Architectural patterns

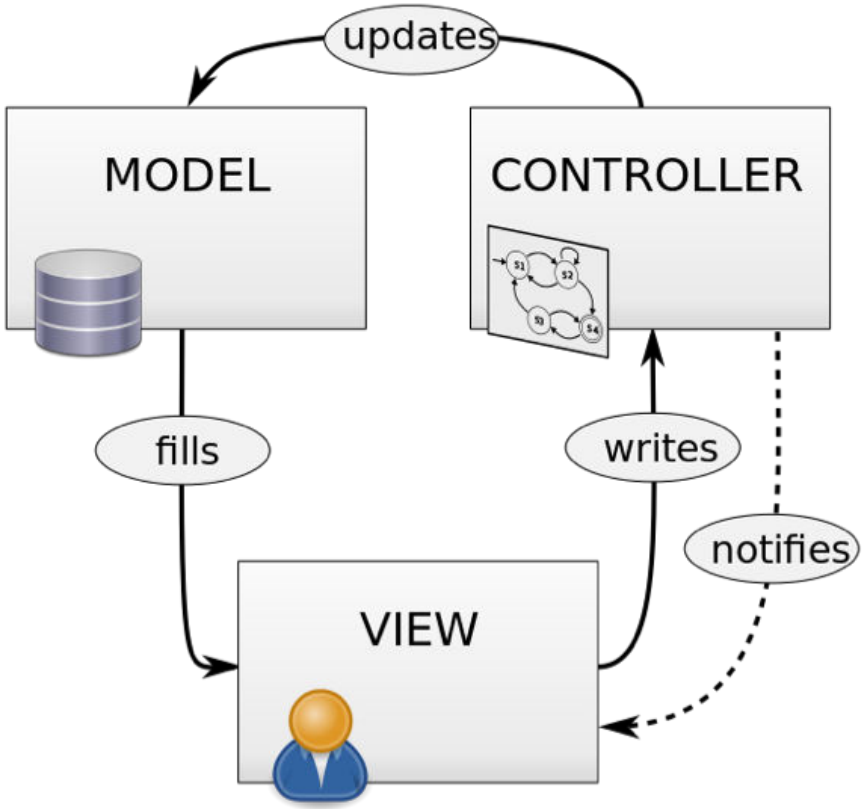
NGUYEN Thi Thu Trang

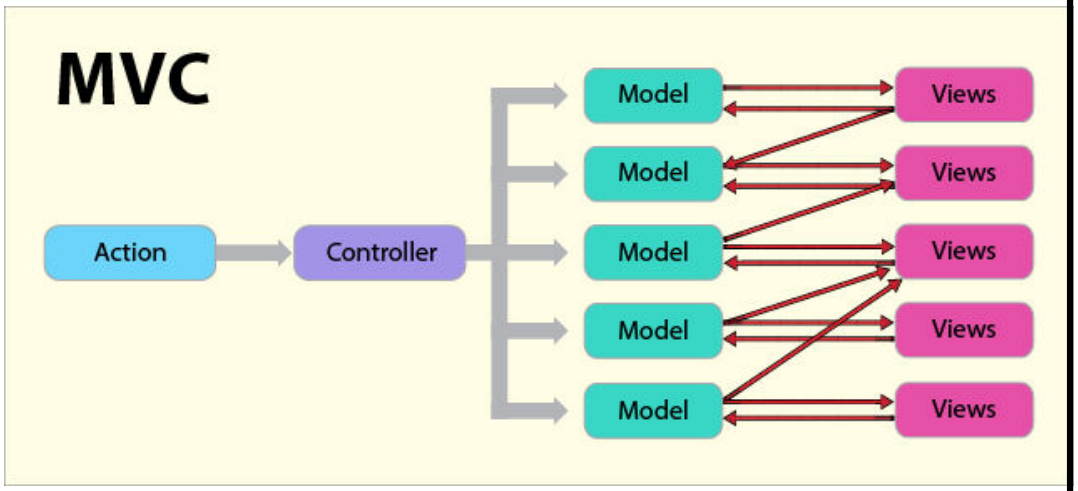
Architectural pattern

- General, reusable solution to a commonly occurring problem in software architecture within a given context.
  - Performance
  - High availability
- Some architectural patterns have been implemented within software frameworks
  - 3-layer/3-tier pattern: DotNetNuke
  - MVC pattern: Laravel, ASP.NET MVC
  - Microservices: Spring Boot, vert.x

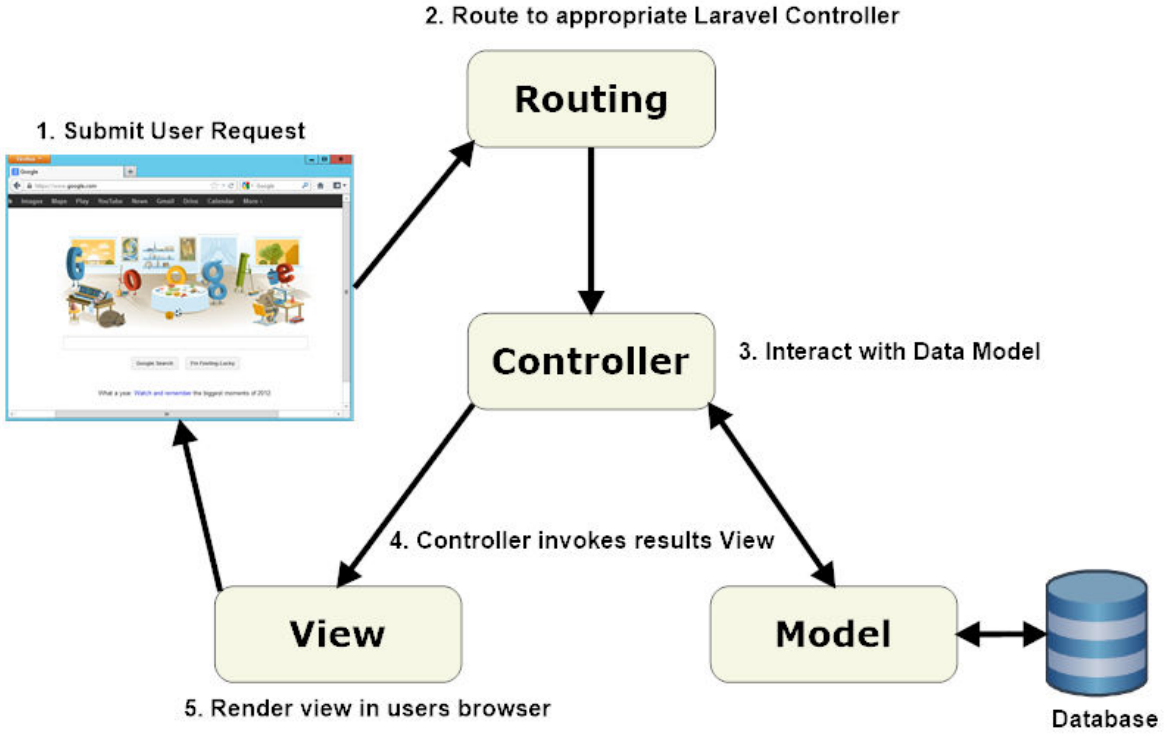
Classic MVC architecture (active)

Problems?

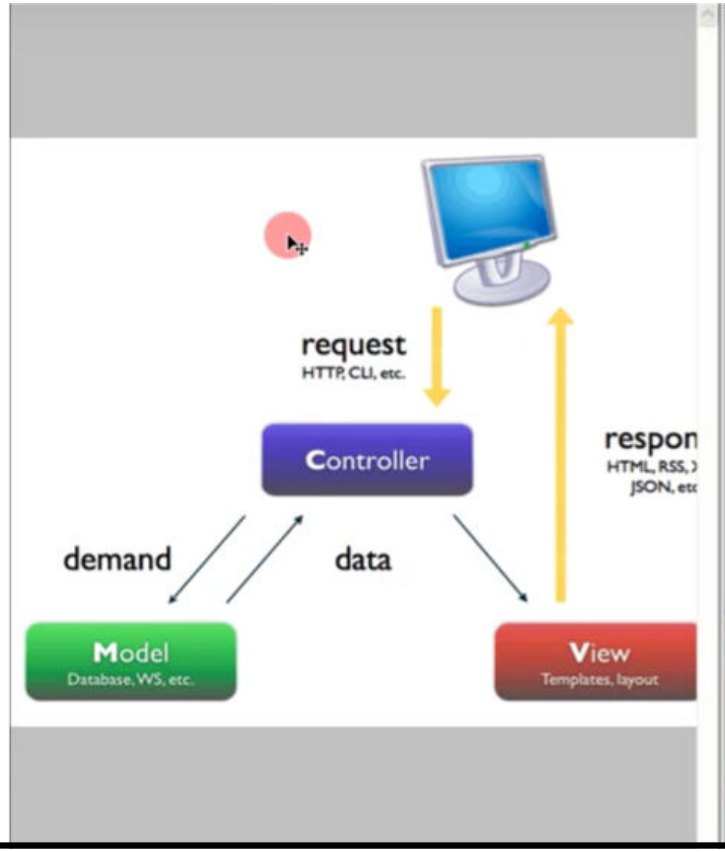


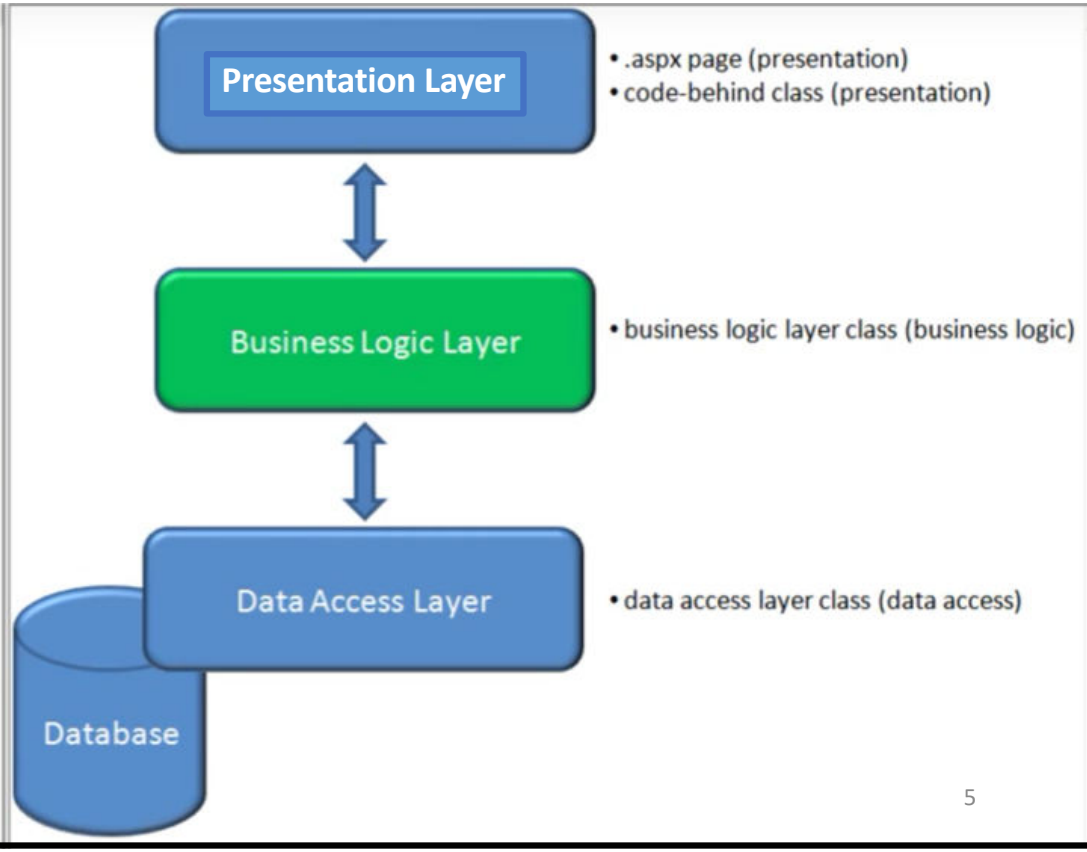


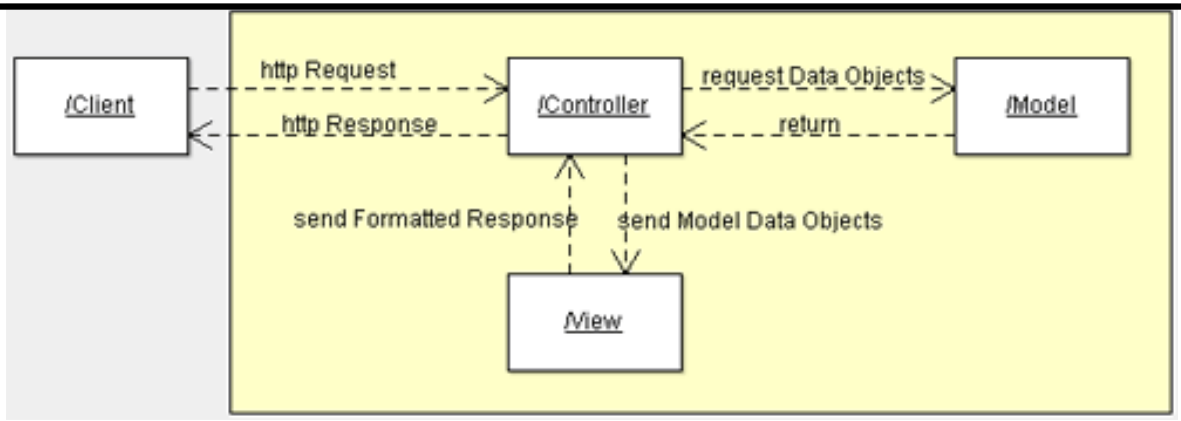
Passive MVC architecture



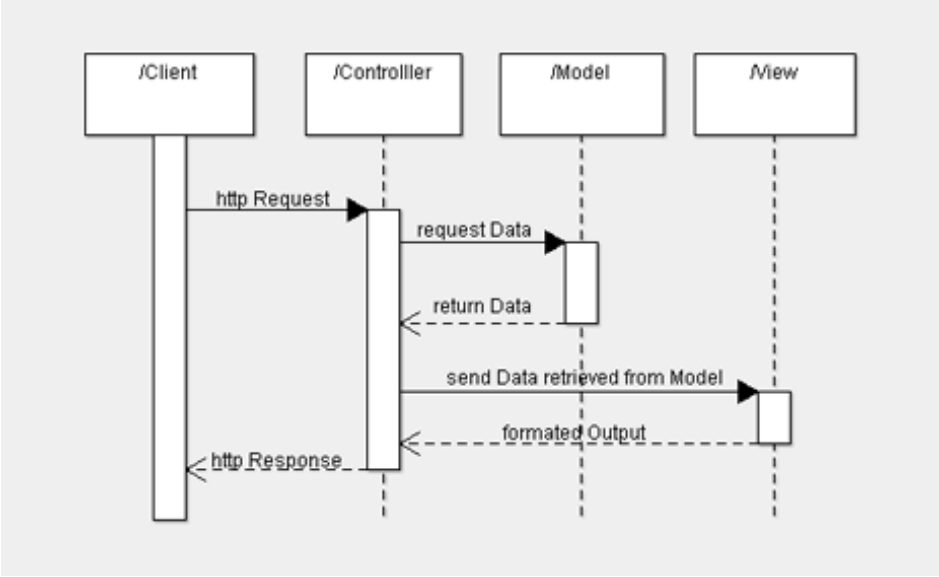
MVC vs 3-tier Architecture

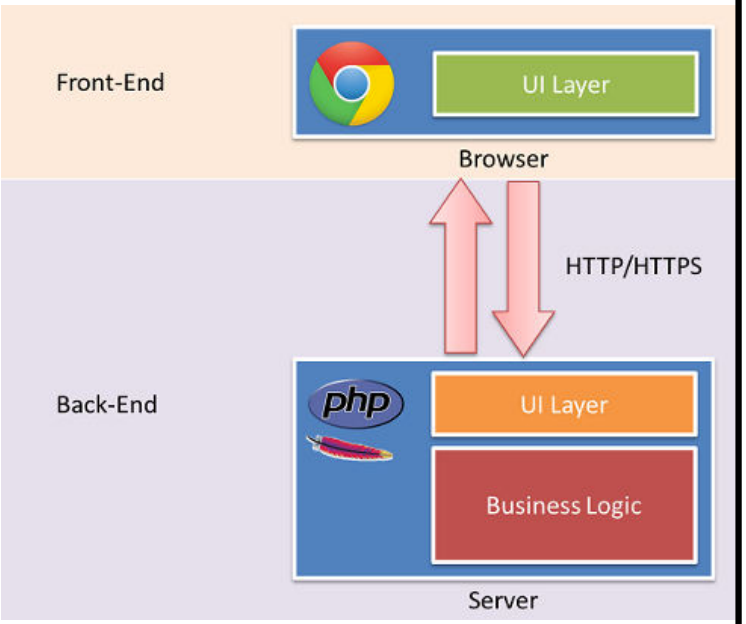




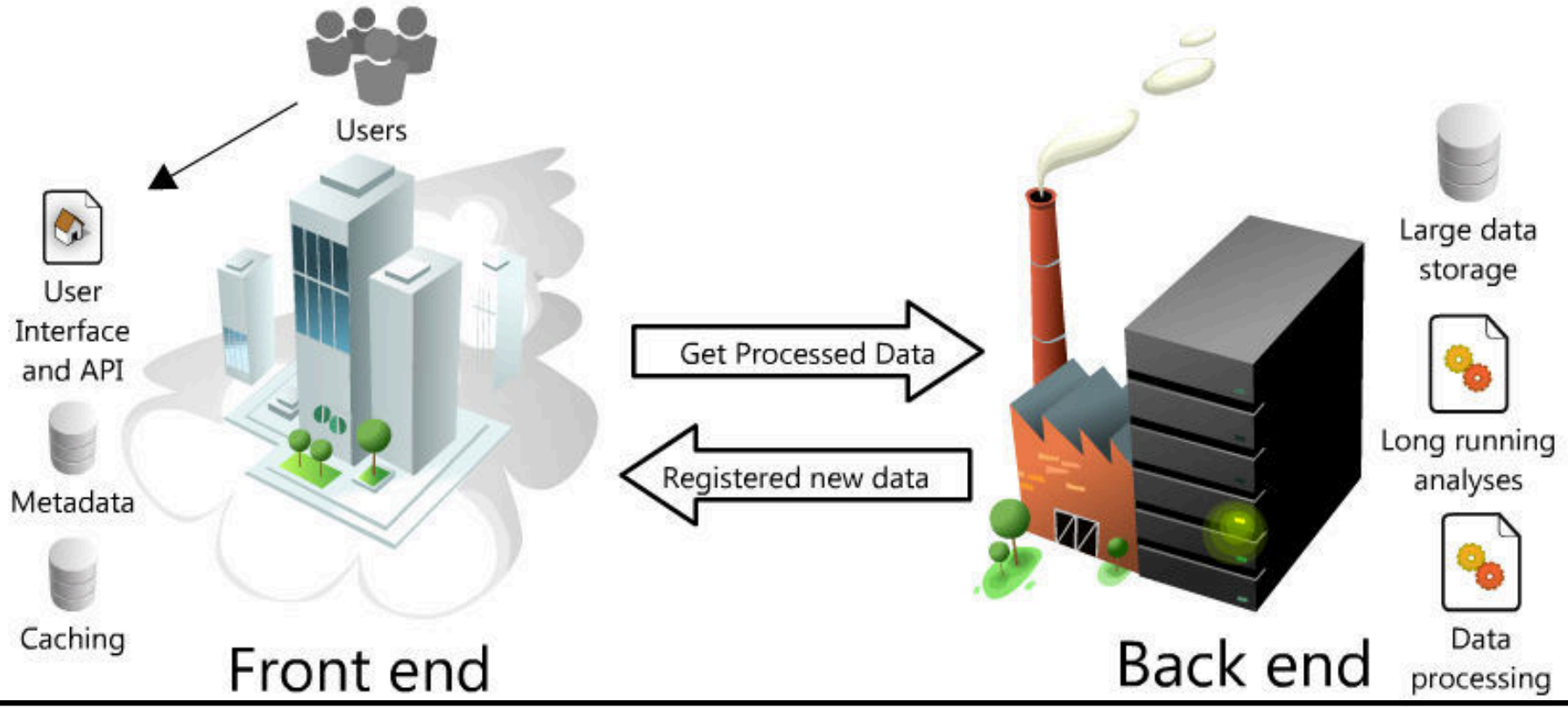

/Client
http Request
http Response
/Controller
request Data Objects
return
/Model
send Formatted Response
send Model Data Objects
/View

MVC


/Client
/Controller
/Model
/View
http Request
request Data
return Data
send Data retrieved from Model
formated Output
http Response

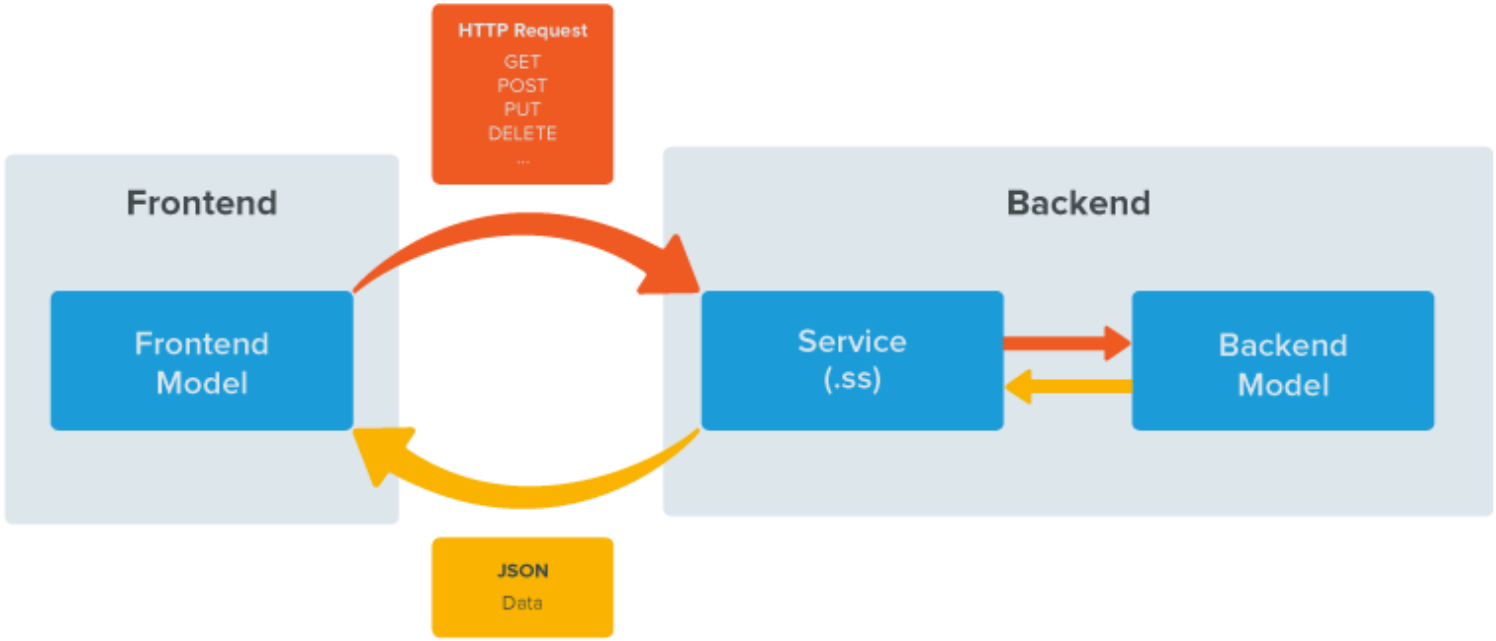

Front-End
UI Layer
Browser
HTTP/HTTPS
Back-End
php
UI Layer
Business Logic
Server

Front-end and Back-end

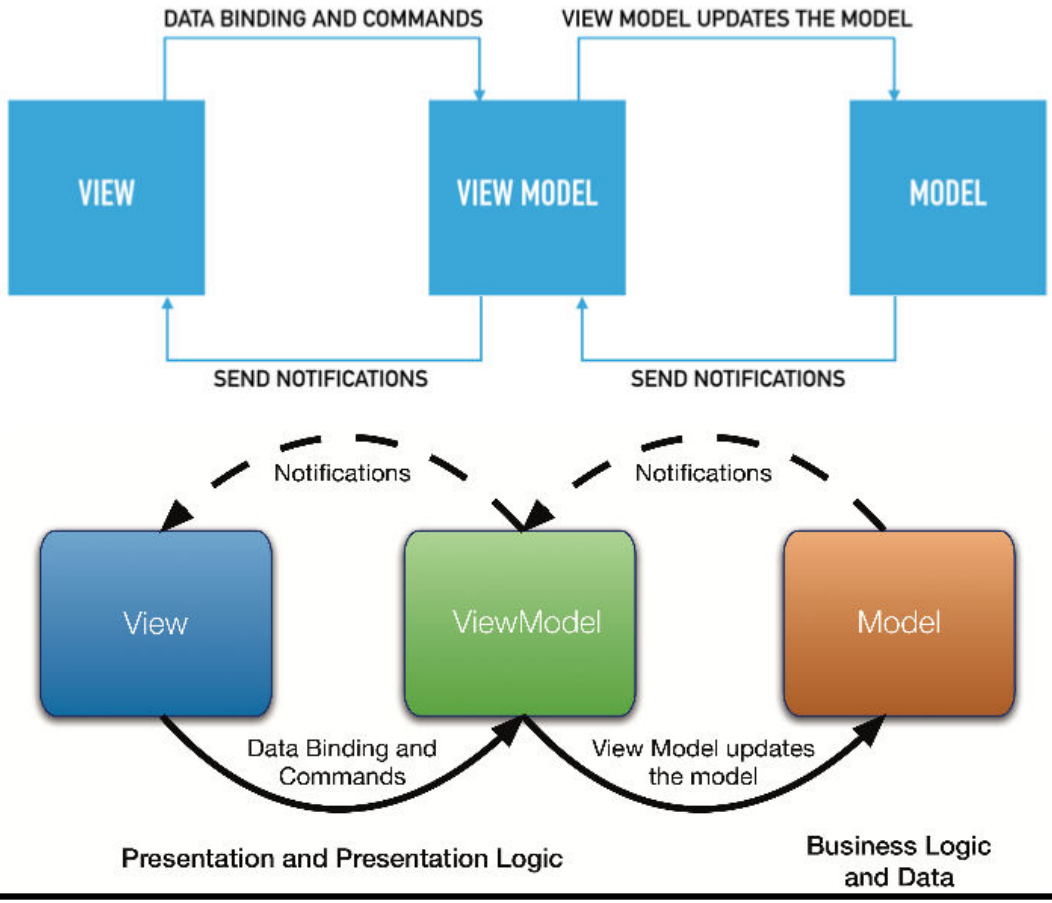


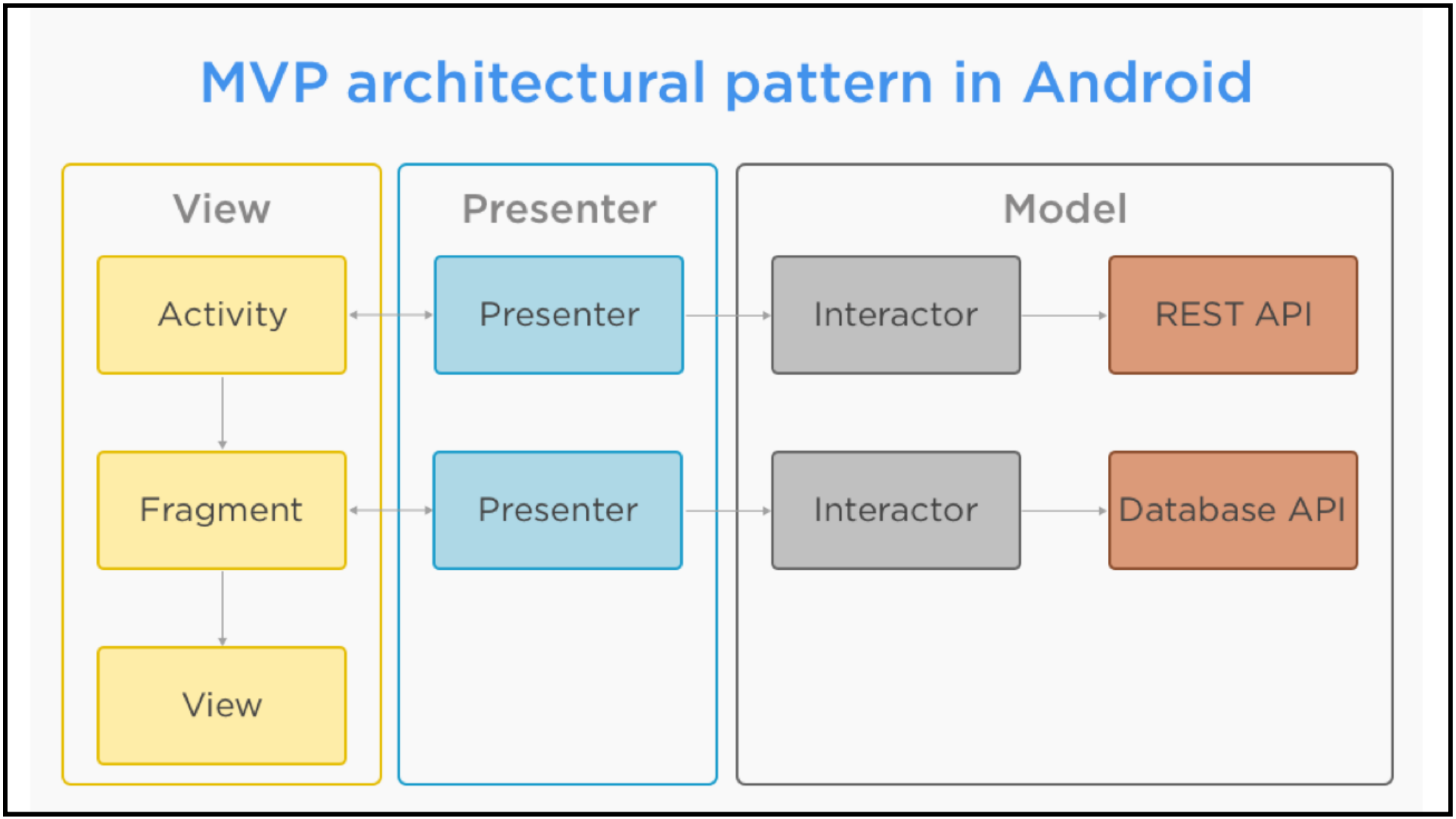
Front end and Back end

- May have different architectural patterns

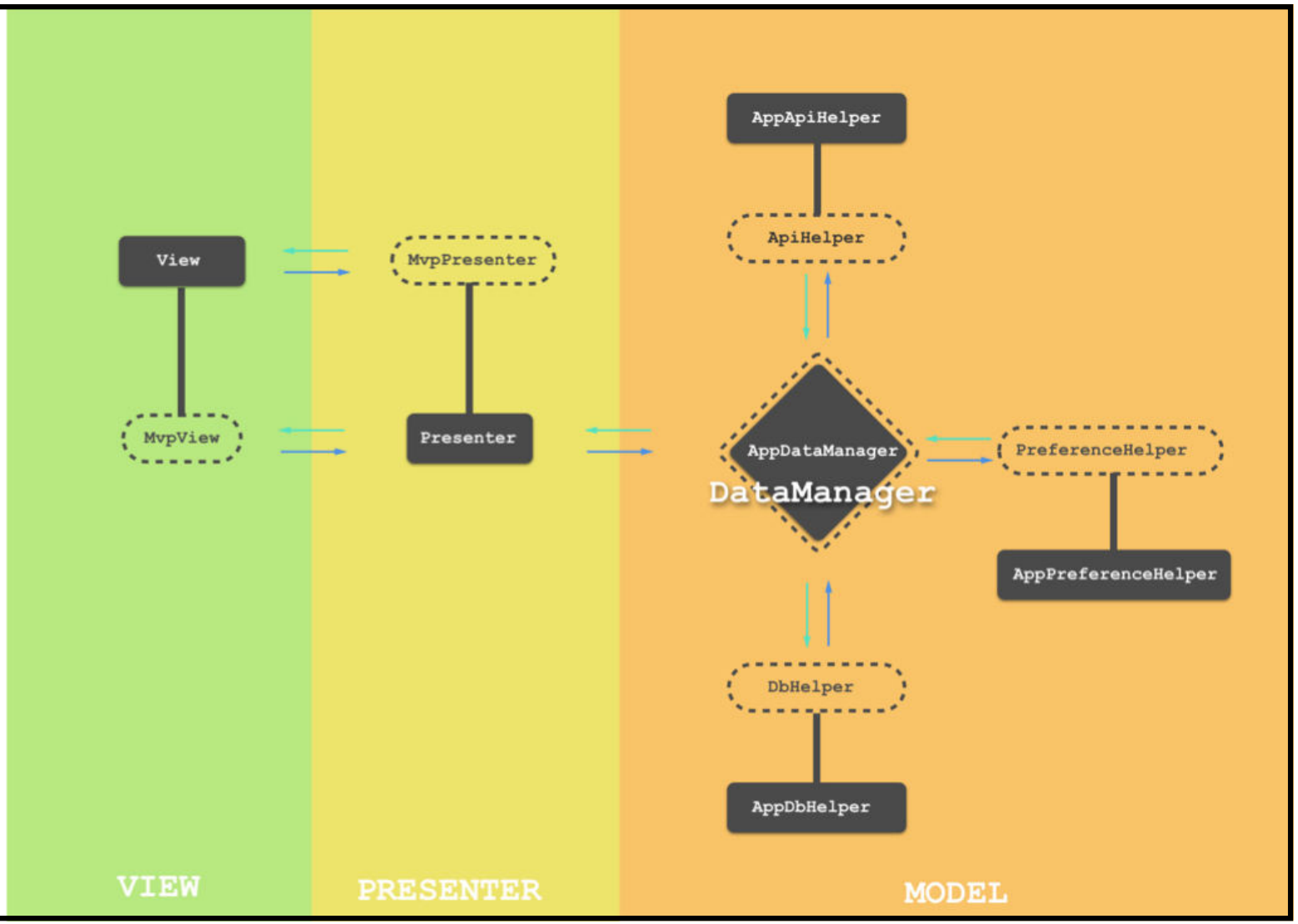


MVVM Architecture

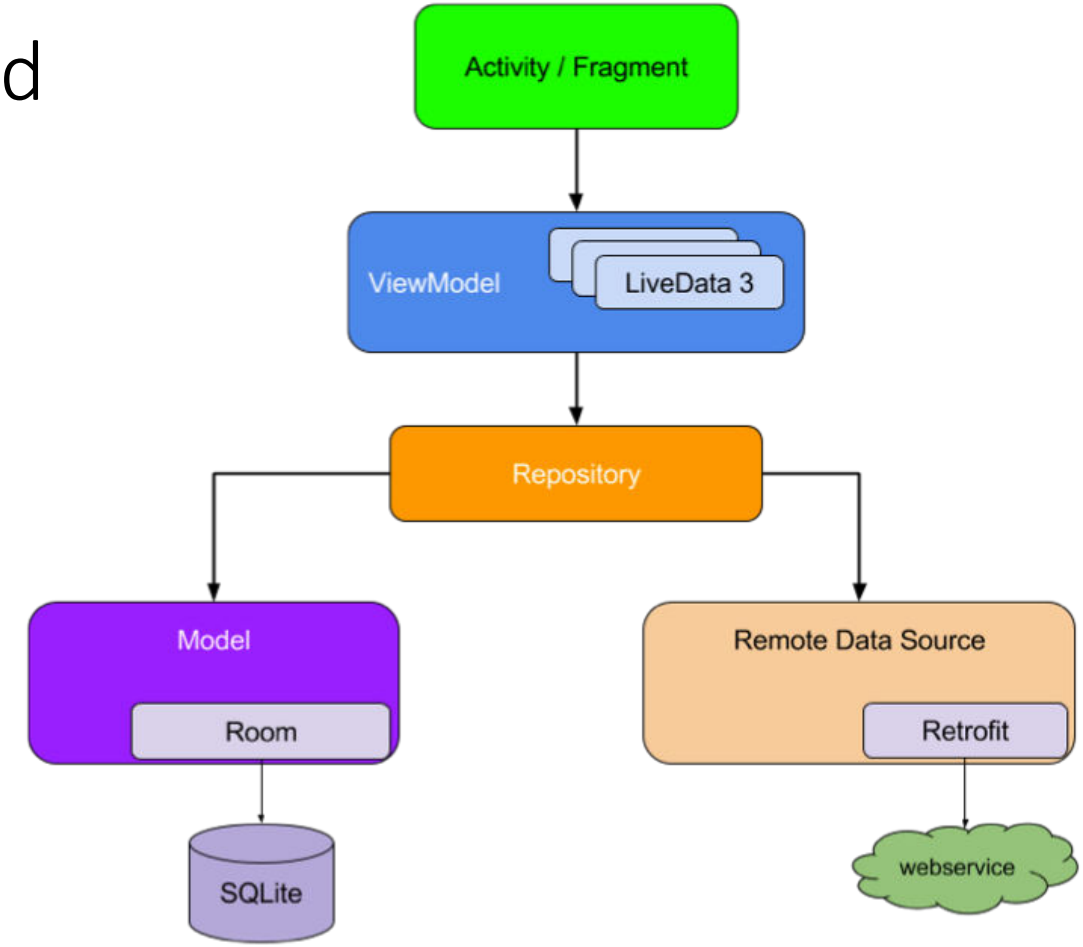



MVP architectural pattern in Android
View
Presenter
Model
Activity
Fragment
View
Presenter
Presenter
Interactor
Interactor
REST API
Database API

MVP in Android



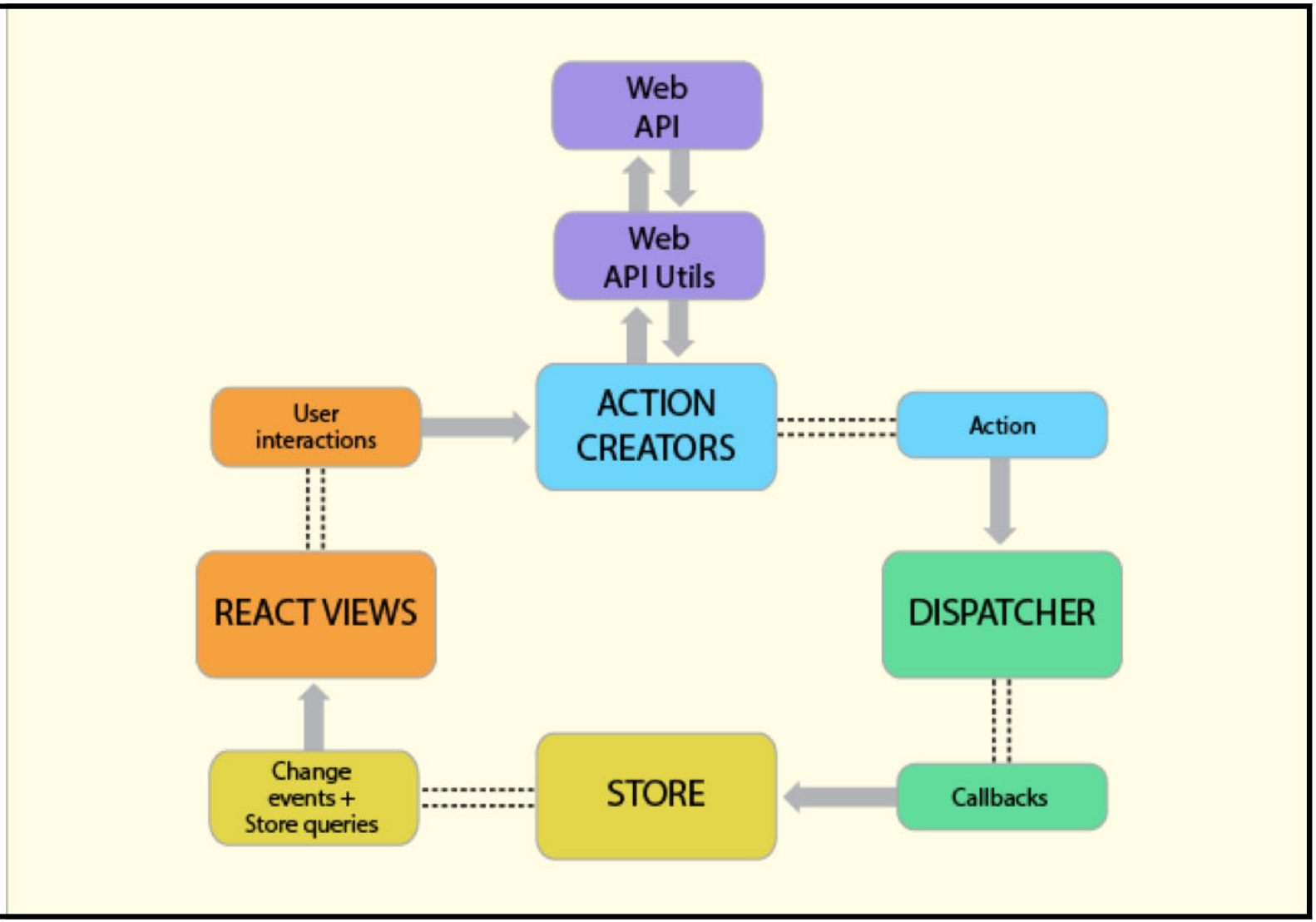
MVP in Android



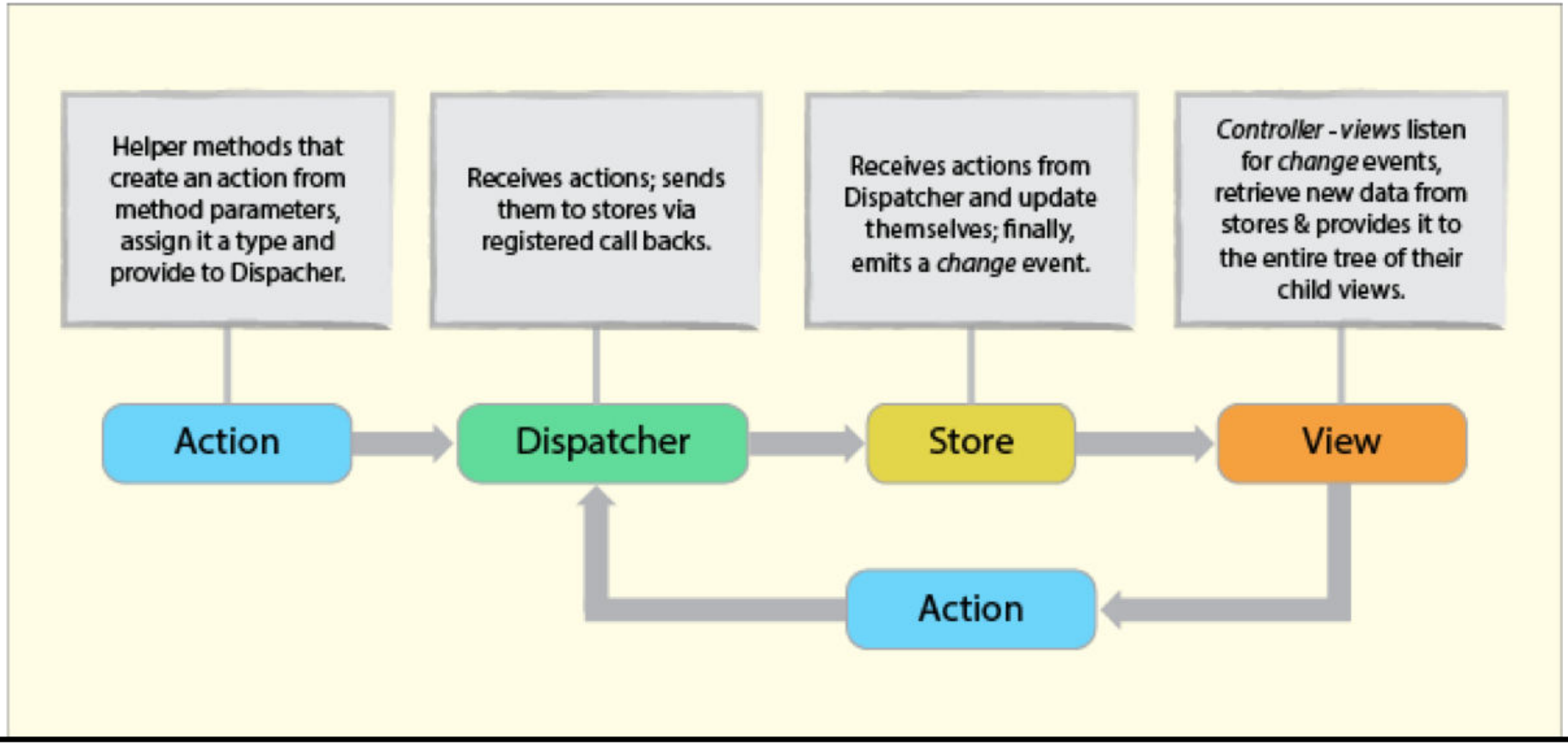
Flux

architecture

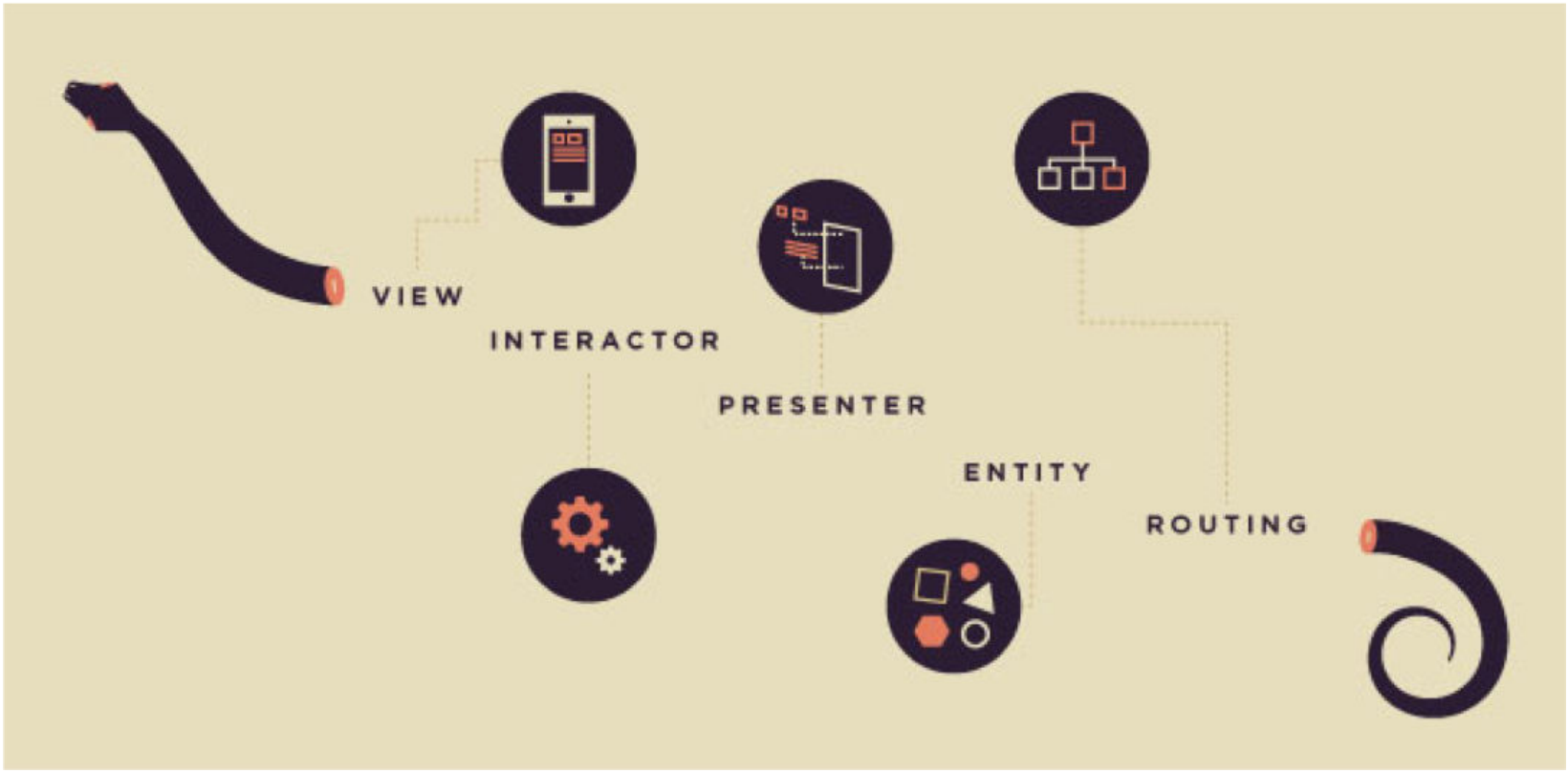
(React)



Flux architecture (React)



VIPER architecture (iOS)



VIPER architecture (iOS)