Control JavaFX with the Presentation Model Pattern

S313995

Dierk König

dierk.koenig@canoo.com
@mittie

Control any* GUI with the Presentation Model Pattern

S313995

Dierk König

dierk.koenig@canoo.com
@mittie

Dierk König

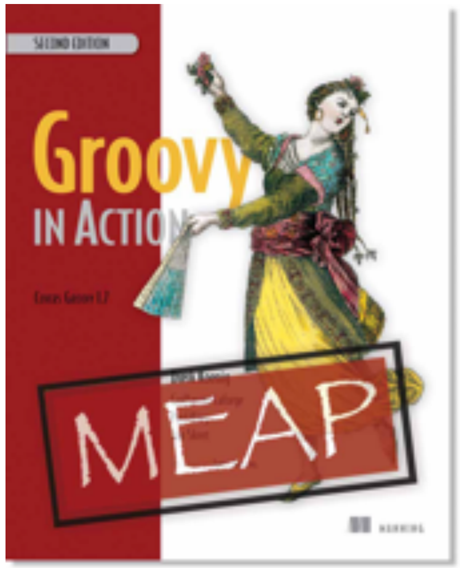
Fellow @ Canoo Engineering AG, Basel (CH)

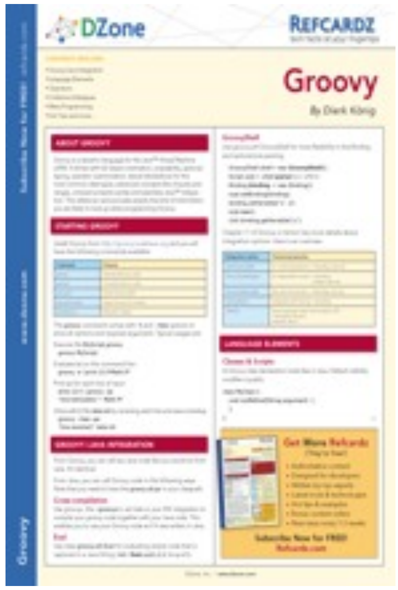
Canoo RIA Suite

Projects, Consulting

www.canoo.com

Open-source committer Groovy, Grails, GPars, Grasp





Structure

- Context of the Presentation Model Pattern
- Problem description by example
- Forces
- Solution idea
- Implementation example with Grasp
- Resulting context - future options

Context of the Presentation Model Pattern

- Applicable to any event-based GUI toolkit, e.g.

  Swing, JavaFX, AWT, SWT, and their complements

  Flex

  Windows Presentation Foundation

  Server side: GWT(?), Canoo RIA Suite

Context of the Presentation Model Pattern

- Applicable to any event-based GUI toolkit, e.g.

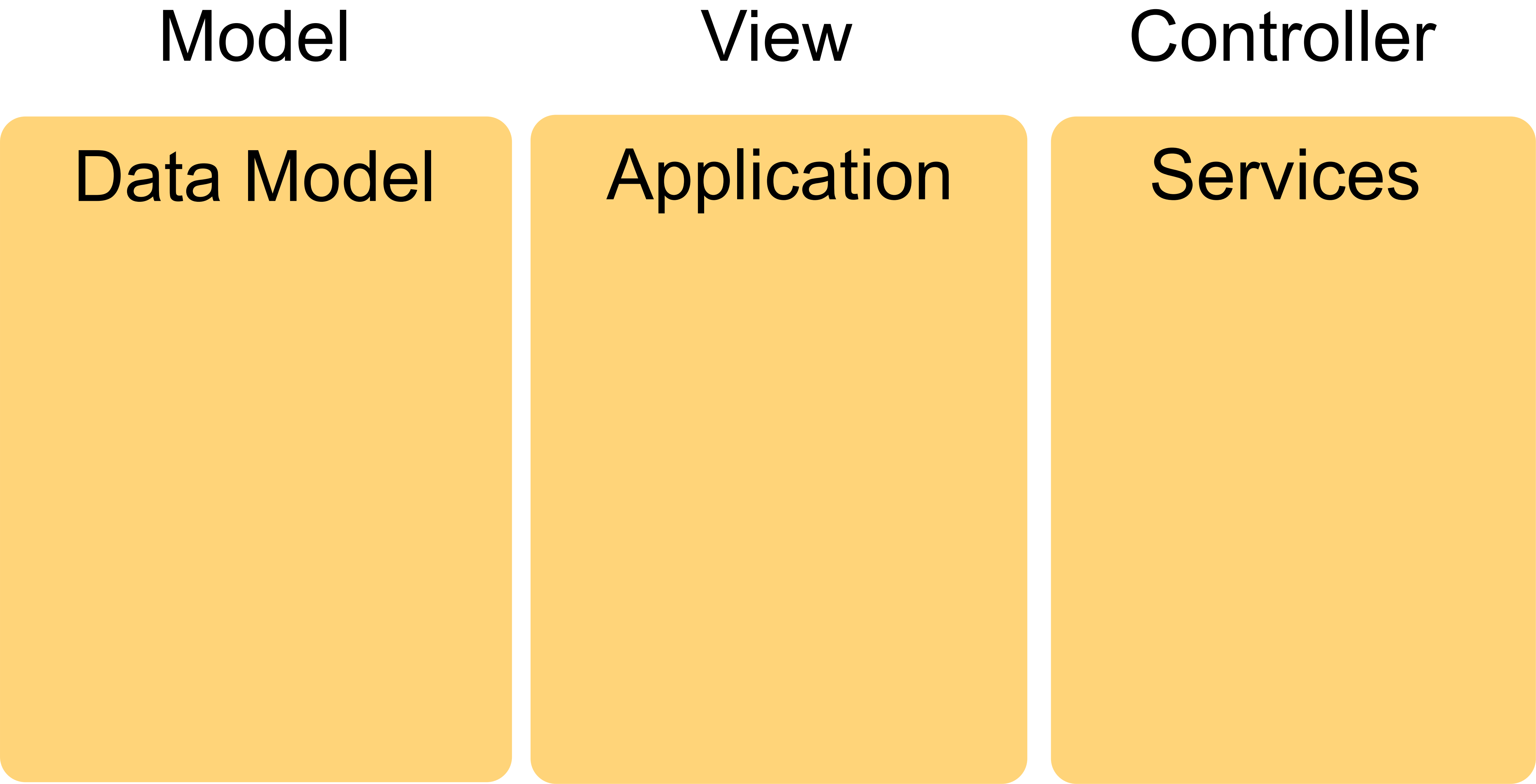
  Swing, JavaFX, AWT, SWT, and their complements

  Flex

  Windows Presentation Foundation

  Server side: GWT(?), Canoo RIA Suite

Model View Controller



Model

View

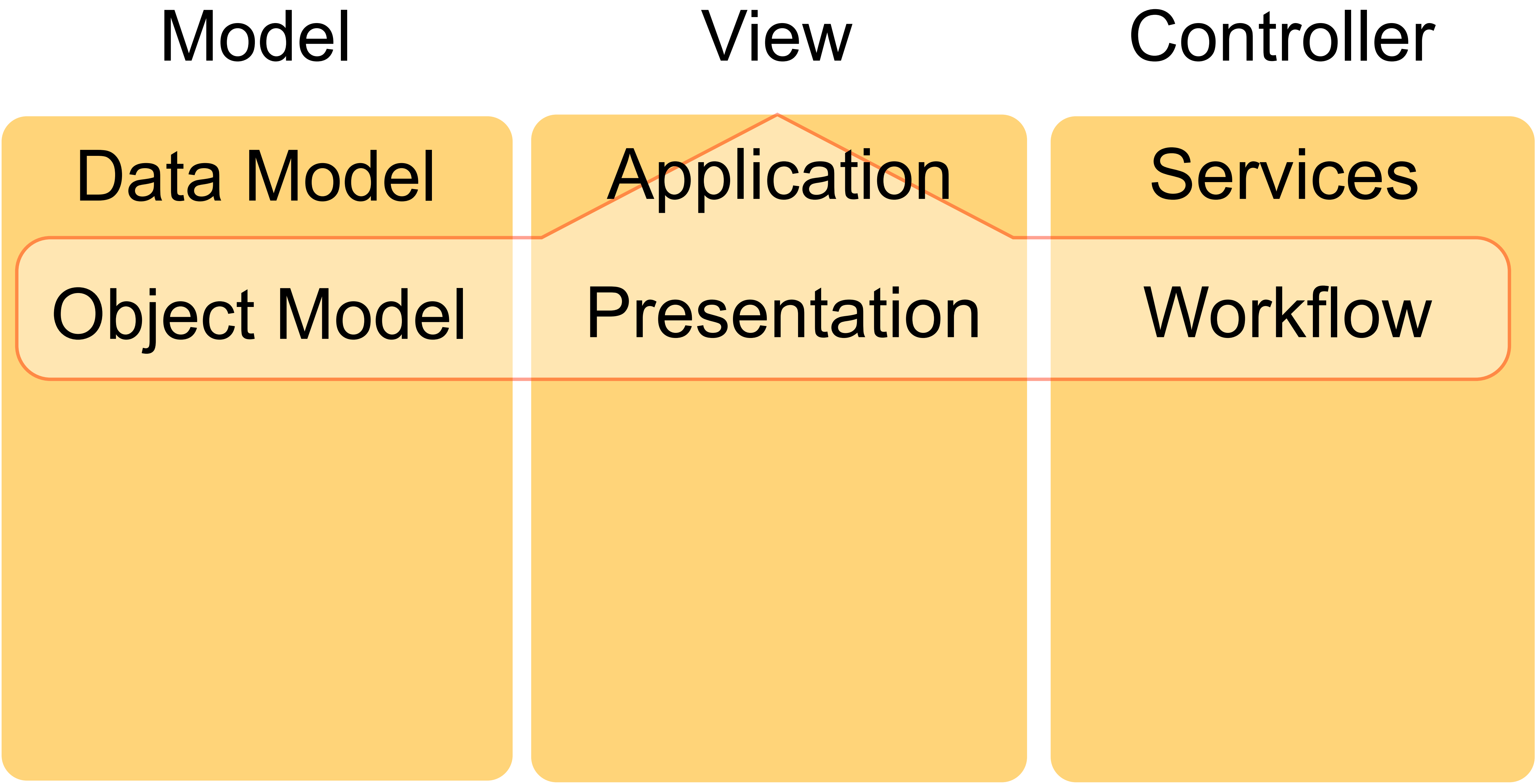
Controller

Data Model

Application

Services

Model	View	Controller
Data Model	Application	Services



Model

View

Controller

Data Model

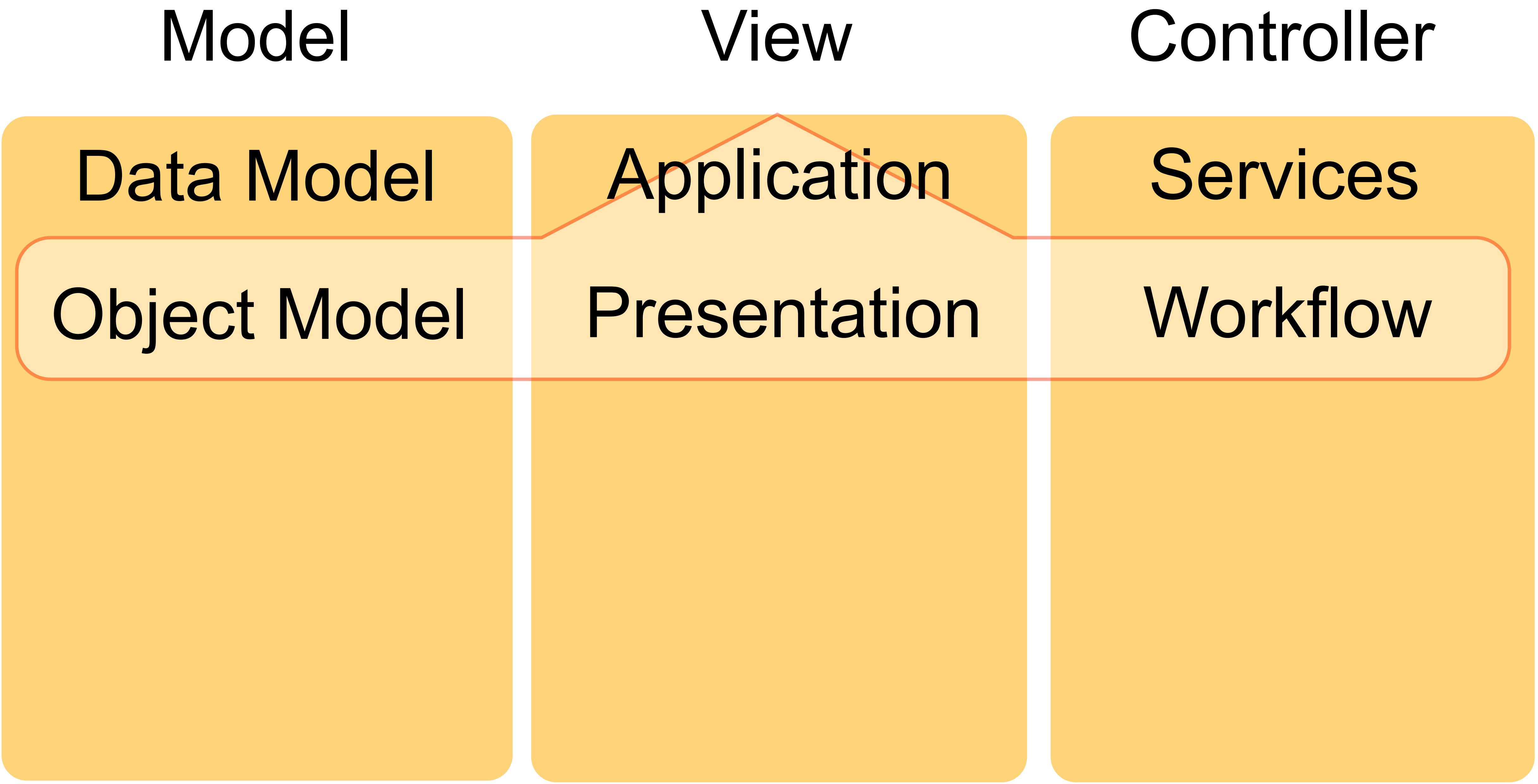
Application

Services

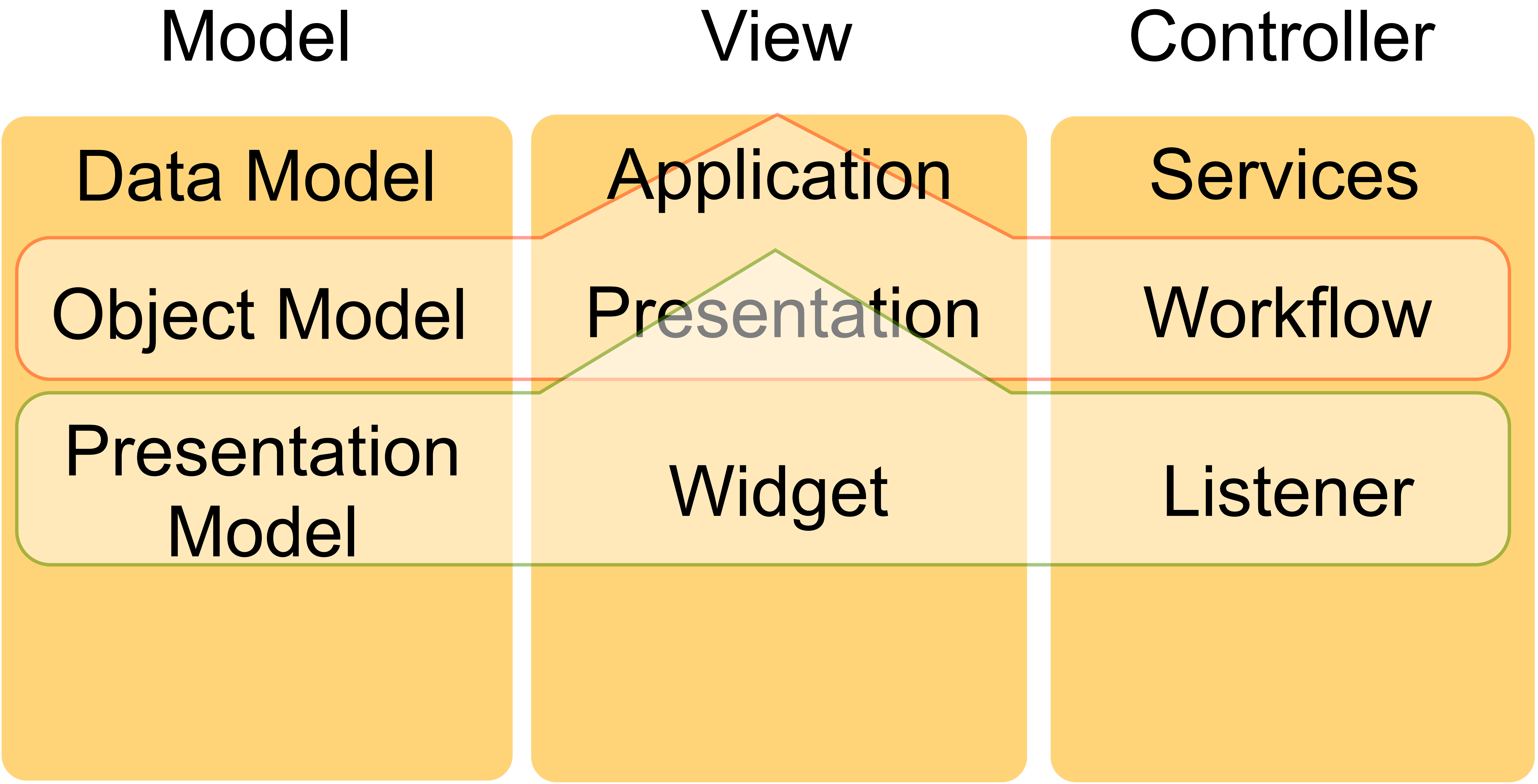
Object Model

Presentation

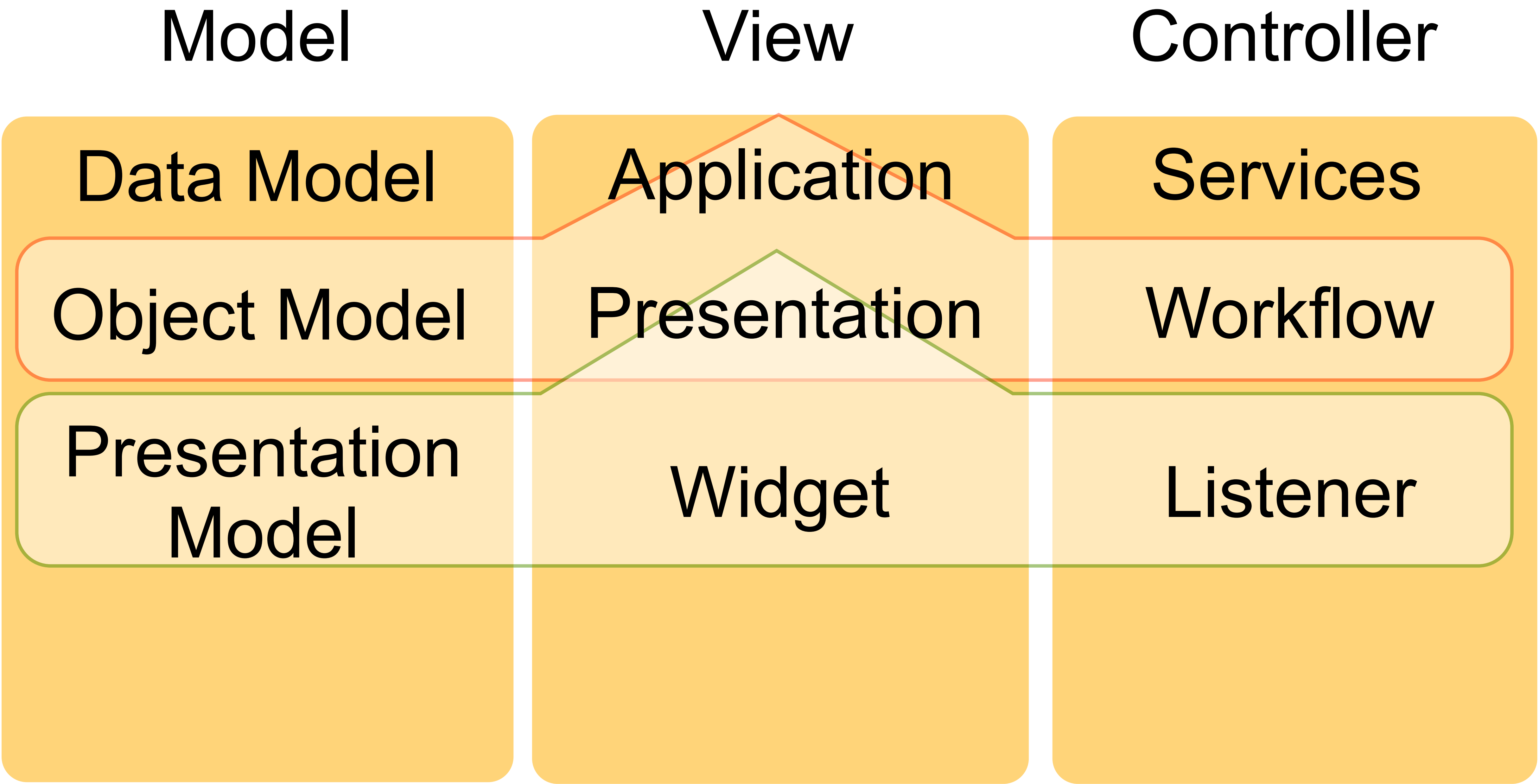
Workflow



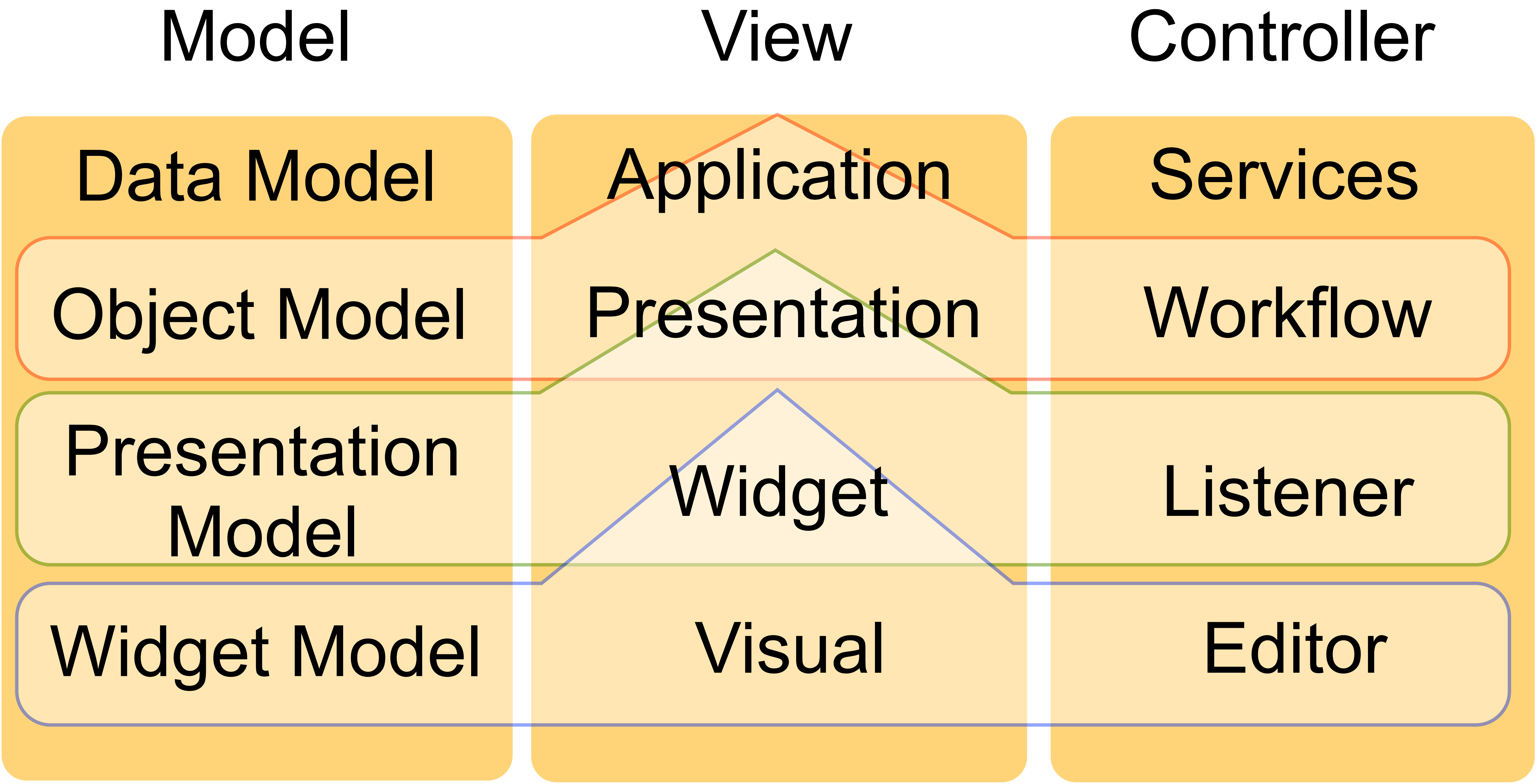
Model	View	Controller
Data Model	Application	Services
Object Model	Presentation	Workflow



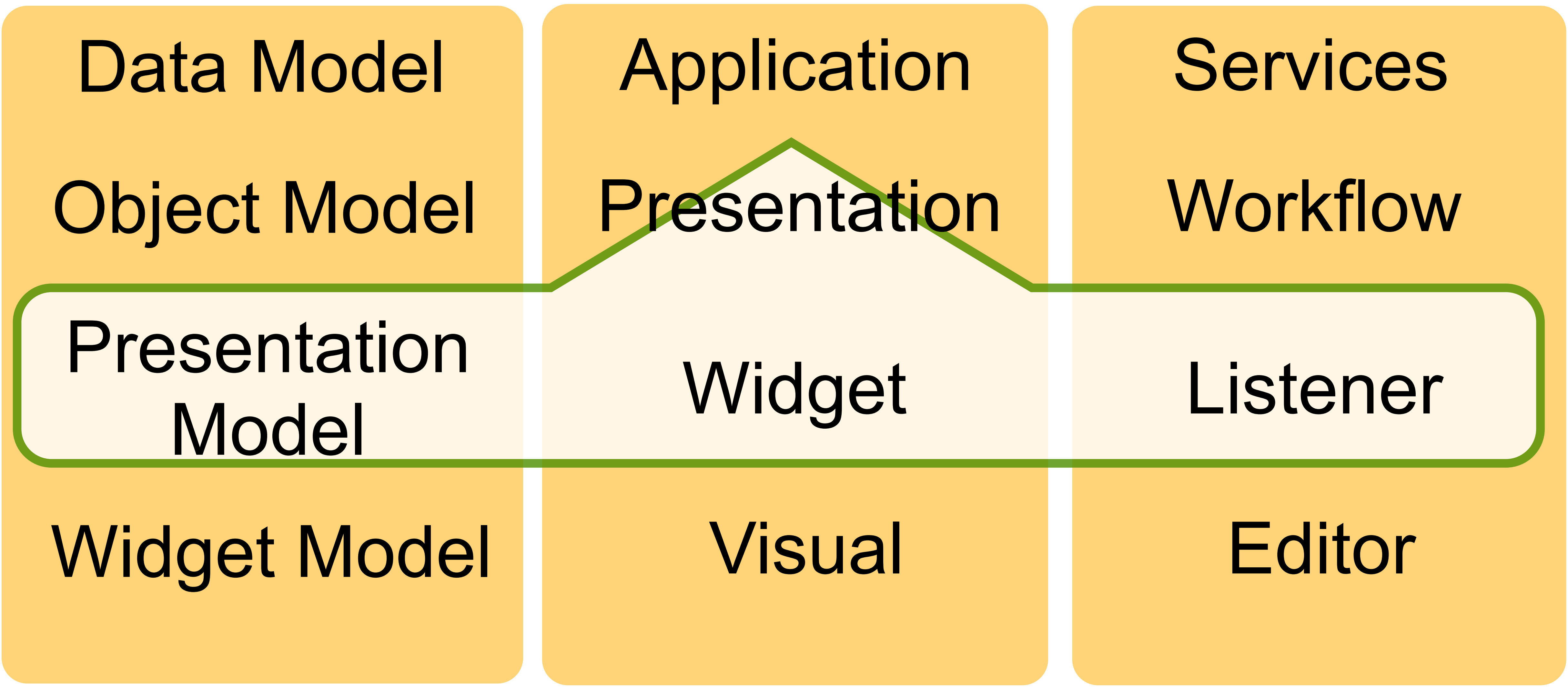
Model	View	Controller
Data Model	Application	Services
Object Model	Presentation	Workflow
Presentation Model	Widget	Listener



Model	View	Controller
Data Model	Application	Services
Object Model	Presentation	Workflow
Presentation Model	Widget	Listener

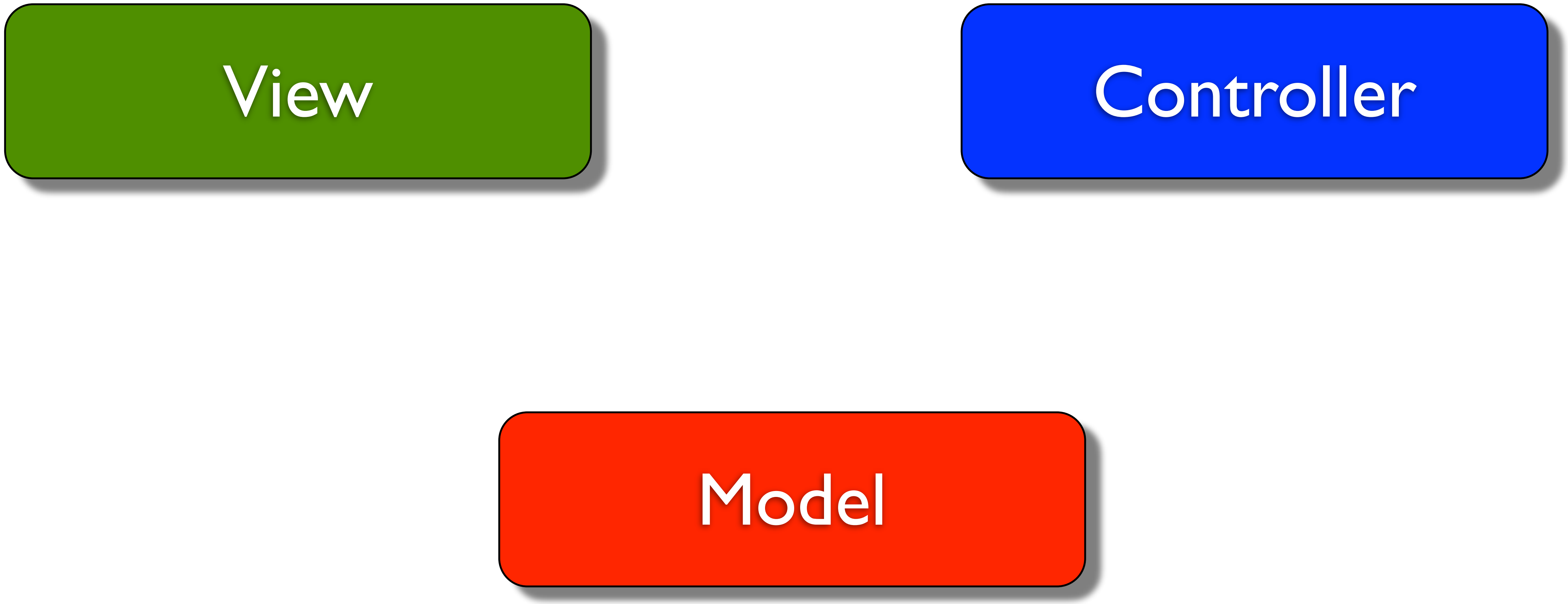


Model	View	Controller
Data Model	Application	Services
Object Model	Presentation	Workflow
Presentation Model	Widget	Listener
Widget Model	Visual	Editor



Model	View	Controller
Data Model	Application	Services
Object Model	Presentation	Workflow
Presentation Model	Widget	Listener
Widget Model	Visual	Editor

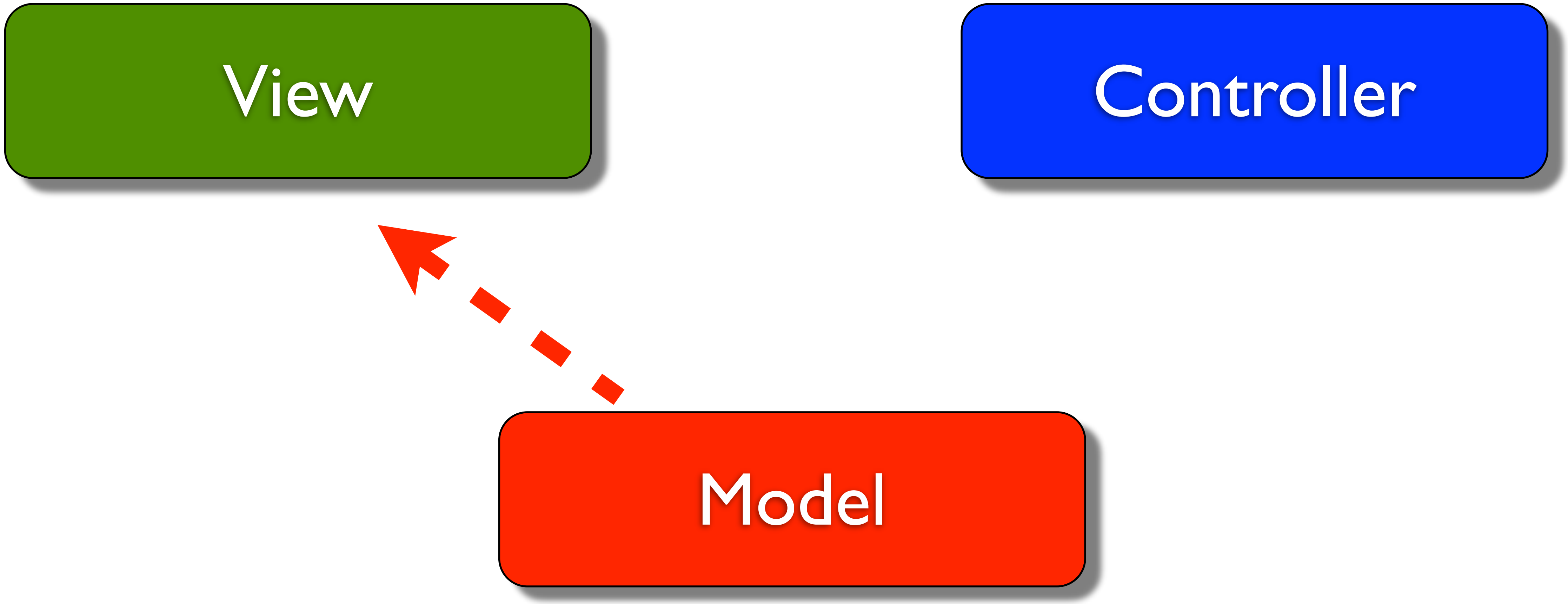
A simple demo



View

Controller

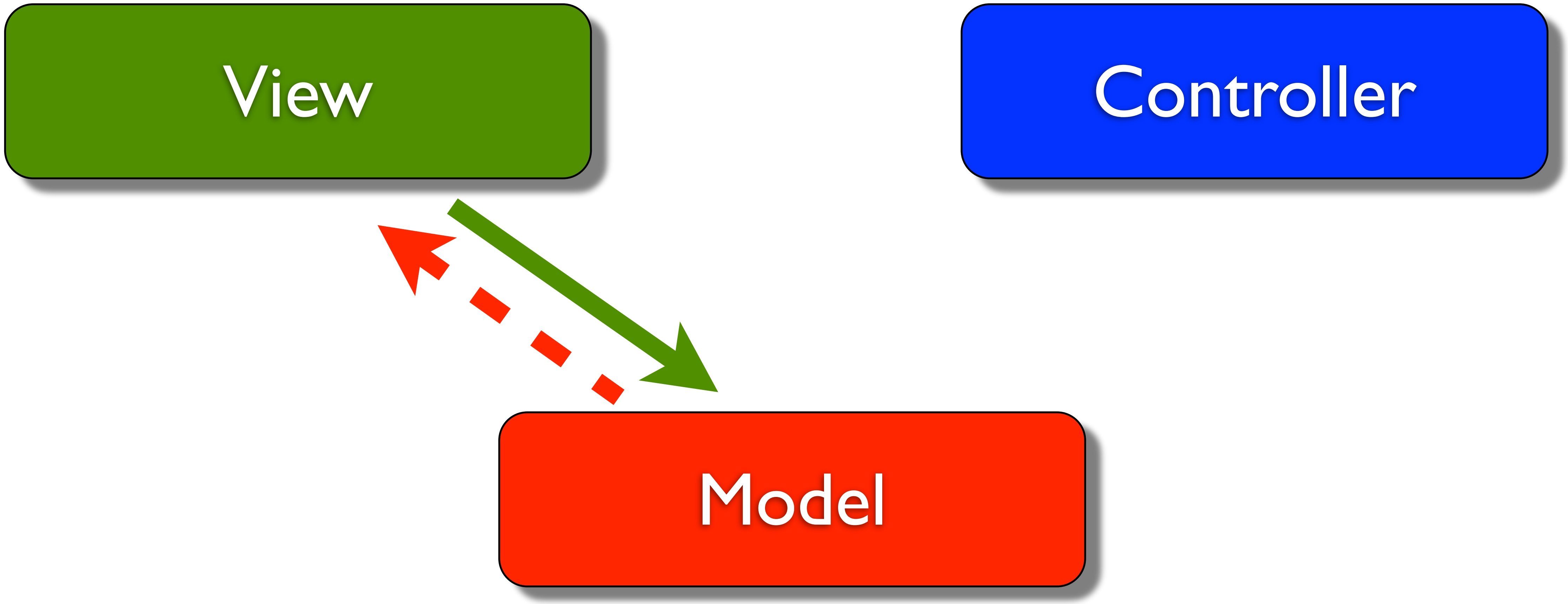
Model

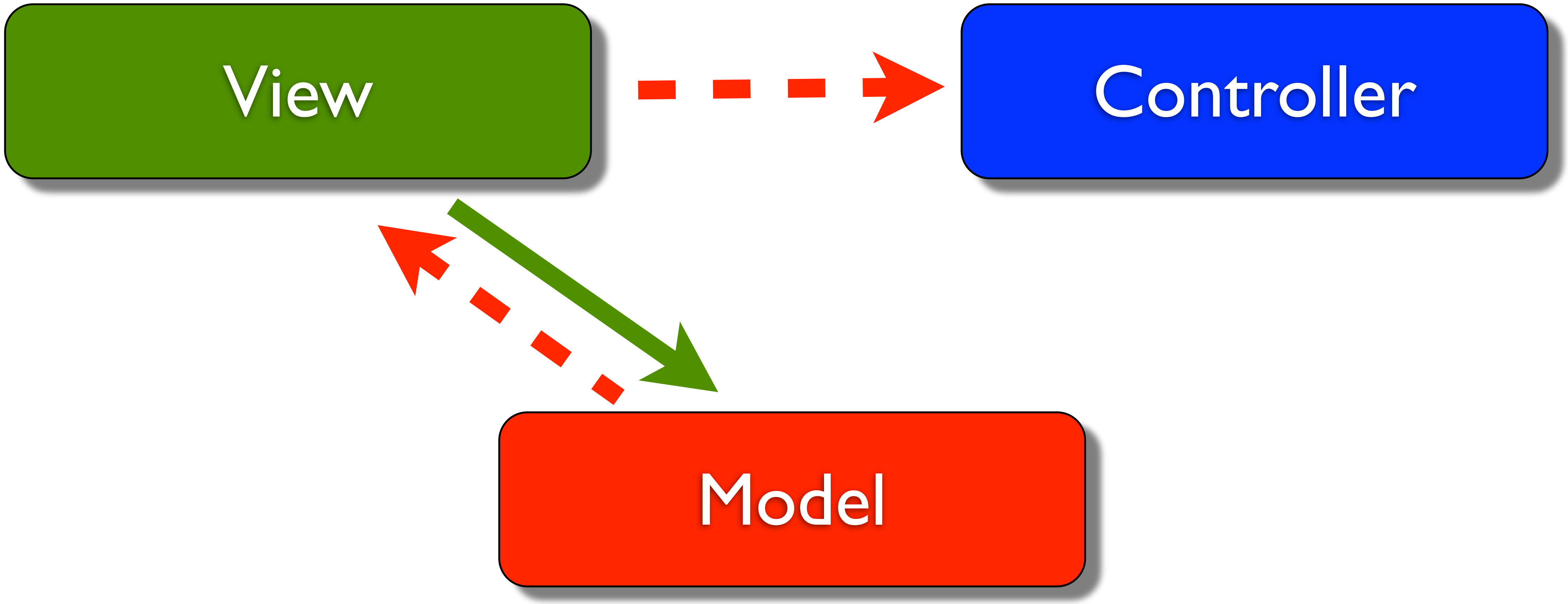


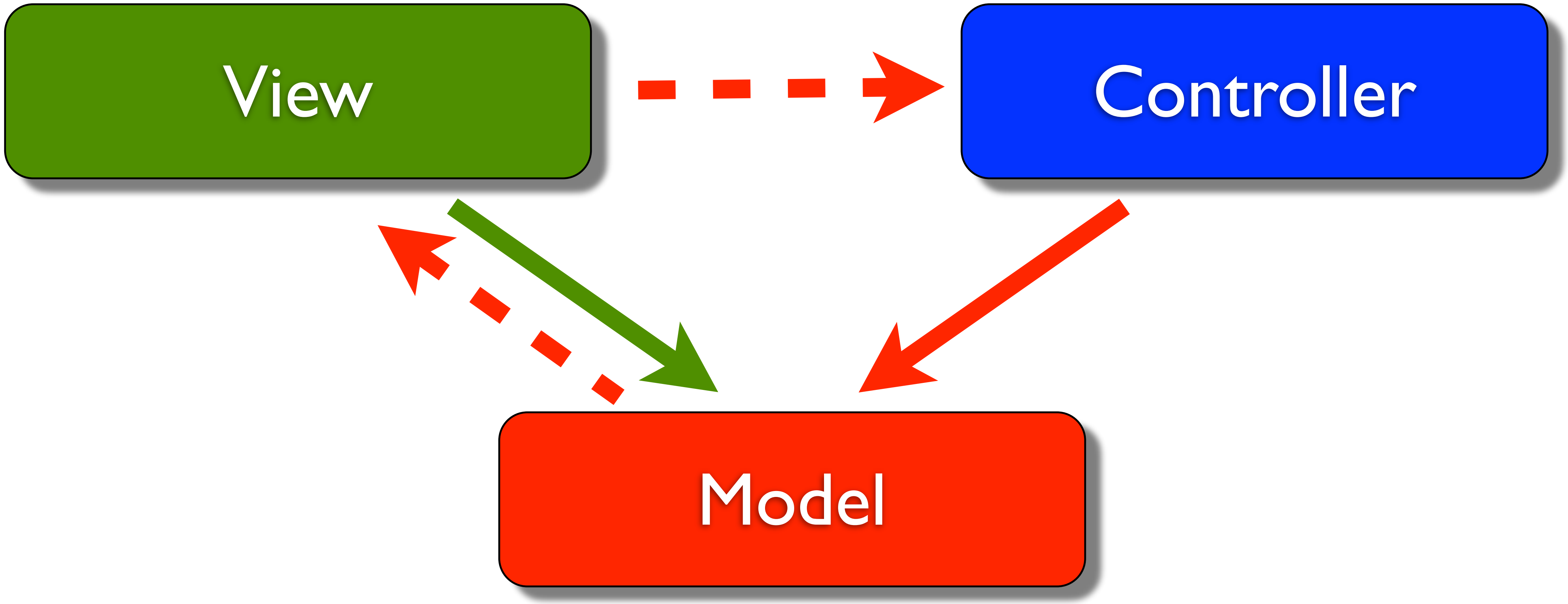
View

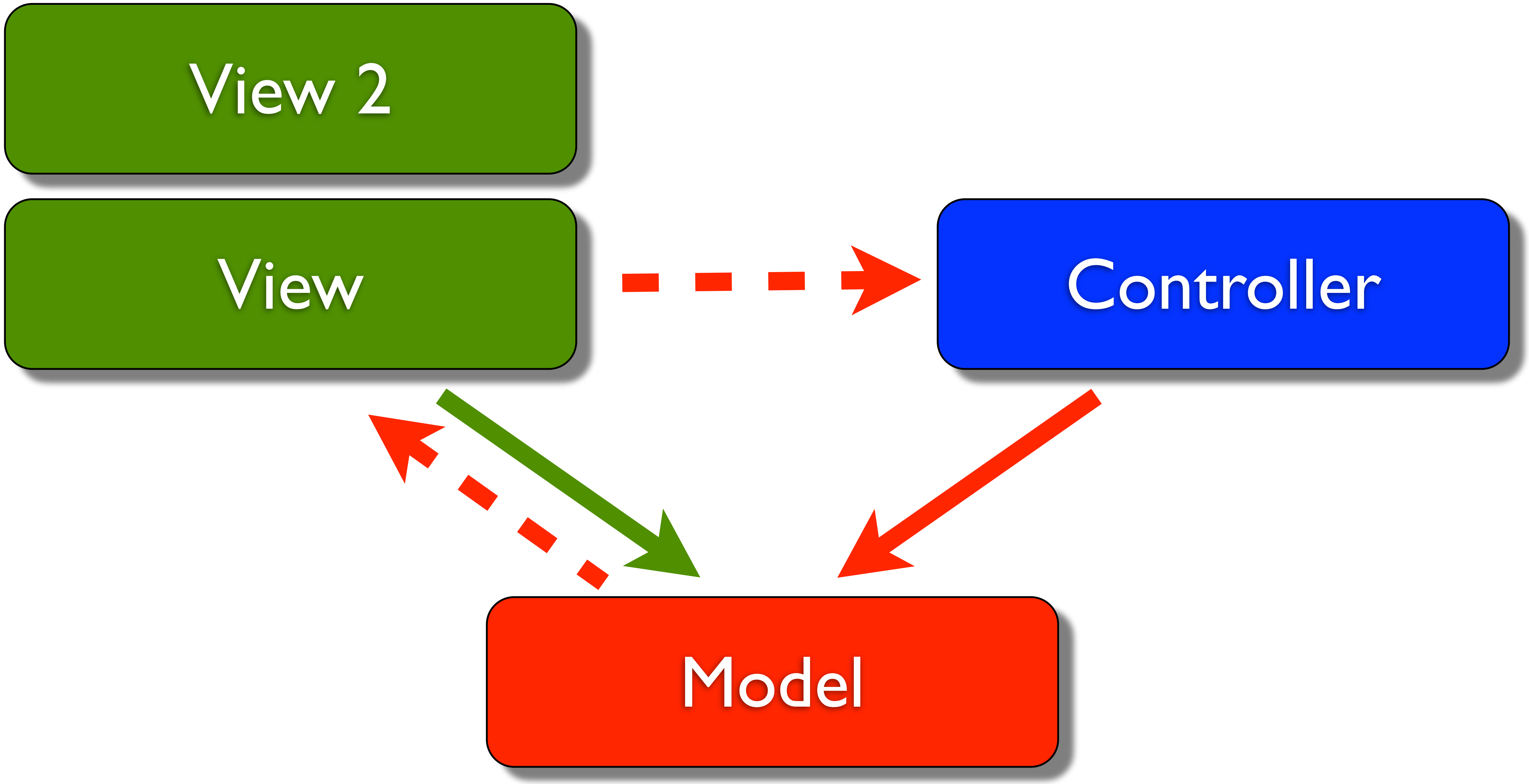
Controller

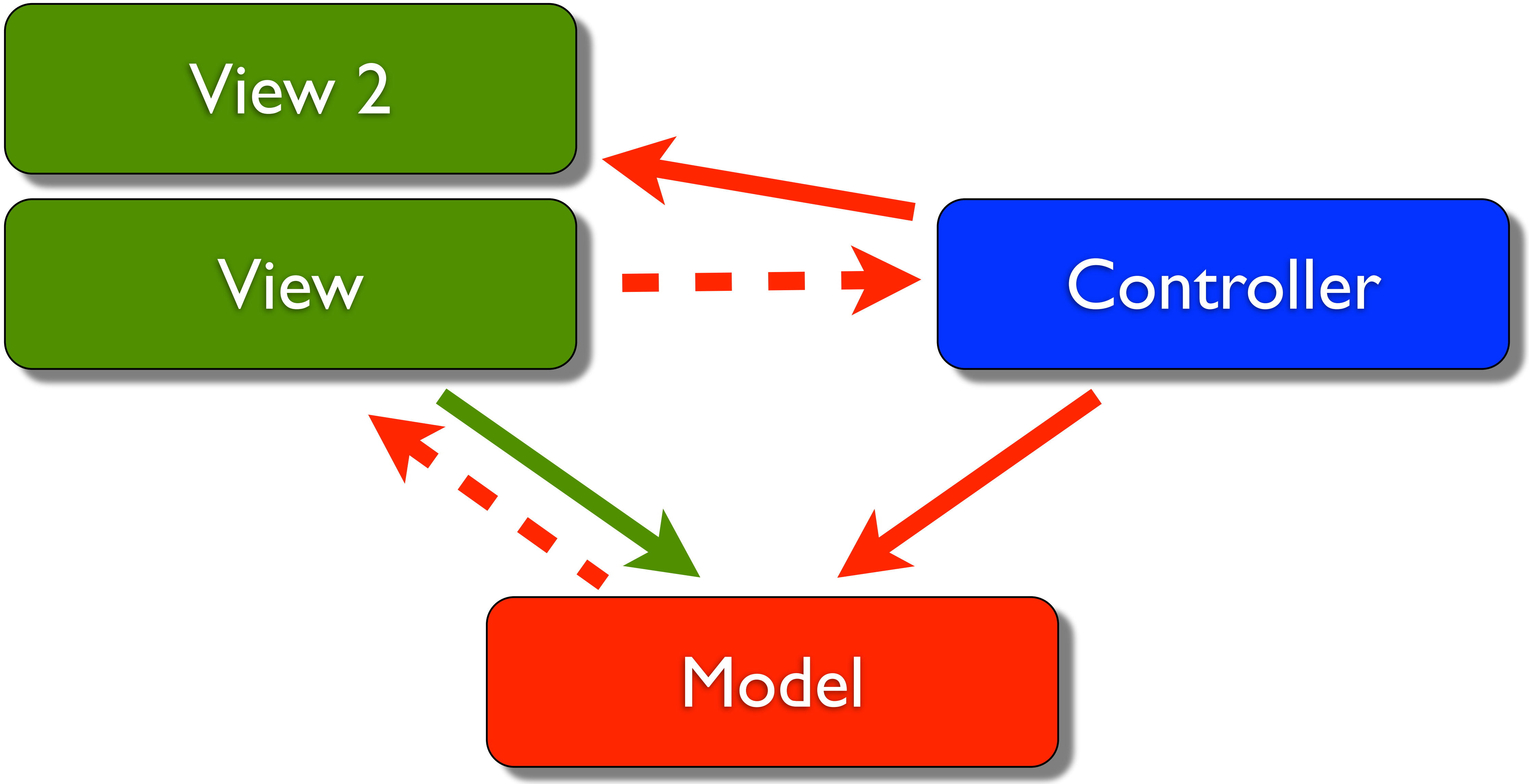
Model

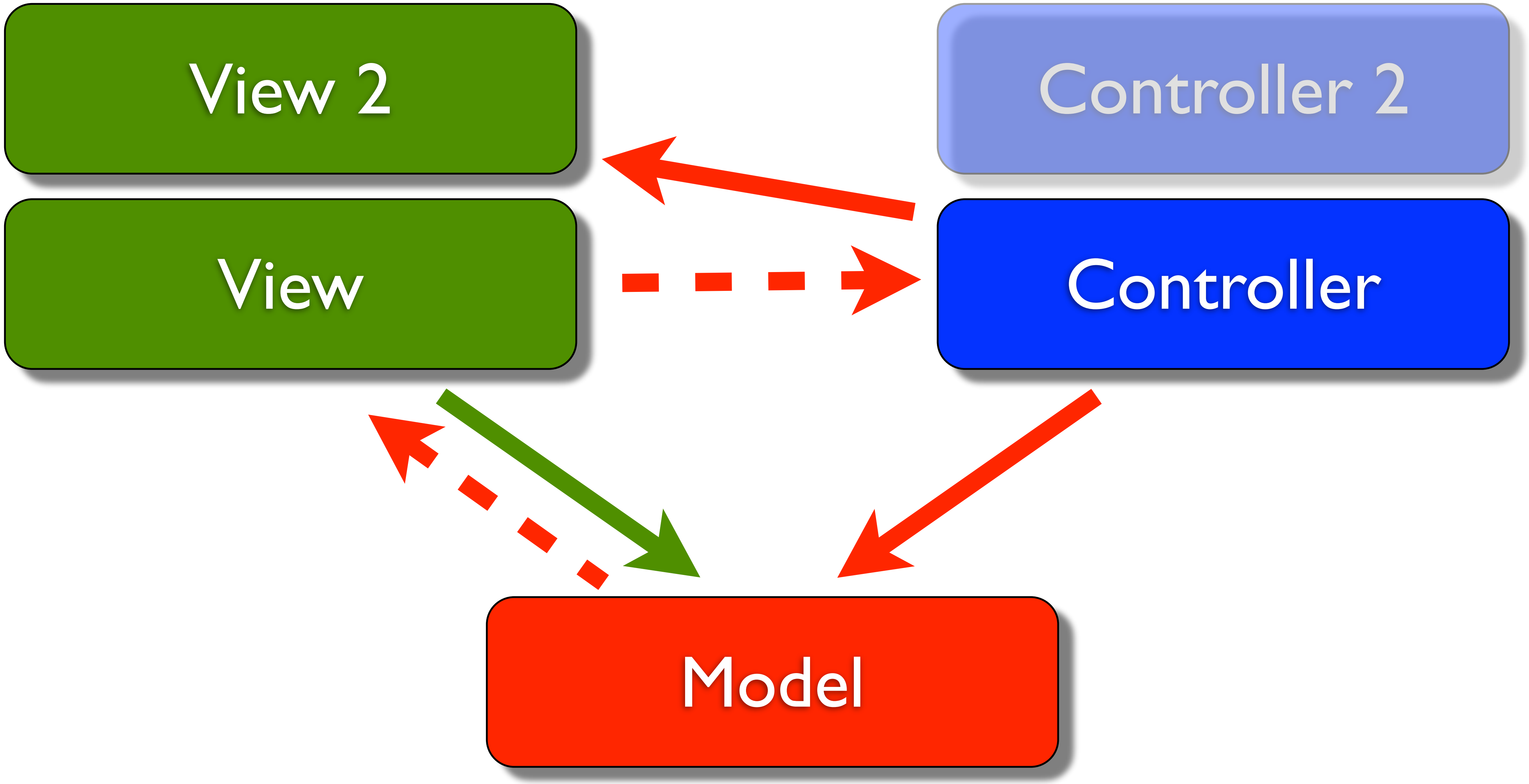

View
Controller
Model

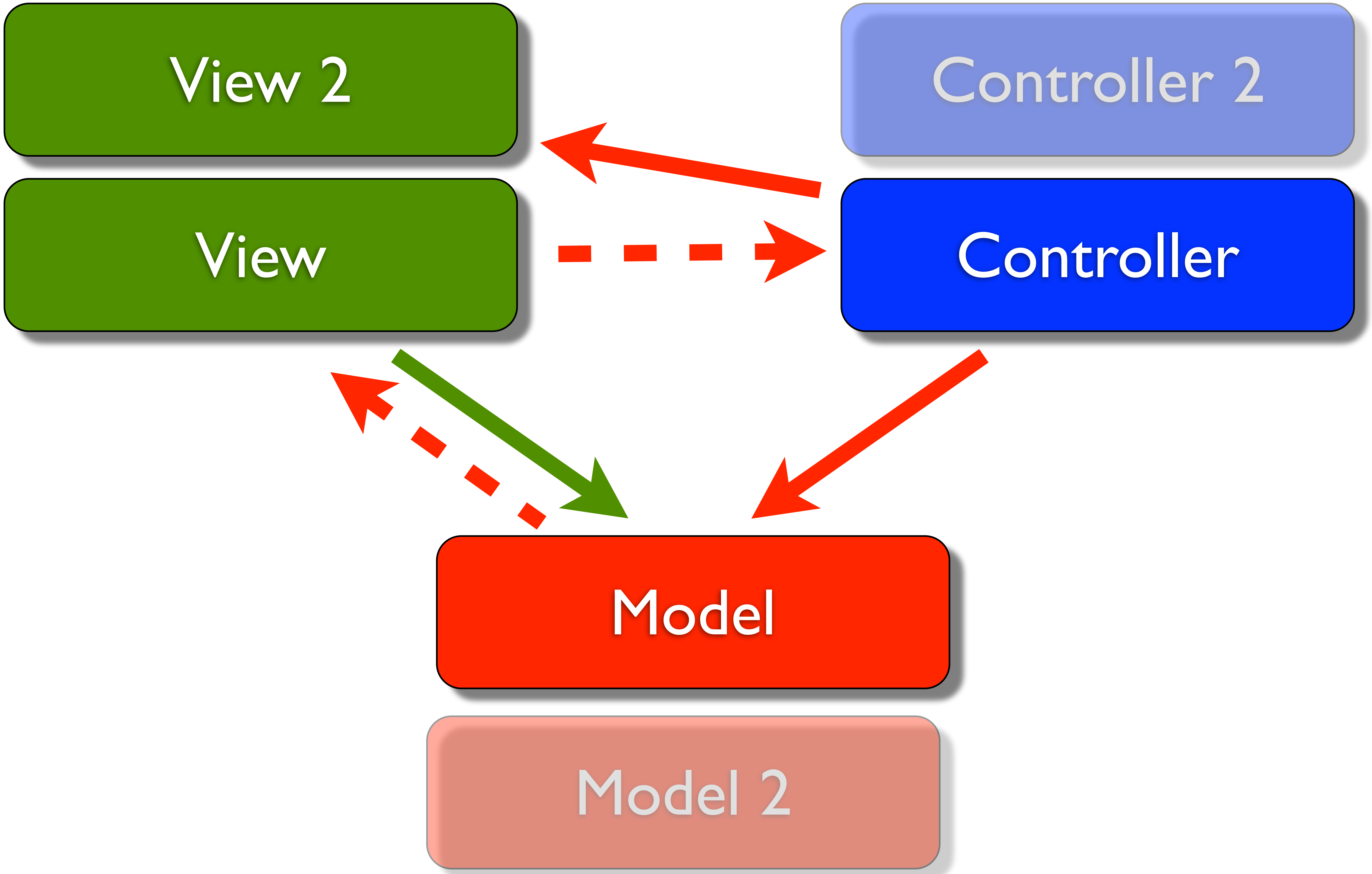

View
Controller
Model

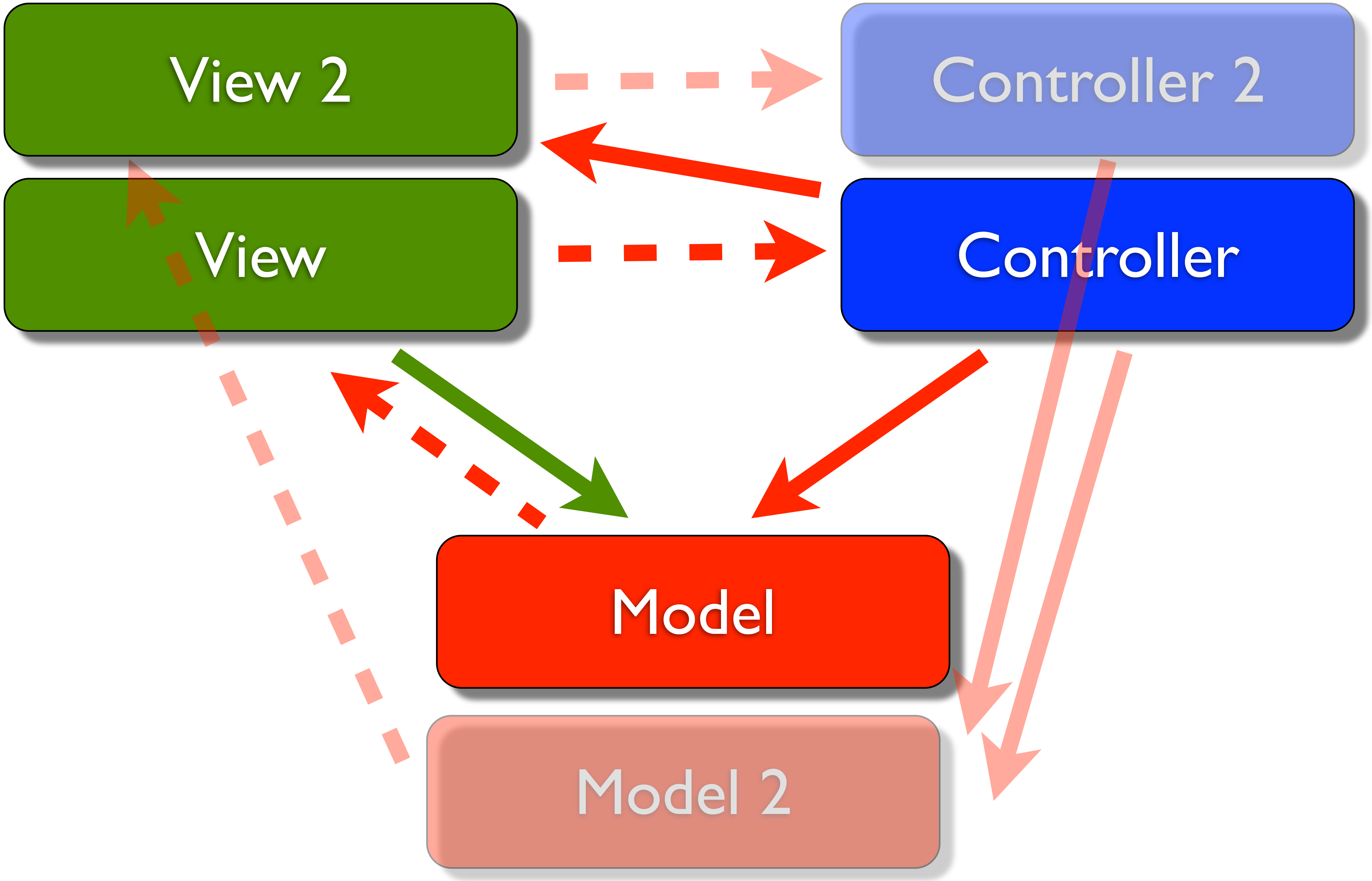

View
Controller
Model


View 2
View
Controller
Model


View 2
View
Controller
Model


View 2
Controller 2
View
Controller
Model

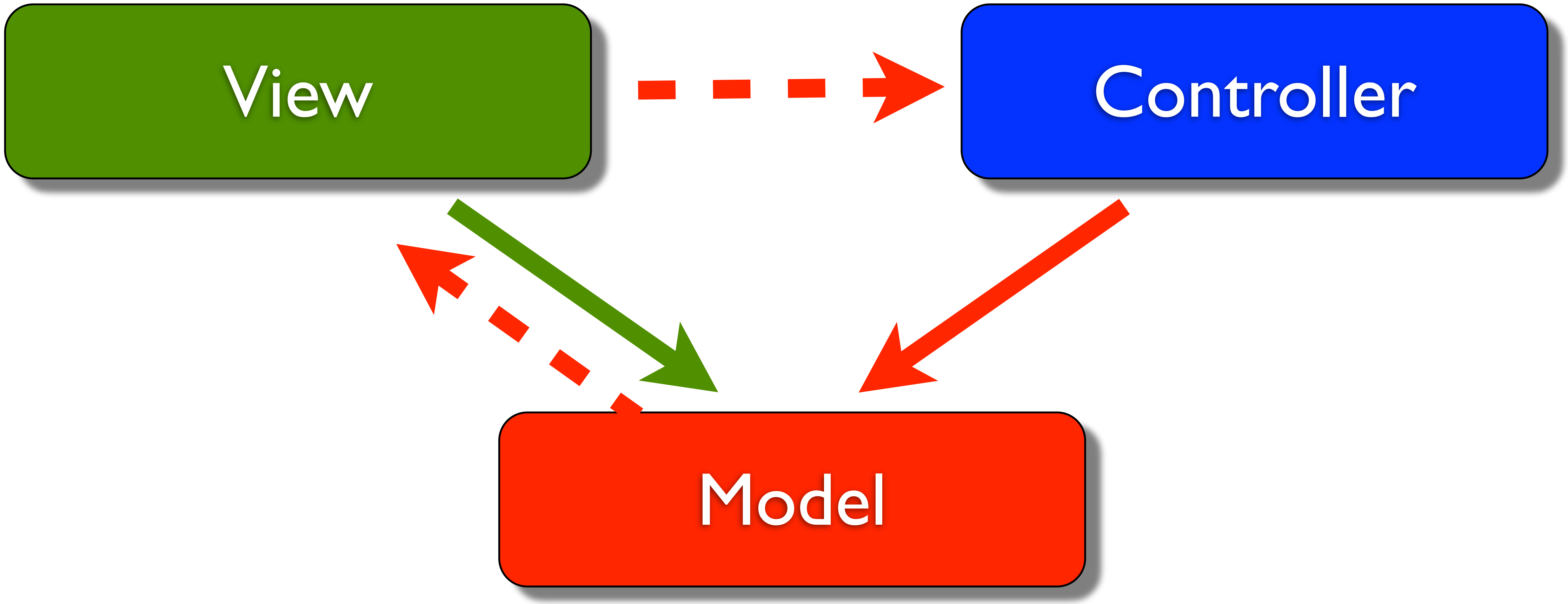

View 2
Controller 2
View
Controller
Model
Model 2

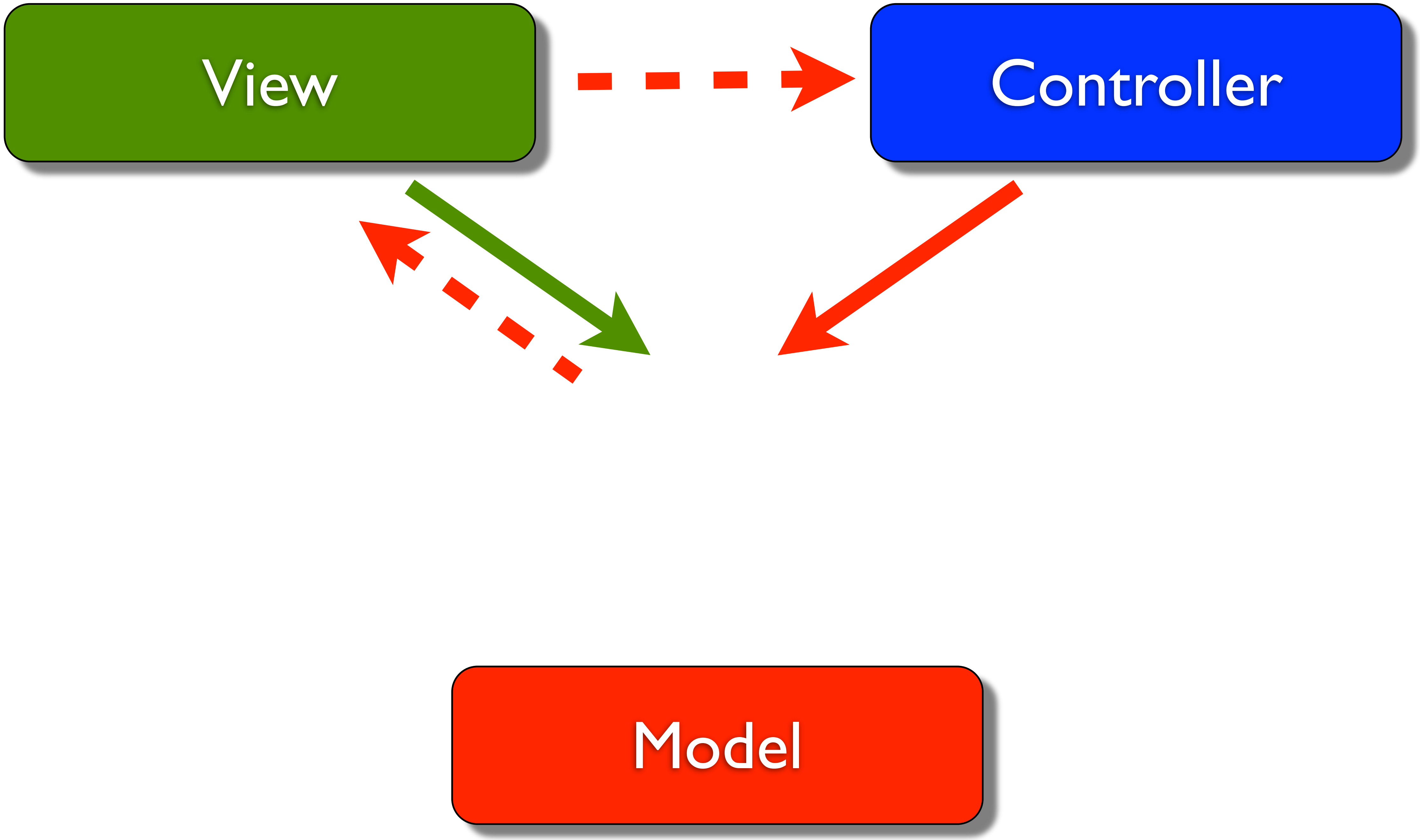

View 2
Controller 2
View
Controller
Model
Model 2

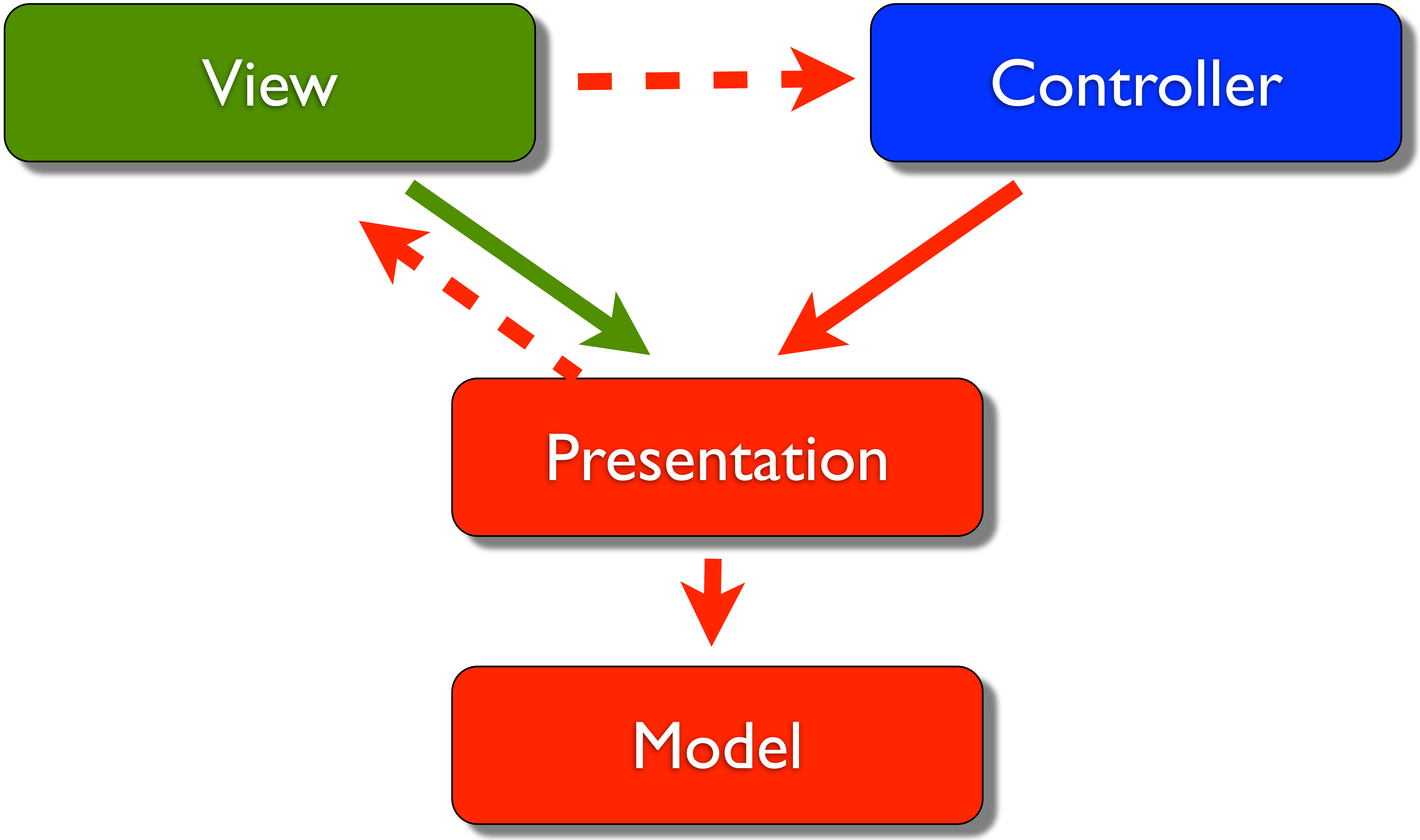
Solution ideas

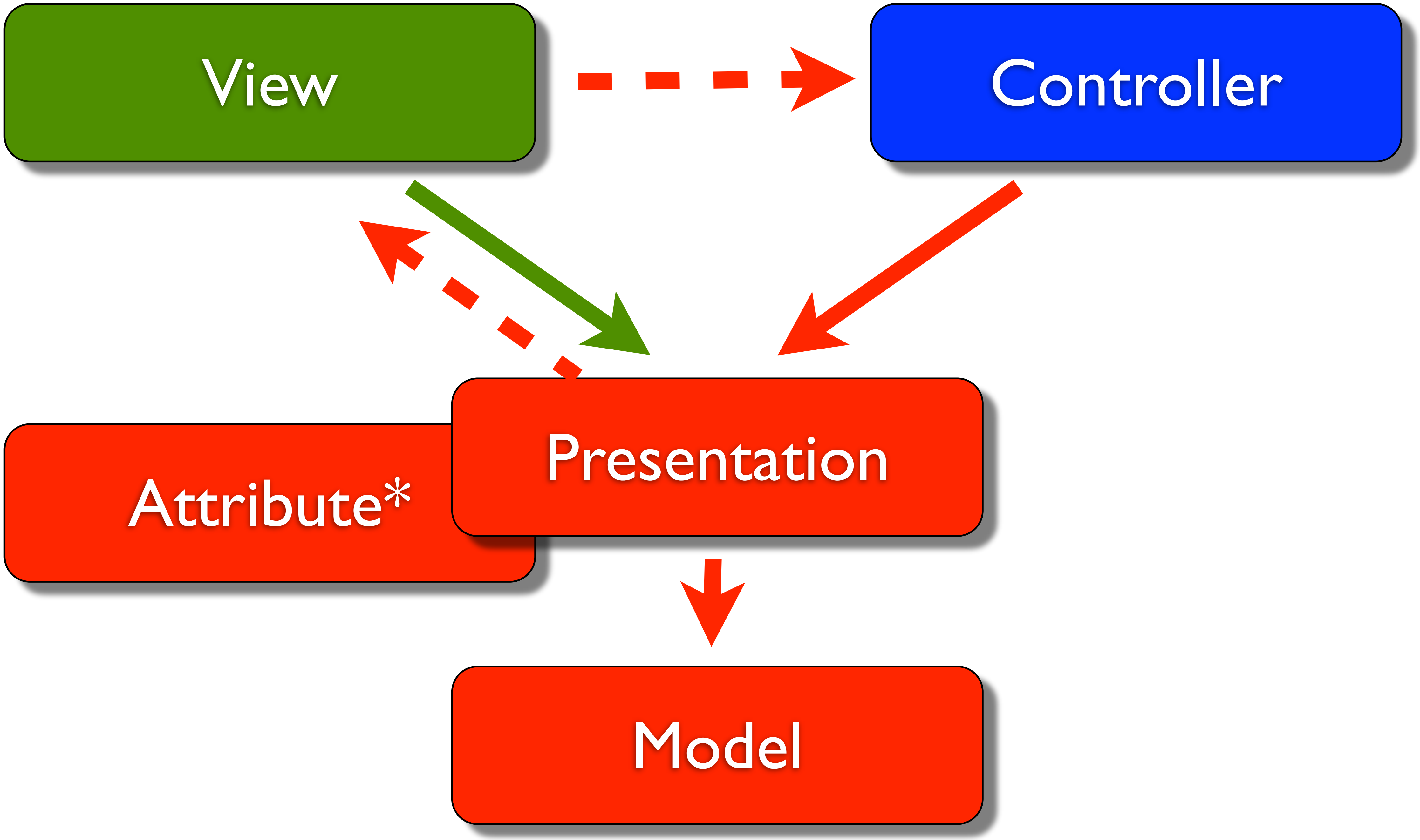
event bus

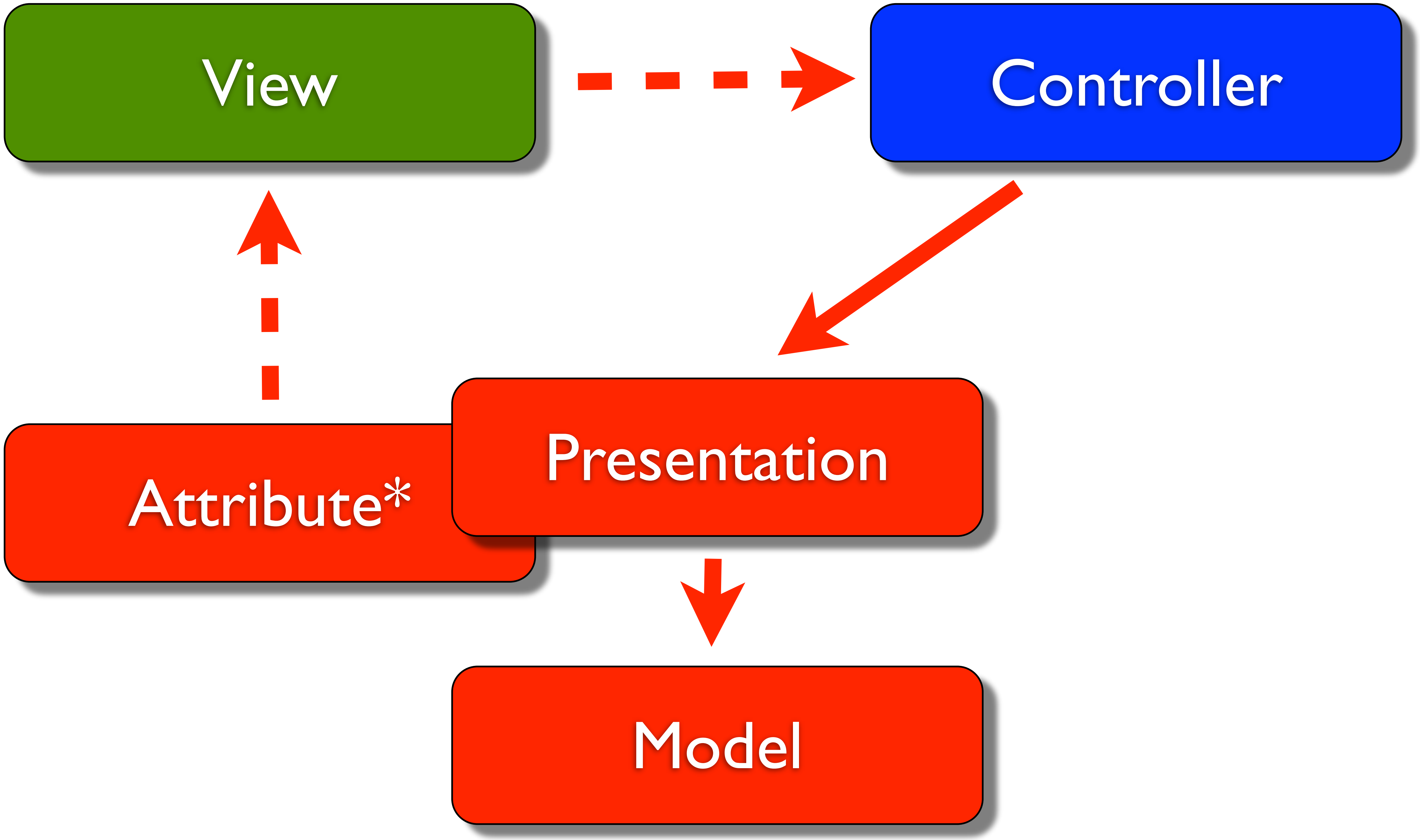
presentation model


View
Controller
Model


View
Controller
Model


View
Controller
Presentation
Model


View
Controller
Presentation
Attribute*
Model


View
Controller
Presentation
Attribute*
Model

Grasp - Groovy application split

- experimental, general, open-source implementation of the Presentation Model Pattern in Groovy
- captures experiences from 3 major
  Canoo RIA Suite projects
  in banking and telco
- github.com/canoo/grasp
  also contains all code of this presentation
- built with Gradle, tested with Spock

Grasp in the Canoo Code Camp

A code walkthrough

Resulting context - future options

- No view knows any other view
  neither directly nor through their controllers
- PMs can easily capture lots of meta-information
  in addition to presentation state
- Collections and Switches need more consideration
- Scaffolded views? Editors? Remoting?

Linkiography

- martinfowler.com/eaaDev/PresentationModel.html
- blogs.adobe.com/paulw/archives/2007/10/presentation_pa_3.html
- msdn.microsoft.com/en-us/library/ff921080%28PandP.20%29.aspx
- www.canoo.com/ulc
- github.com/canoo/grasp

Q & A

Q & A

dierk.koenig@canoo.com
@mittie