

MVVM

• Model-View-ViewModel

A “code organizing” architectural design paradigm.

Must be adhered to for SwiftUI to work.

It is different from MVC (Model View Controller) that UIKit (old-style iOS) uses.



MVVM

Model

View



MVVM

Model

UI Independent

View



MVVM

Model

UI Independent
Data + Logic

View



MVVM

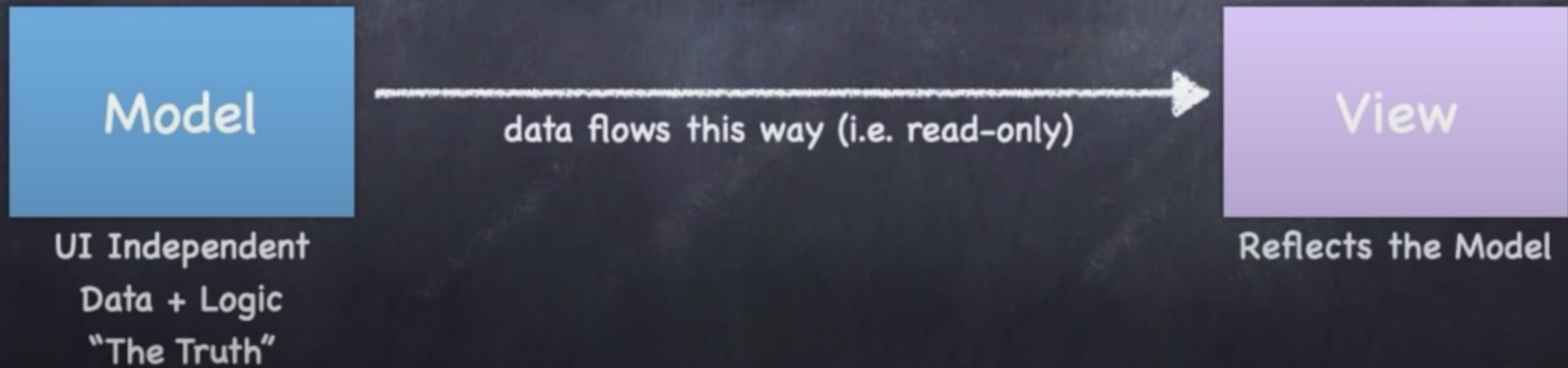
Model

UI Independent
Data + Logic
"The Truth"

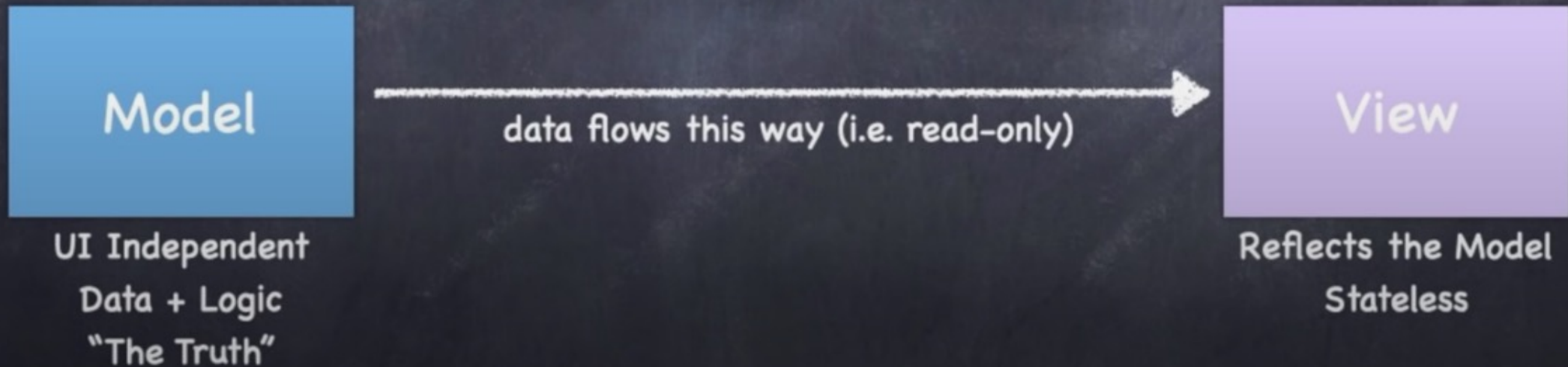
View



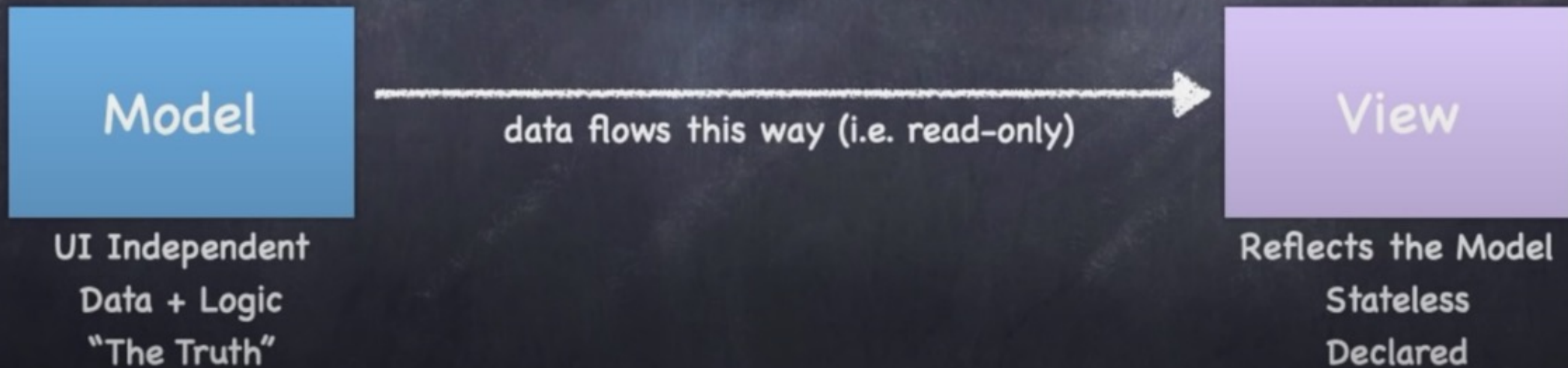
MVVM



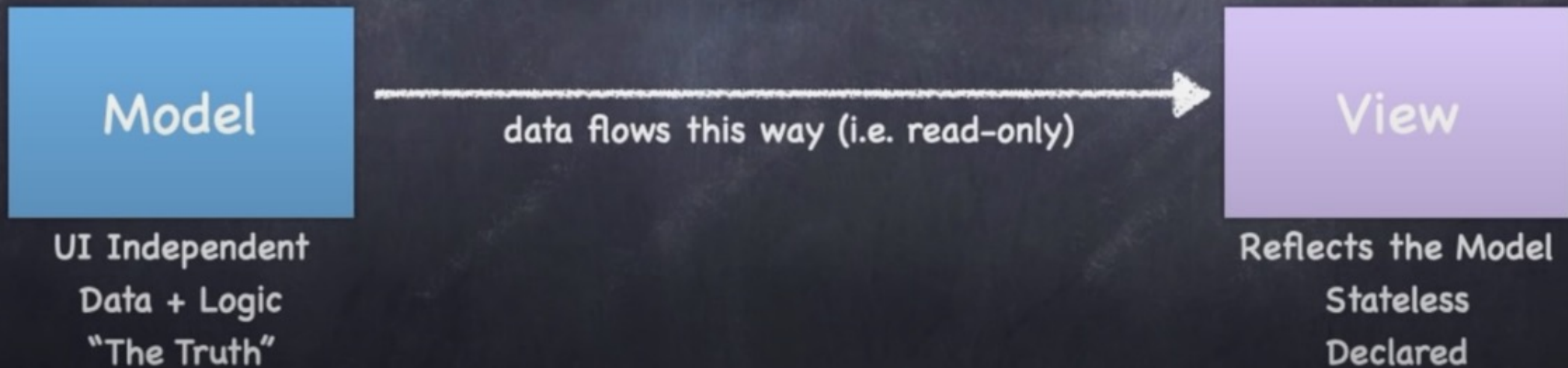
MVVM



MVVM



MVVM



MVVM

ViewModel

Model

UI Independent
Data + Logic
"The Truth"

data flows this way (i.e. read-only)

View

Reflects the Model
Stateless
Declared



MVVM

ViewModel

Binds View to Model

Model

UI Independent
Data + Logic
"The Truth"

data flows this way (i.e. read-only)

View

Reflects the Model
Stateless
Declared



MVVM

ViewModel

Binds View to Model
Interpreter

Model

UI Independent
Data + Logic
"The Truth"

data flows this way (i.e. read-only)

View

Reflects the Model
Stateless
Declared



MVVM

ViewModel

Binds View to Model
Interpreter
Gatekeeper

Model

UI Independent
Data + Logic
"The Truth"

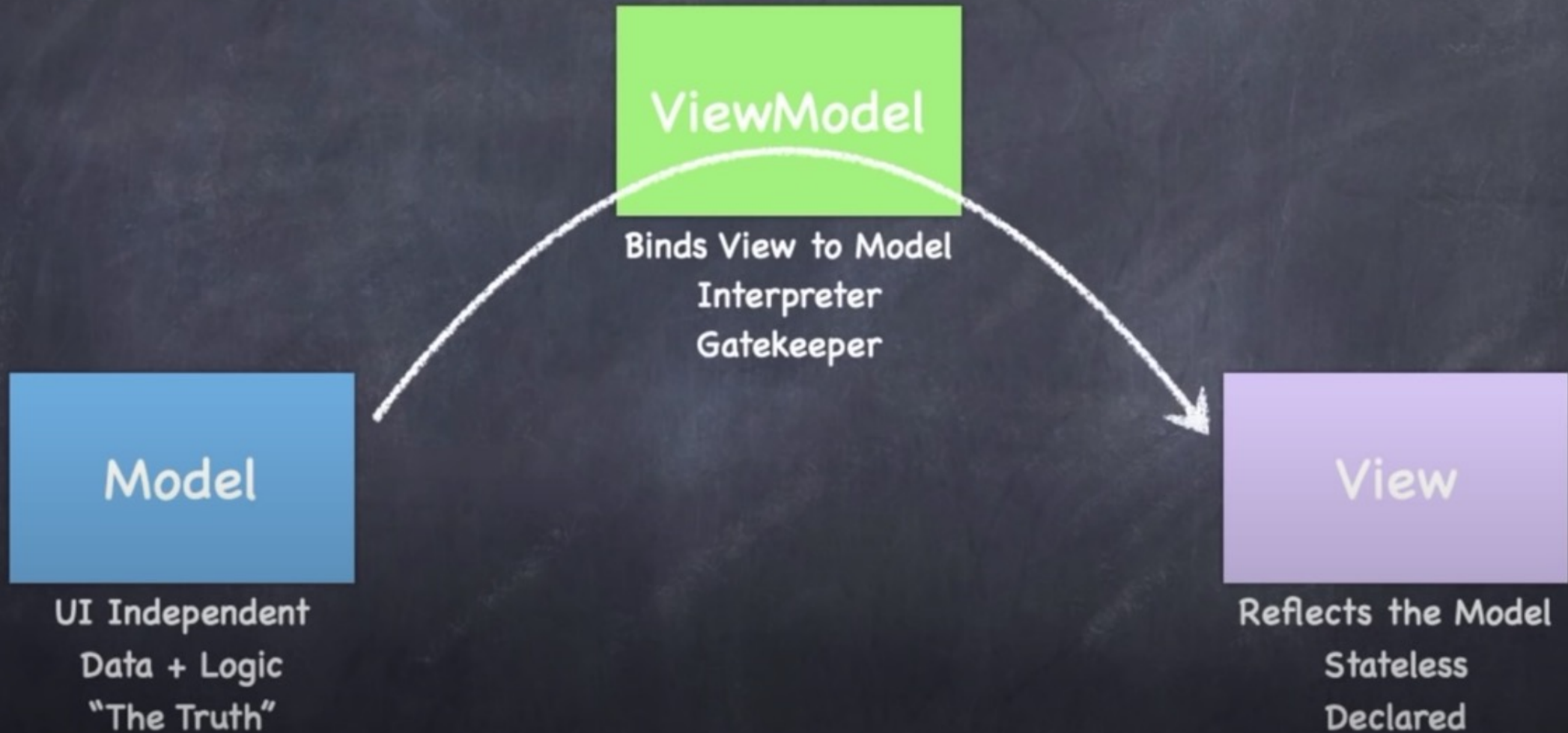
data flows this way (i.e. read-only)

View

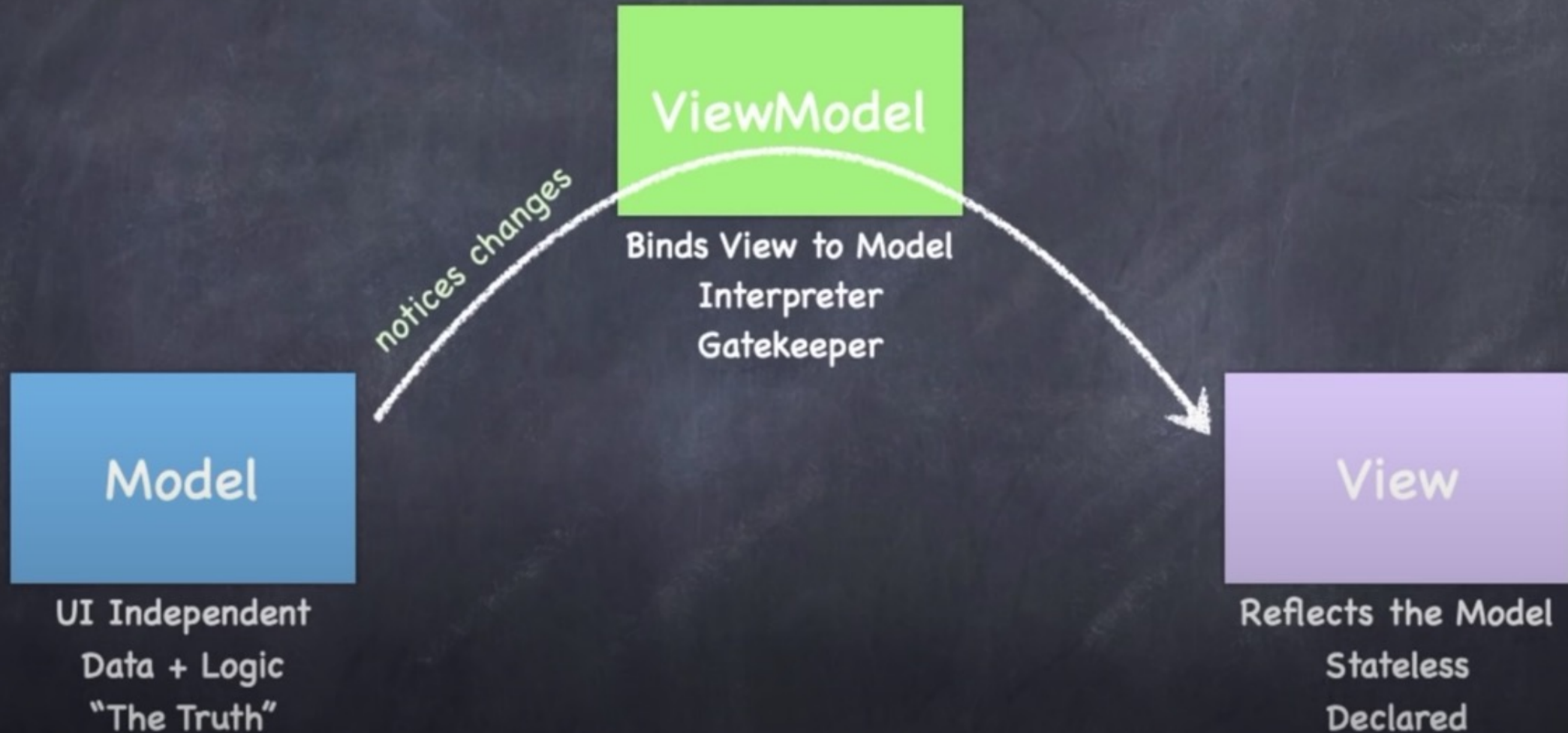
Reflects the Model
Stateless
Declared



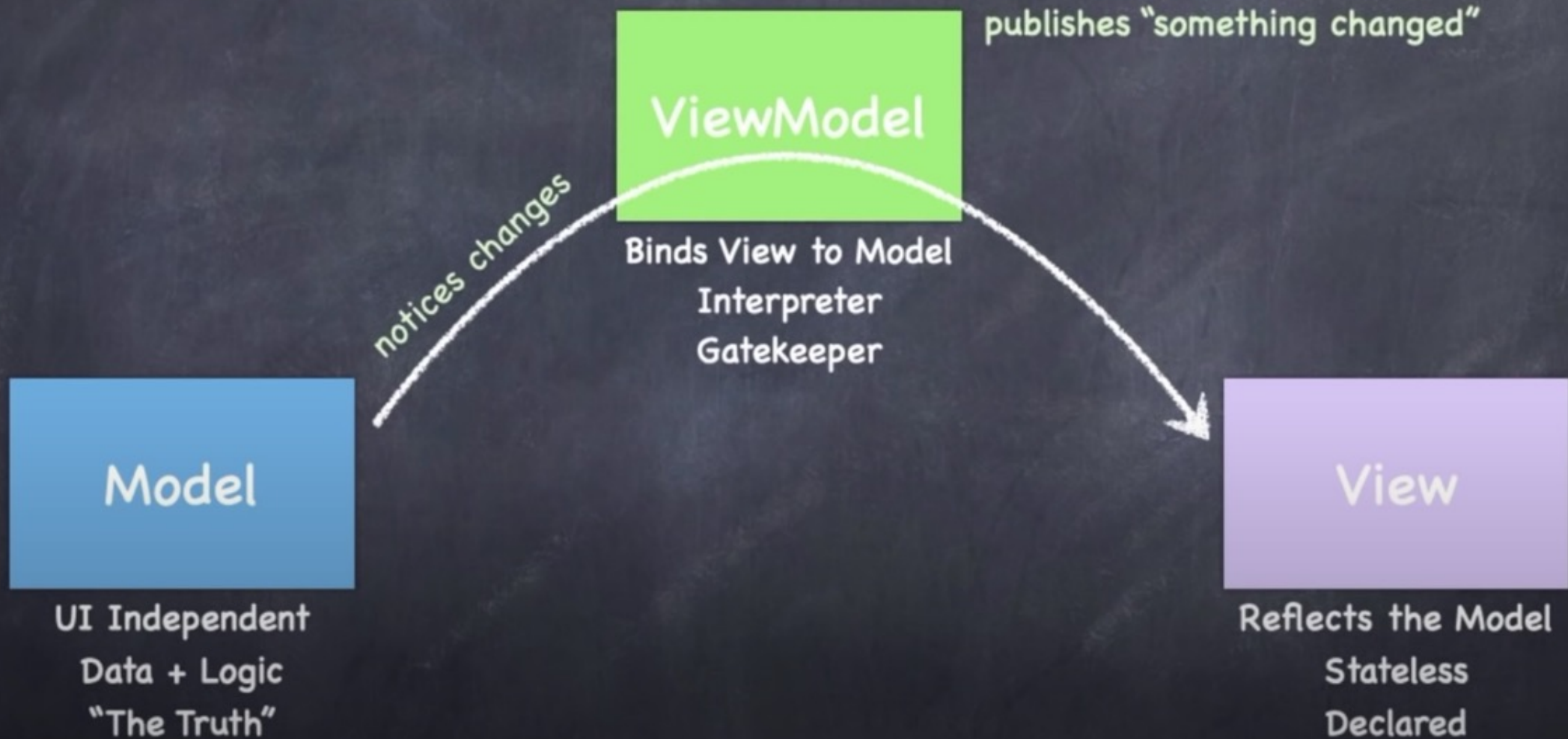
MVVM



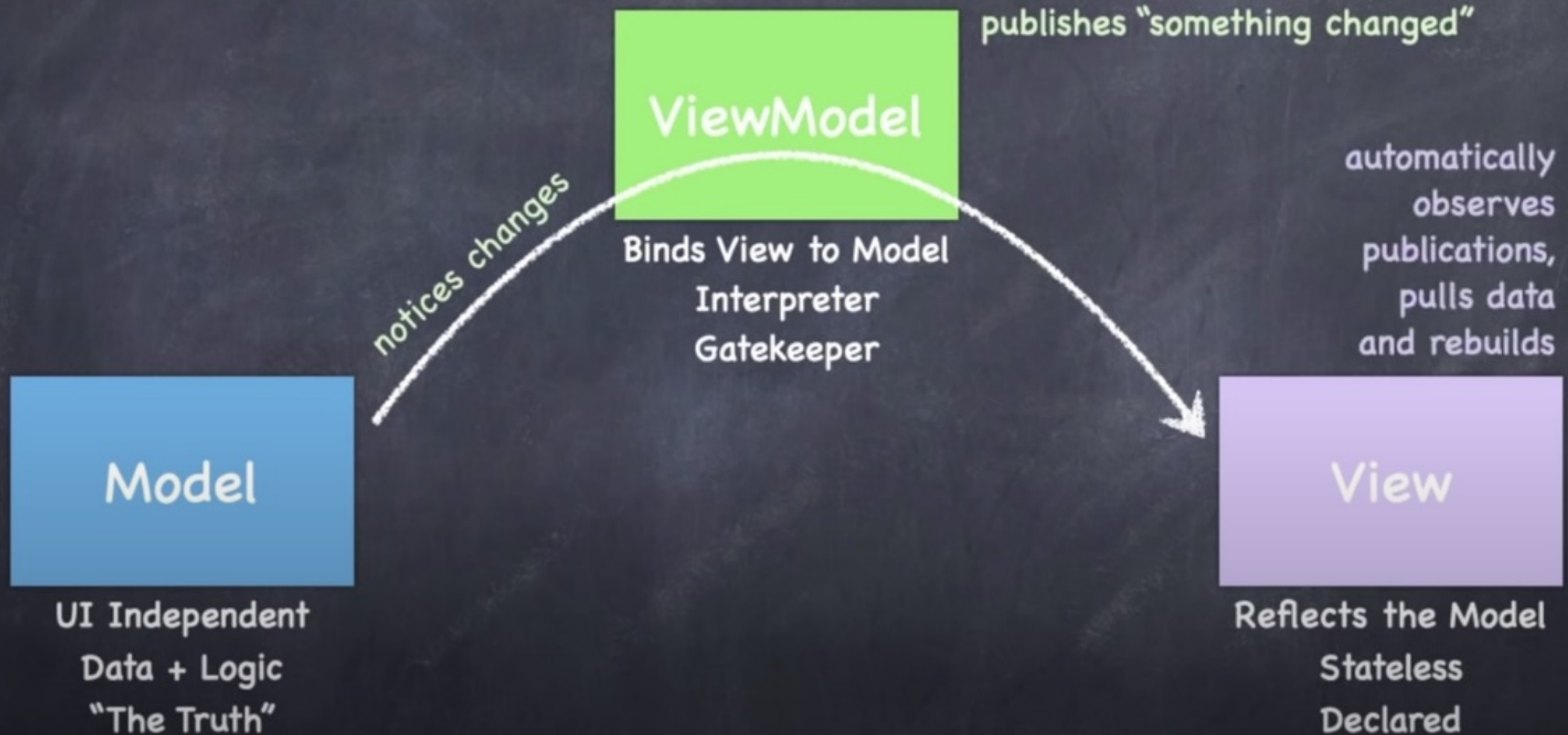
MVVM



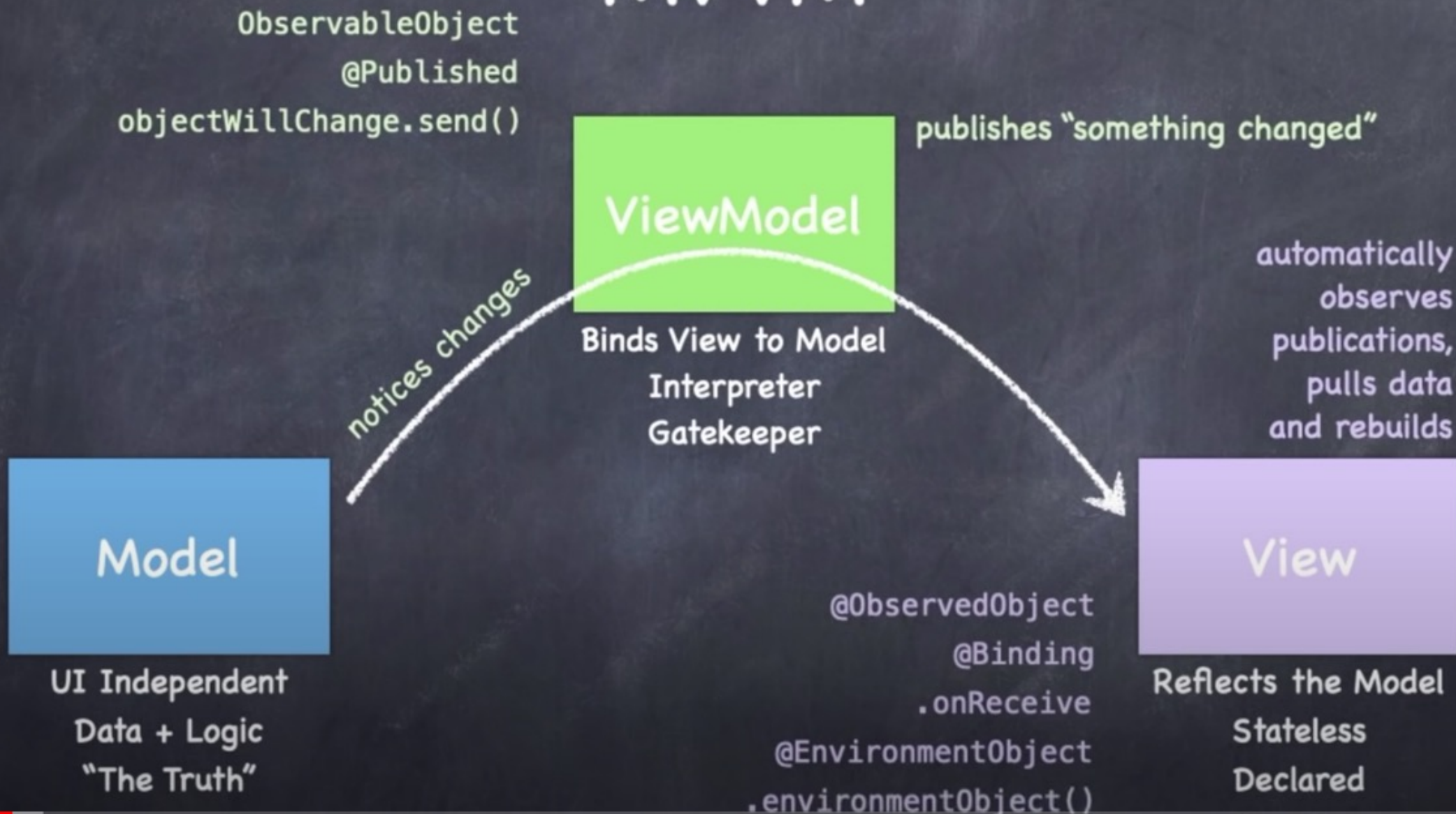
MVVM



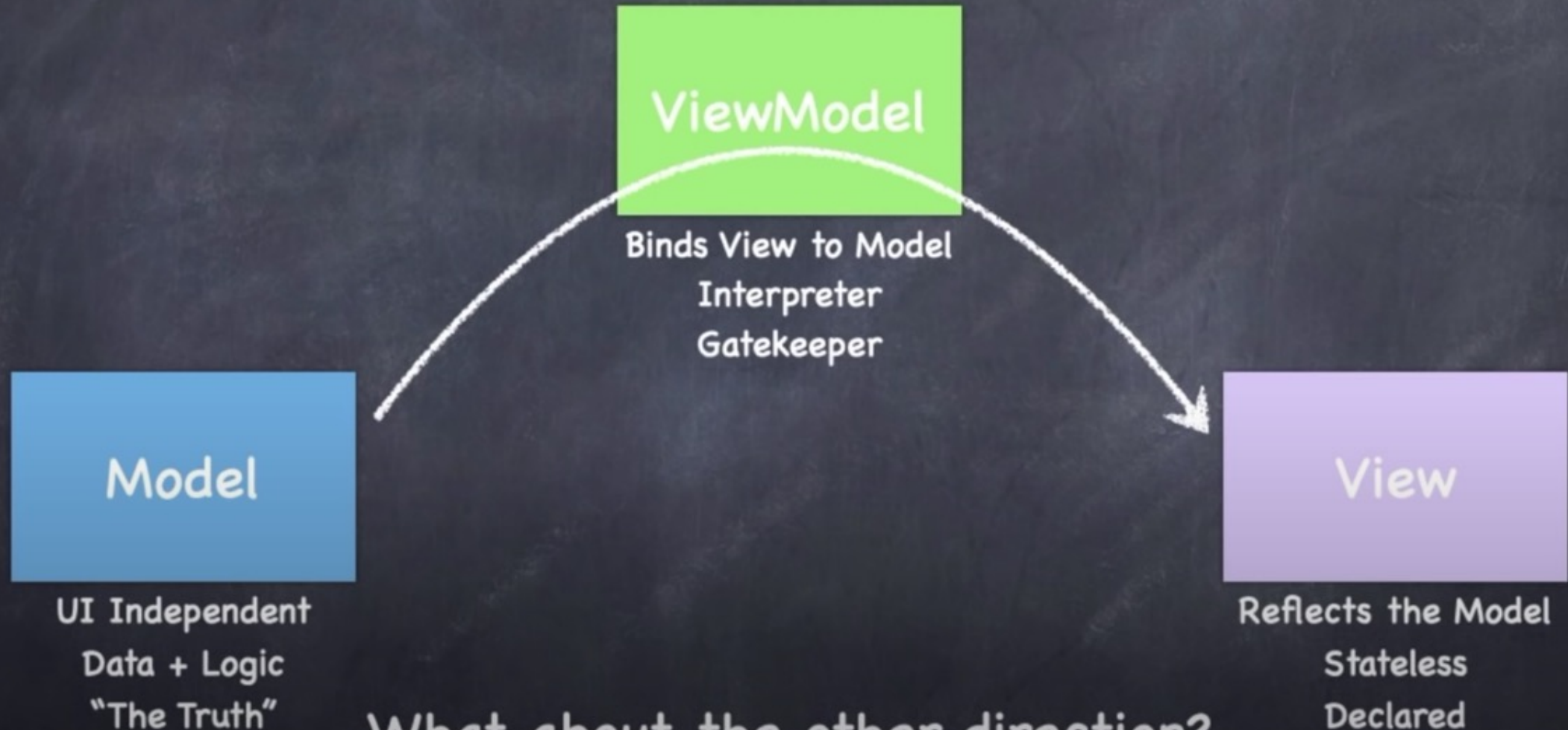
MVVM



MVVM



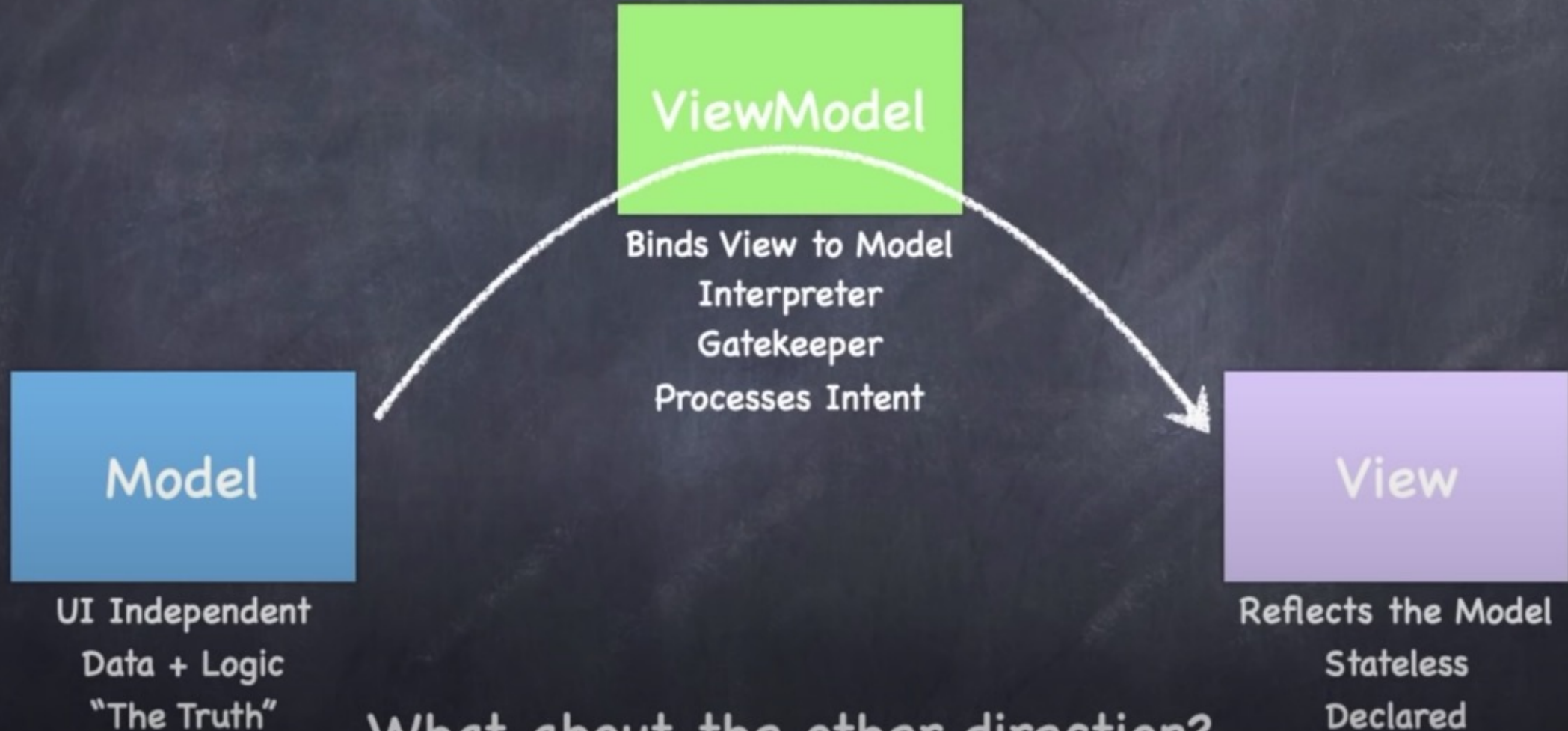
MVVM



What about the other direction?



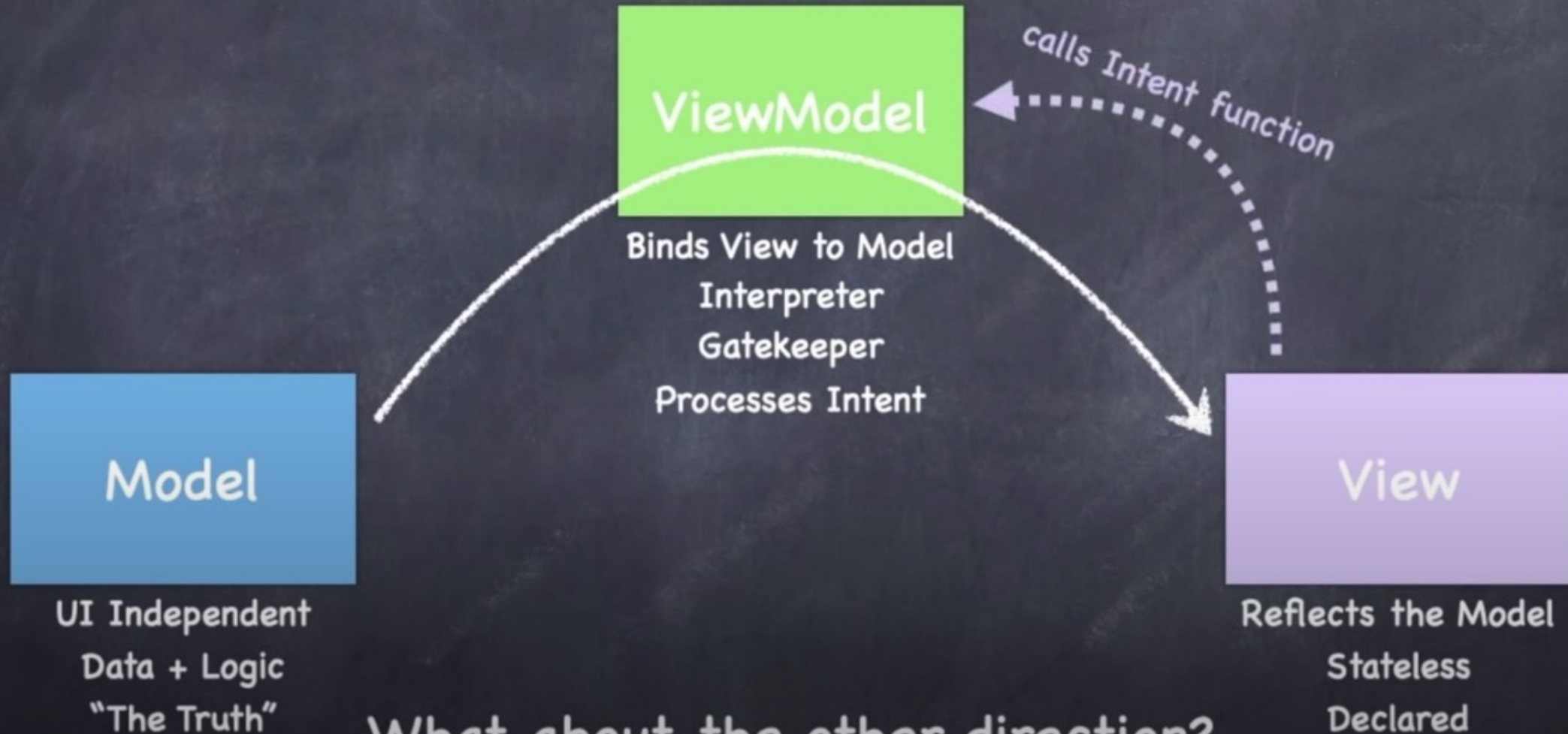
MVVM



What about the other direction?



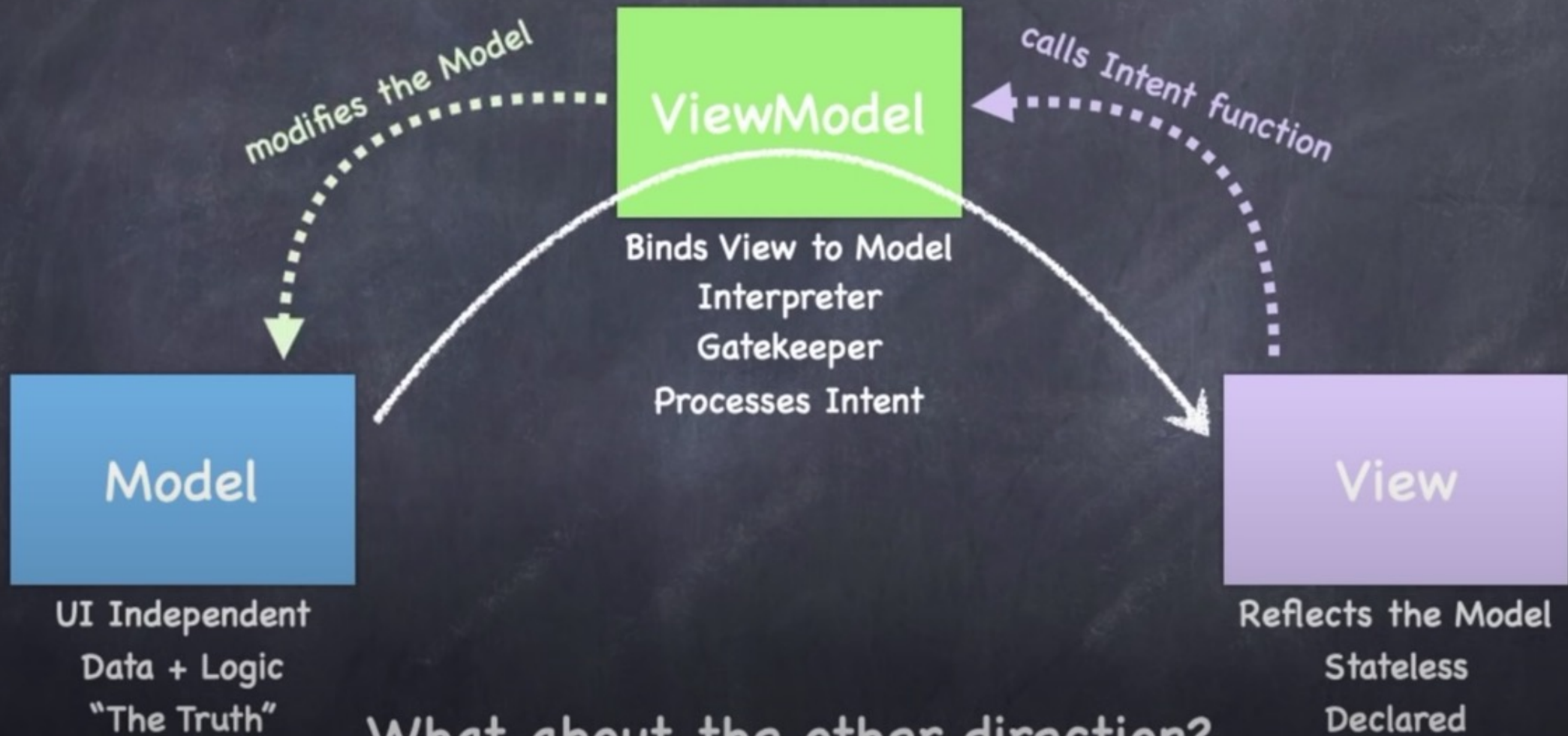
MVVM



What about the other direction?



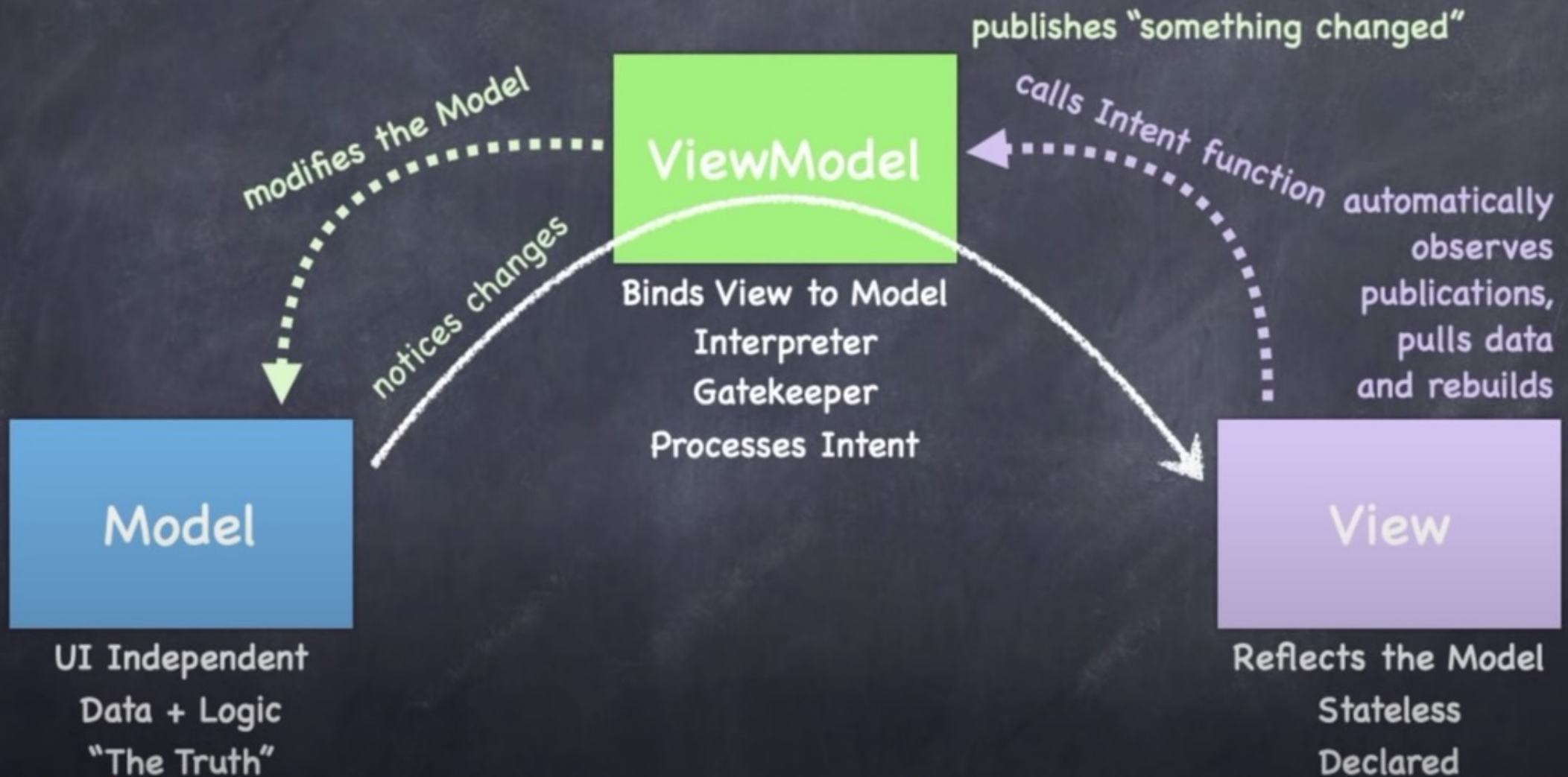
MVVM



What about the other direction?



MVVM



MVVM

ObservableObject
@Published
objectWillChange.send()

