LAYERED ARCHITECTURES

Request

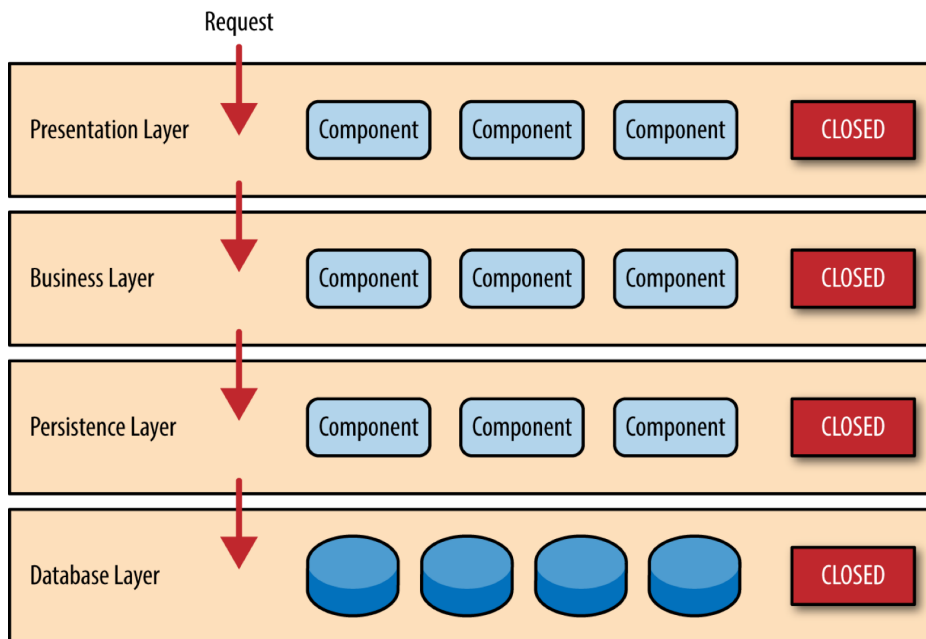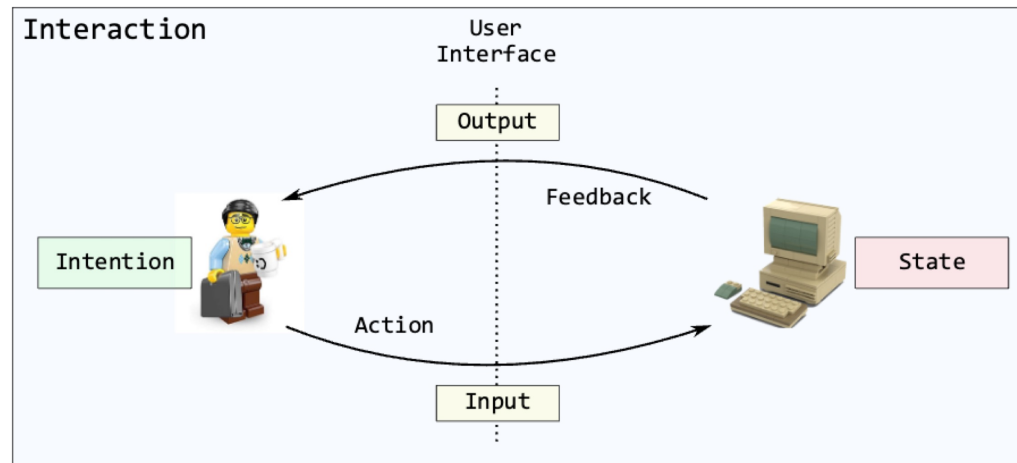| Presentation Layer | Component | Component | Component | CLOSED |
Business Layer	Component	Component	Component	CLOSED
Persistence Layer	Component	Component	Component	CLOSED
Database Layer				CLOSED

Advantages
- Low coupling ("separation of concerns")
- High testability.
- Flexibility
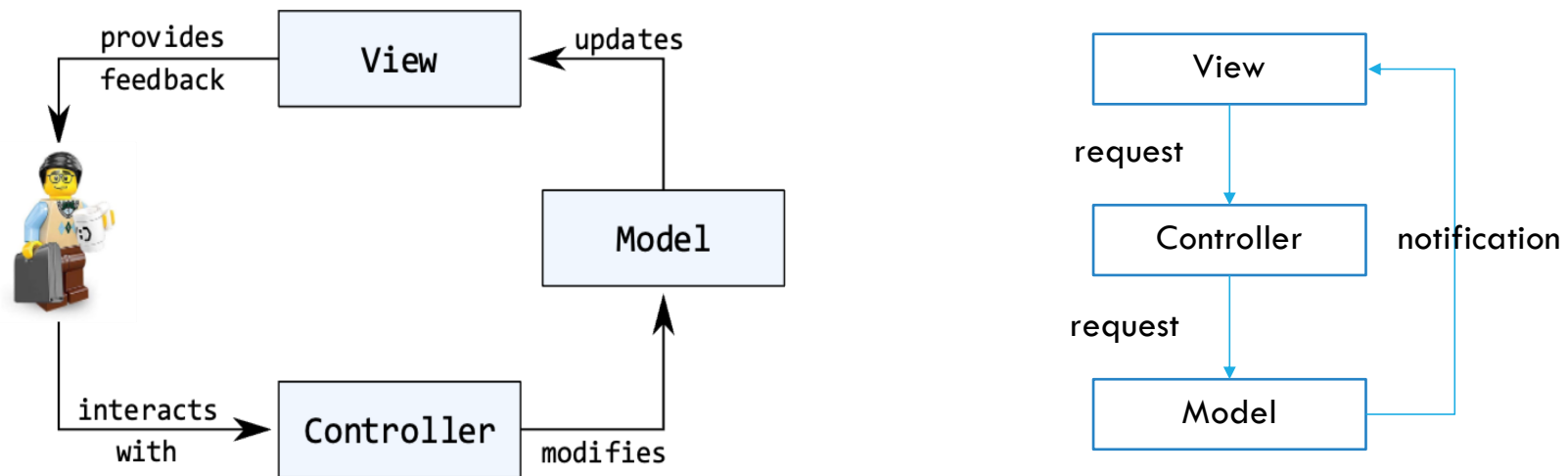- Ease of development.

Problems?
- How do you get feedback from lowest layers to highest layers?
- How do you model a complex UI?

1

INTERACTIVE APPLICATIONS?

User interaction is a feedback loop. How do we model this?

MODEL-VIEW-CONTROLLER (MVC) PATTERN

MVC is sort-of a layered architecture. Requests go down, notifications come back up.

MVVM

Adds a
ViewModel
between the
Model and View

View

-Controller controller
-ViewModel viewModel

«Interface»
ISubscriber

+update()

Controller

+Model model

+invoke(Event)

ViewModel

-View view
-Model model

+update()

«Interface»
IPublisher

-List<Subscriber> subscribers

+notify()

Model

+var data

+subscribe(ISubscriber)
+unsubscribe(ISubscriber)