CSC207 Tutorial 6

TUT0401 - 20th October 🍂

before we begin...

- have a team member fork the repo
- share it with your teammates
- all teammates: clone the repo & try running the code

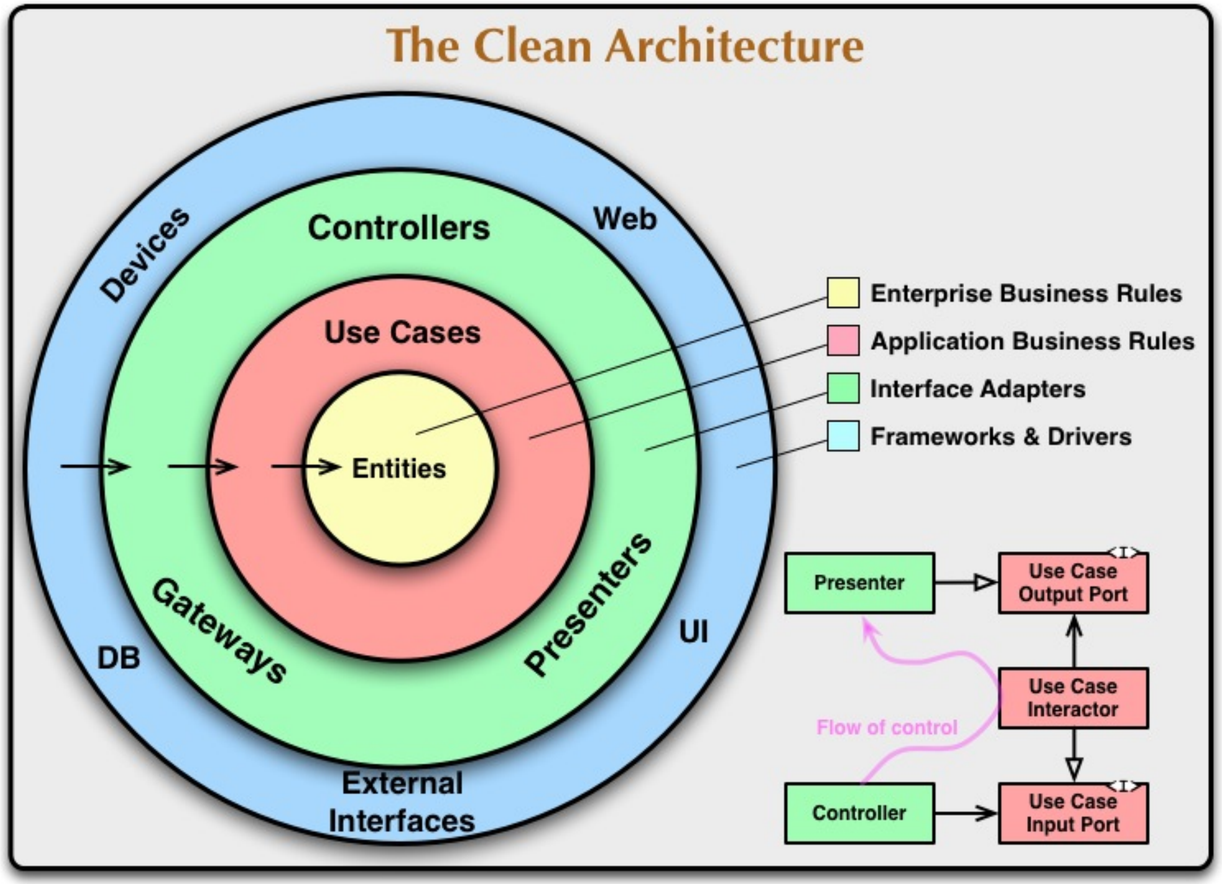
lab overview

- Introduction to Clean Architecture
- **Group Activity:** Login/logout
  - Goal: implement the logout use case
  - to be completed by the end of lab (for credit!)
- Start filling out your project blueprint (due by next lab!)

clean architecture

a refresher

- A layered model of how control flows during a user action
- Outer layers depend on inner layers, but not vice-versa
- UI -> Controller -> Interactor -> Presenter -> UI



clean architecture

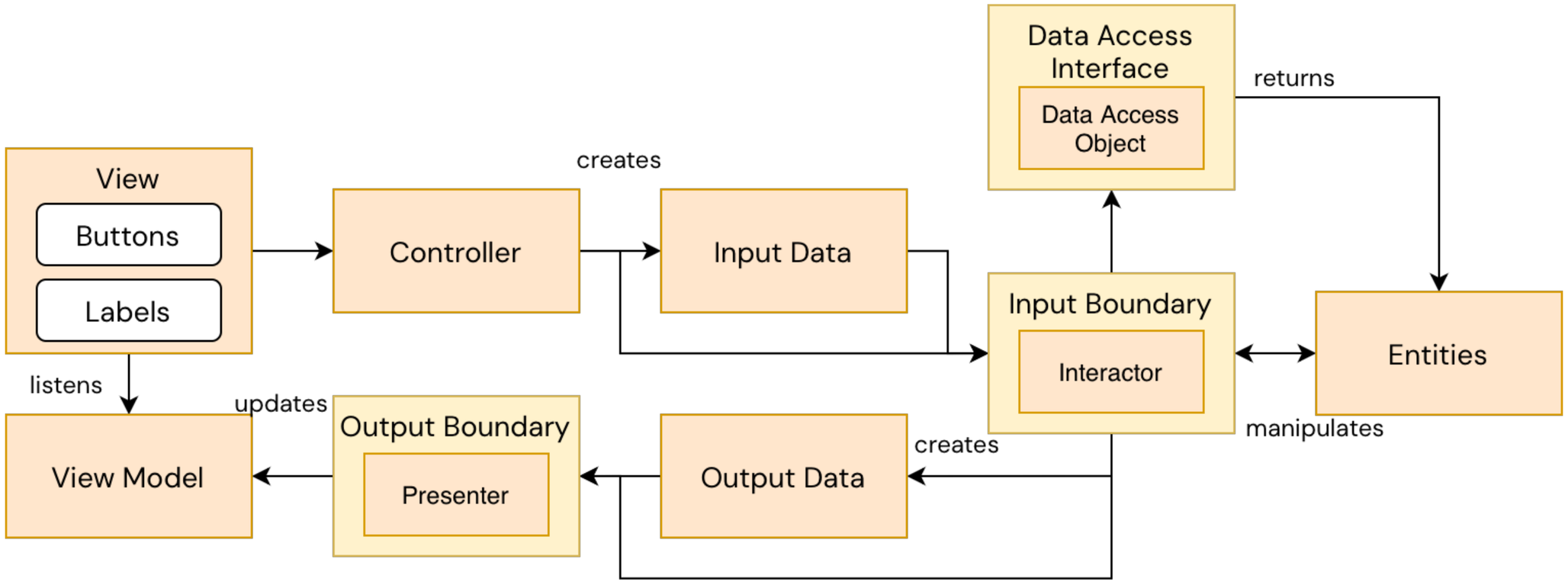
a word of reassurance...

- this was super confusing to me at first.
- thankfully, this is not really used in industry.
- **HOWEVER**, what's important is the concepts of abstraction, and the skill of writing clean, modular and testable code.

this opinion is not endorsed by the 207 teaching team!!!

clean architecture

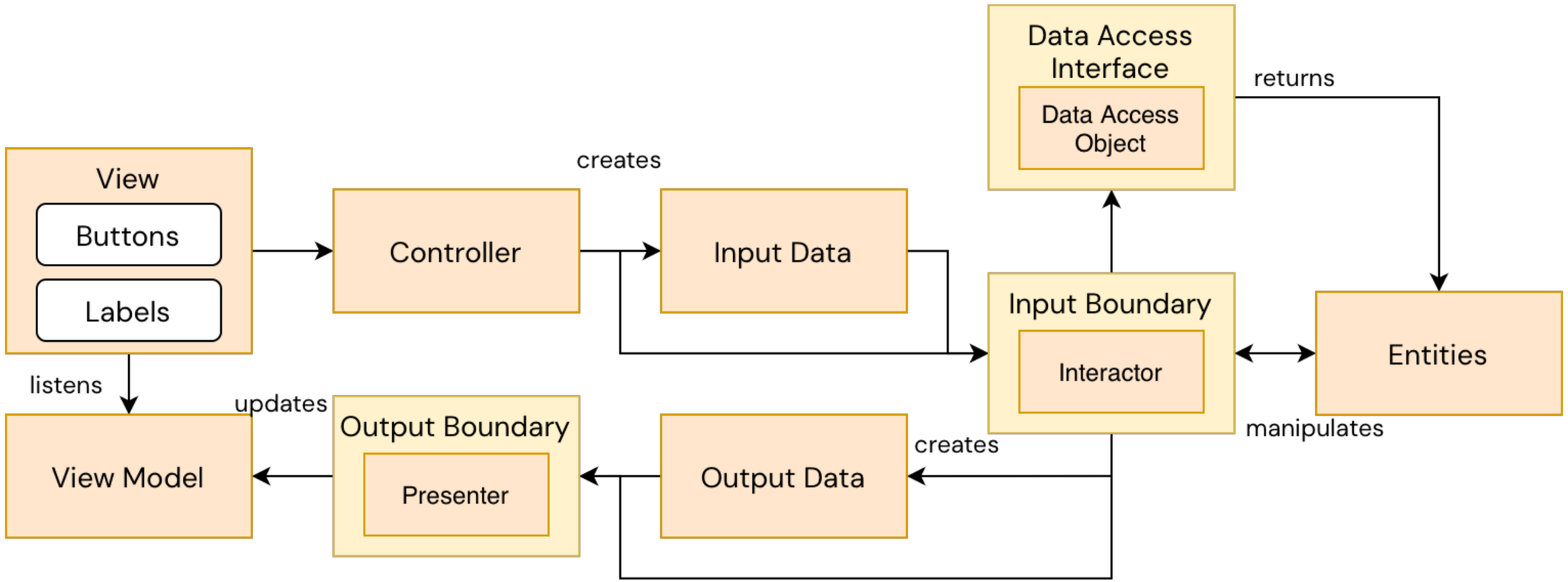
a visualization of a use case



orange = classes; yellow = interfaces

clean architecture

a visualization of a use case



notice how dependency inversion is adhered to!

clean architecture

major components, explained

- **view:** displays the UI *(LoginView: text fields for username & password)*
- **controller:** given a user action, call the interactor with the right inputs
- **interactor:** given inputs, manipulate entities and call the presenter with the right outputs *(check if user exists / password matches)*
- **presenter:** given outputs, update the viewModel & switch views

clean architecture

okay, how is the UI managed?

- each UI view is managed by a View, a State, and a ViewModel
- views are managed by a ViewManager
- views are updated by the Presenter by each use case
- **View:** UI components
- **ViewModel**: an abstraction of *how* the data being presented
- **State**: an abstraction of *what* is being presented