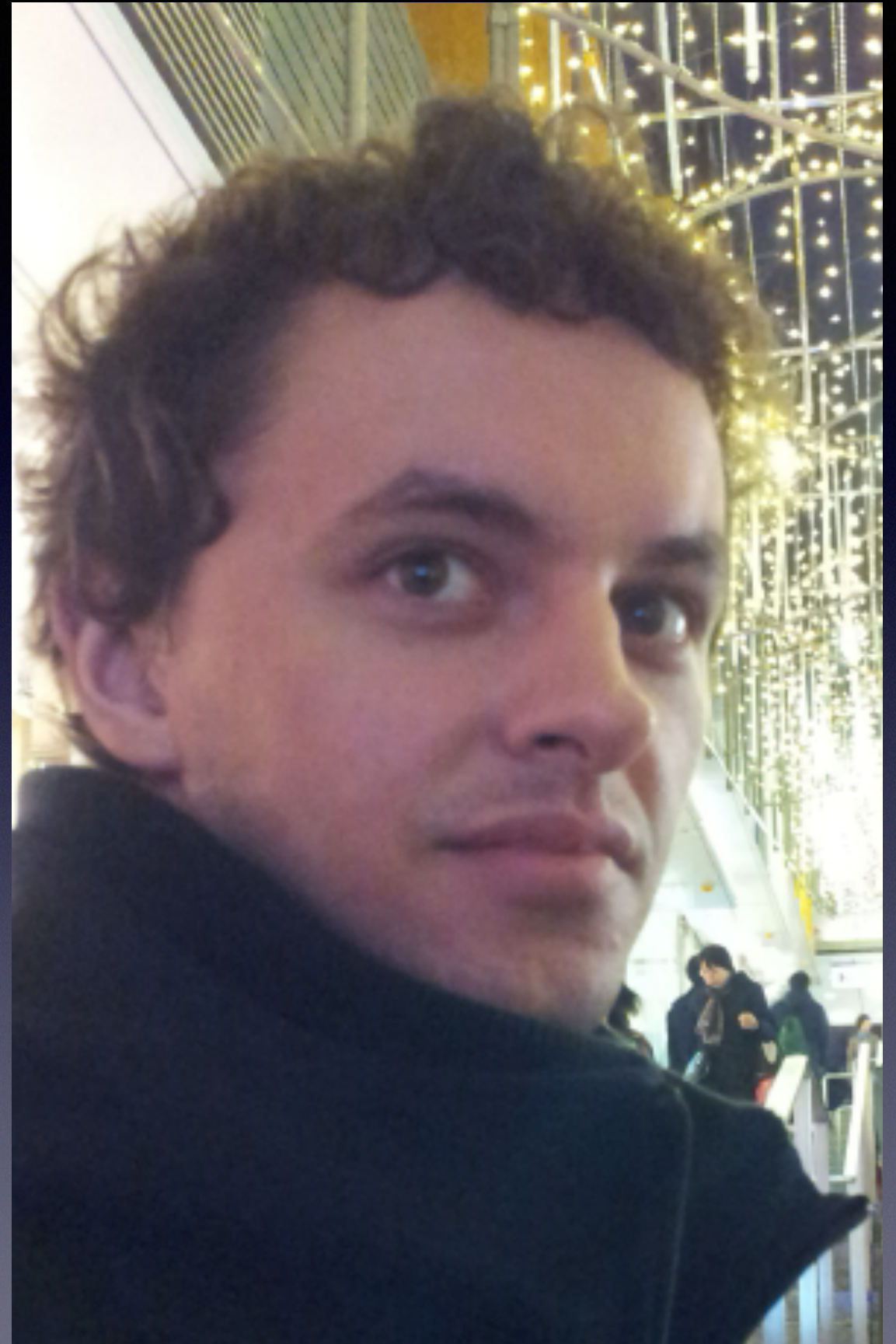


Dynamic ViewControllers with Interactors

Caesar Wirth
November 10, 2015

Caesar Wirth

- Online
 - Twitter: @cjwirth
 - Github: cjwirth
- CyberAgent, Inc - 2012年に入社
 - パシャオク (👤)
 - Simplog
 - Ameba Ownd
- SwiftSummitに行ってきた
 - 時差ボケがひどい



Common Problems in ViewControllers

- Does too much — violates Single Responsibility Principle
- Hard to change or test (and so we don't...)
- Too much mutable state
- MVC - Massive View Controller

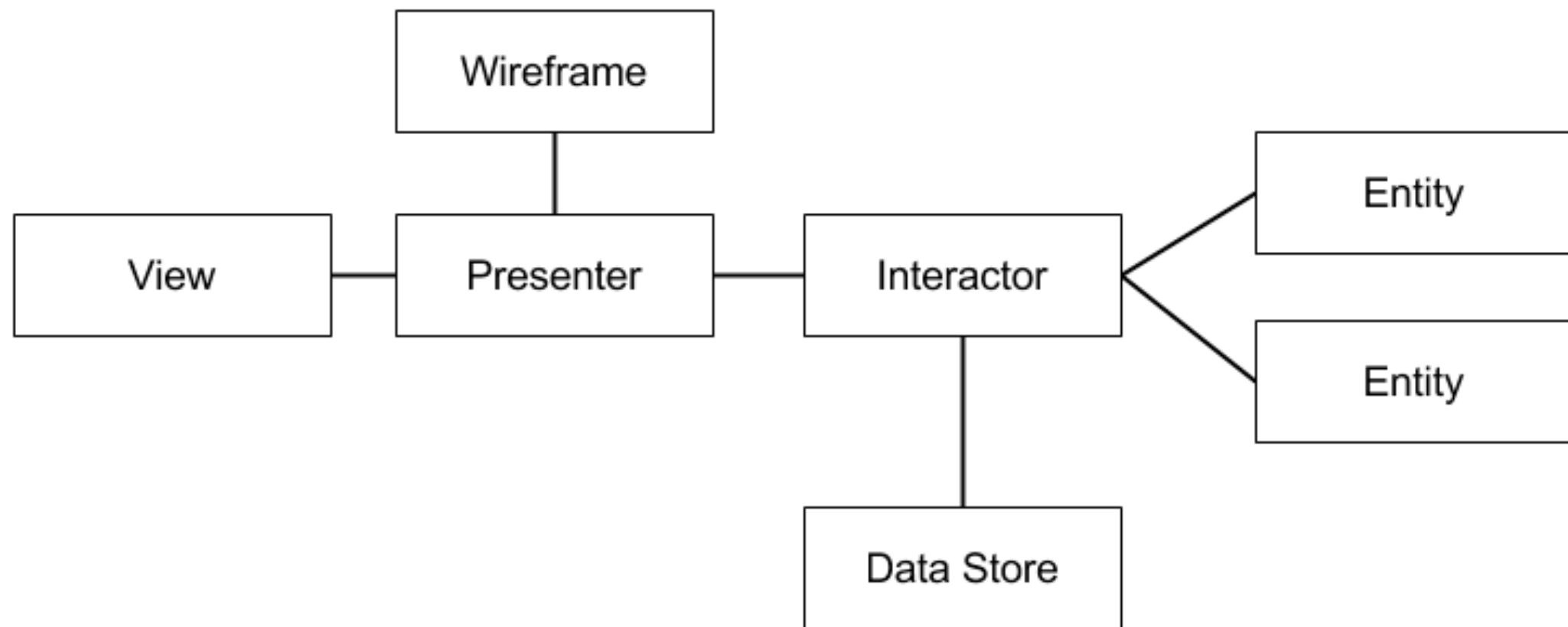
ViewControllers should:

- Control views — show, hide, move, and display
- Have data passed into them to display
- Not care about any other ViewControllers

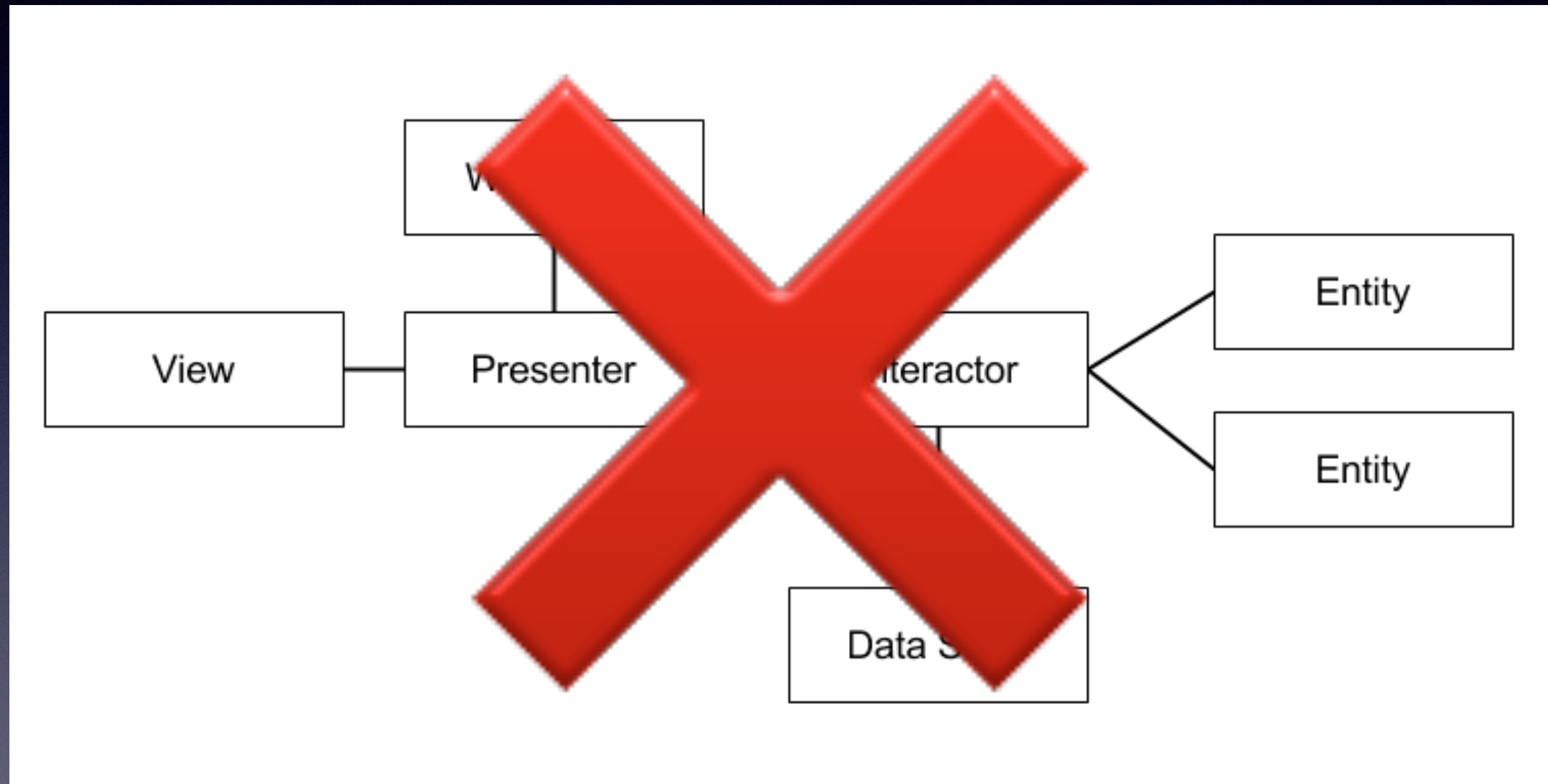
VIPER



VIPER



VIPER





“But refactoring to this is hard...”

—Me, a few months ago



“I’ll just refactor the user interactions!”

–Me, a few months ago

Interactors in Ownd

- Interactor is passed in on creation
- All user actions go through an Interactor
- Interactor decides what happens for each action
- Interaction is based on state, for example login/logout

Example

Benefits

- ViewController has less responsibility
- Behavior can be defined/changed at runtime
- ViewControllers are more reusable
- Easier to change and test
- Classes are shorter, easier to reason about

参考

- Let's Play: Refactor the Mega Controller!
Andy Matuschak
- The Clean Architecture - Uncle Bob
- Introduction to VIPER - Mutual Mobile
- Architecting iOS Apps with VIPER - objc.io
- Design Patterns (Strategy) - Gang of Four

Example app and slides

<https://github.com/cjwirth/Tweets-InteractorExample>



Thanks!