Client Side Javascript Frameworks

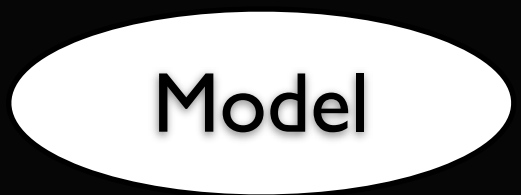
wifi: growlabwifi
pw: Acc3lr8r
https://github.com/anthonyc/MVC

What is MVC?

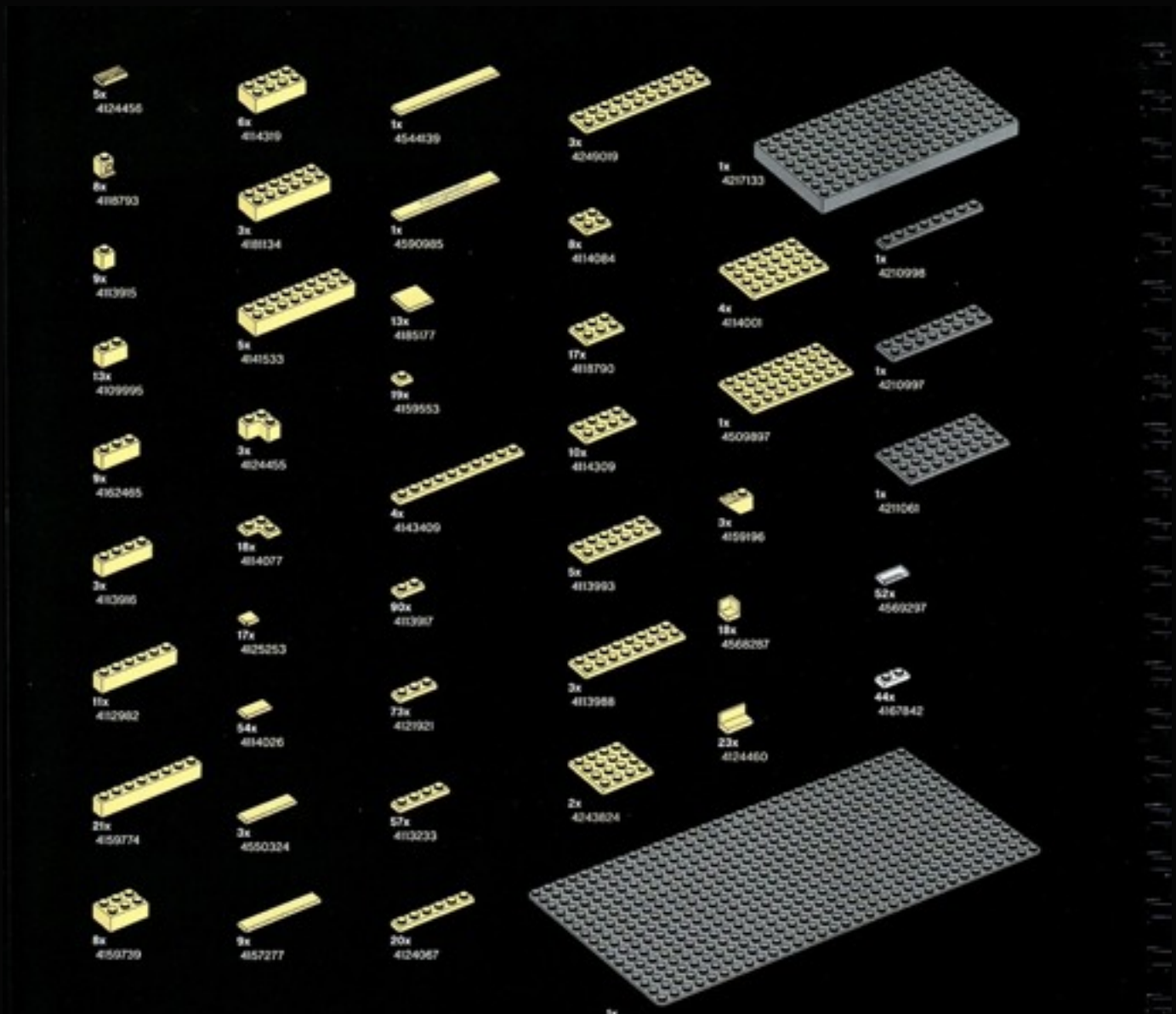
A design pattern developed in the 70s which separates the representation of information from the user's interaction with it.

Model

The atomized domain specific data of your application (photo, tweet, user)



Model


5x 4124456
8x 4114319
1x 4544139
3x 4249019
1x 4217133
8x 4118793
3x 4181134
1x 4590985
8x 4114084
1x 4210998
9x 4113915
5x 4141533
13x 4185177
4x 4140001
1x 4210997
13x 4109995
19x 4159553
17x 4118793
1x 4509897
9x 4162465
3x 4124455
10x 4114309
1x 4211061
3x 4143409
3x 4159196
3x 4113996
18x 4114077
5x 4113993
52x 4569297
17x 4125253
92x 4113987
18x 4568287
11x 4112982
54x 4114026
73x 4121921
3x 4113988
44x 4167842
23x 4124460
21x 4159774
3x 4550324
57x 4113233
2x 4143824
8x 4159739
9x 4157277
20x 4124067
1x

Model

```
USER
firstName:
gender:
id:
is_friend:
name:
searchTokens:
showVideoPromo:
social_snippets:
thumbSrc:
type:
uri:
vanity:
```

Model

USER
firstName:
gender:
id:
is_friend:
name:
searchTokens:
showVideoPromo:
social_snippets:
thumbSrc:
type:
uri:
vanity:

TWEET
created_at:
from_user:
from_user_id:
from_user_id_str:
from_user_name:
geo:
id:
id_str:
iso_language_code:
metadata:
profile_image_url:
profile_image_url_https:
source:
text:

Model

USER
firstName:
gender:
id:
is_friend:
name:
searchTokens:
showVideoPromo:
social_snippets:
thumbSrc:
type:
uri:
vanity:

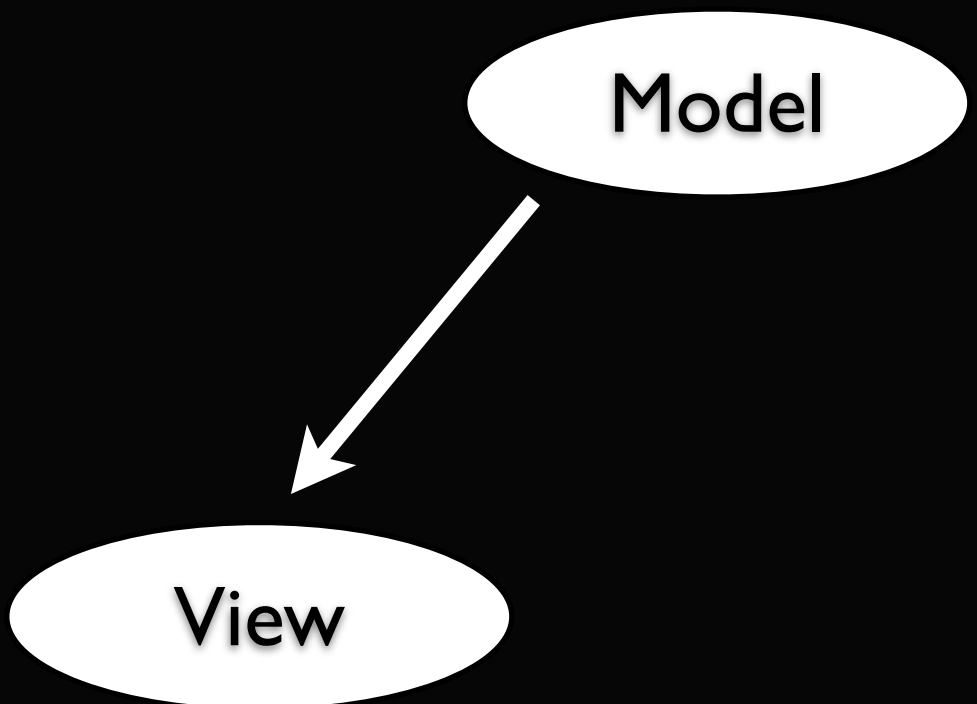
TWEET
created_at:
from_user:
from_user_id:
from_user_id_str:
from_user_name:
geo:
id:
id_str:
iso_language_code:
metadata:
profile_image_url:
profile_image_url_https:
source:
text:

PHOTO
caption:
code:
comments:
date:
display_src:
id:
is_video:
likes:
owner:
status:

Model

The atomized domain specific data of your application (User, photo, tweet)

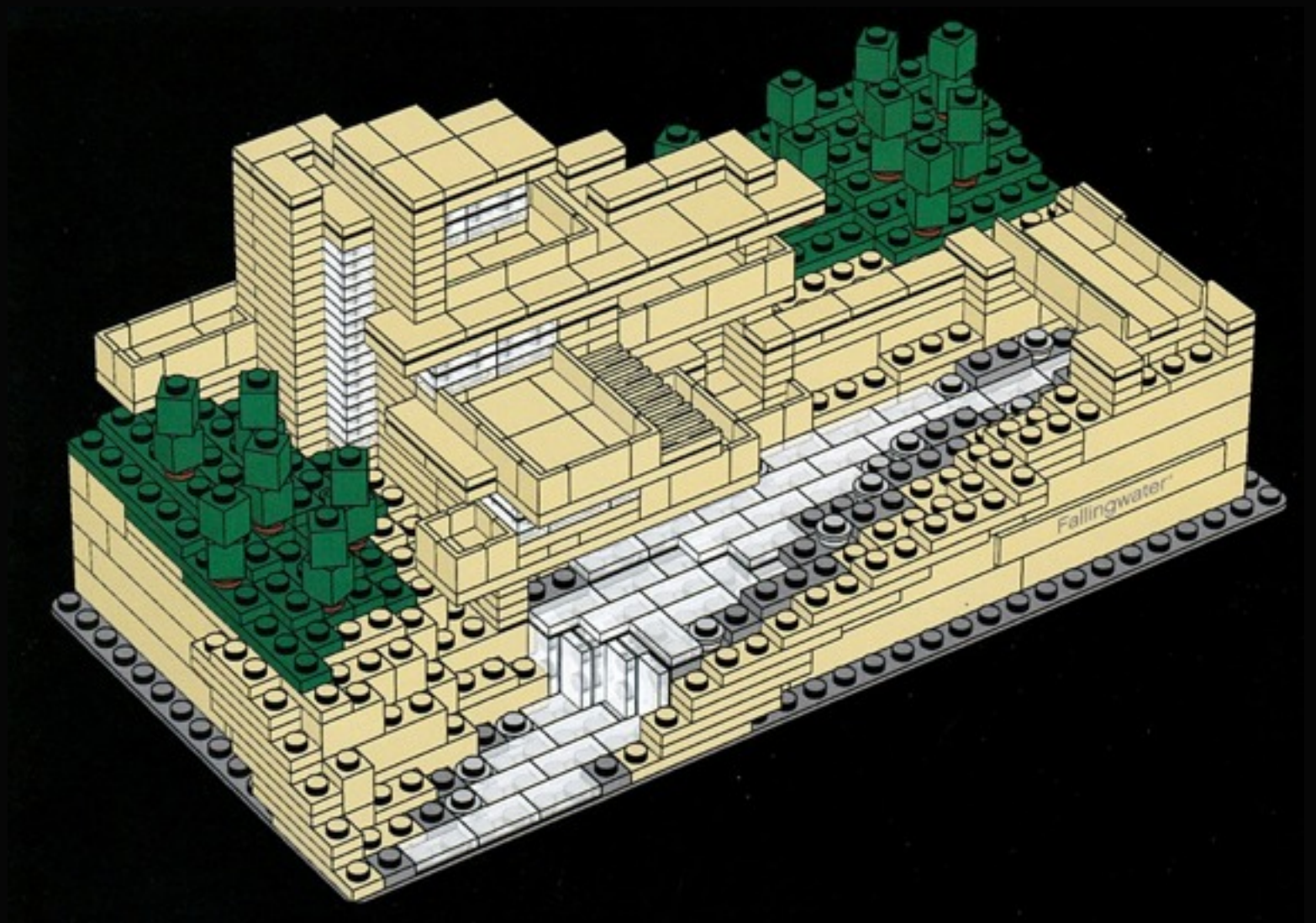
Notify observers of state/data changes



View

The UI/Presentation Layer

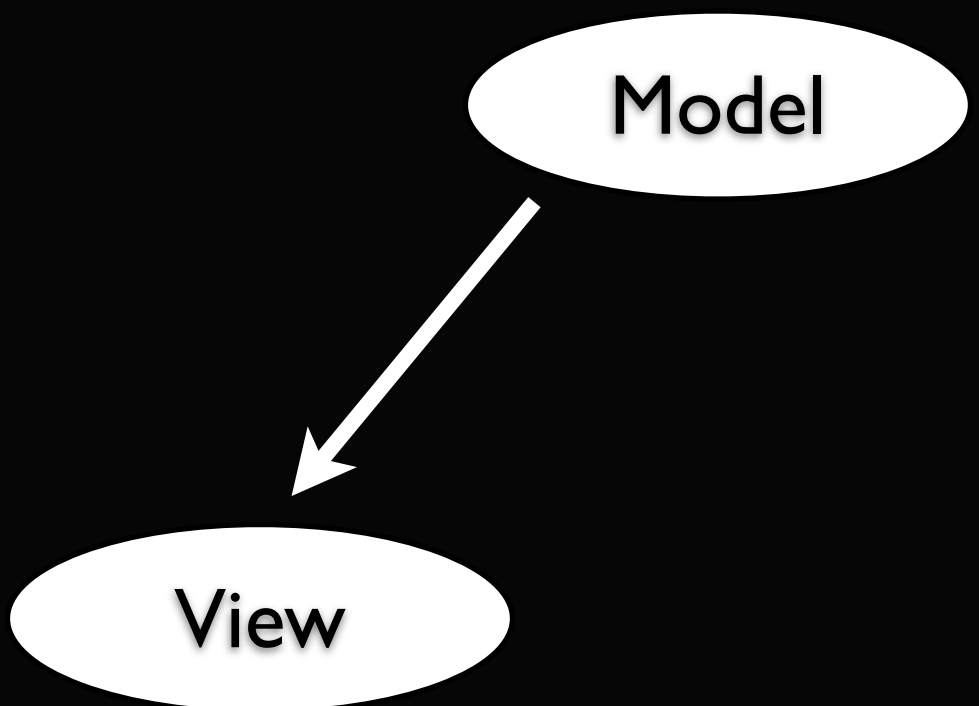
View



View

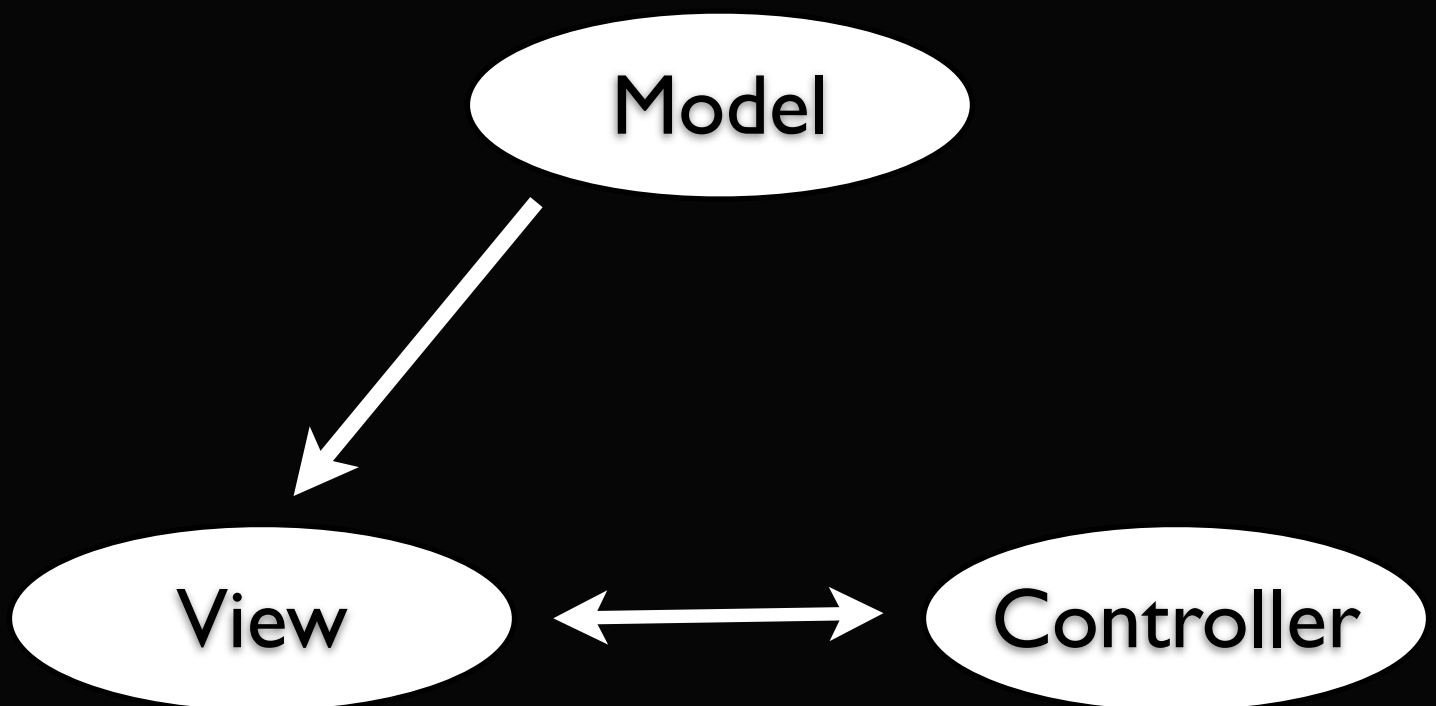
The UI/Presentation Layer

Observe models and update UI accordingly

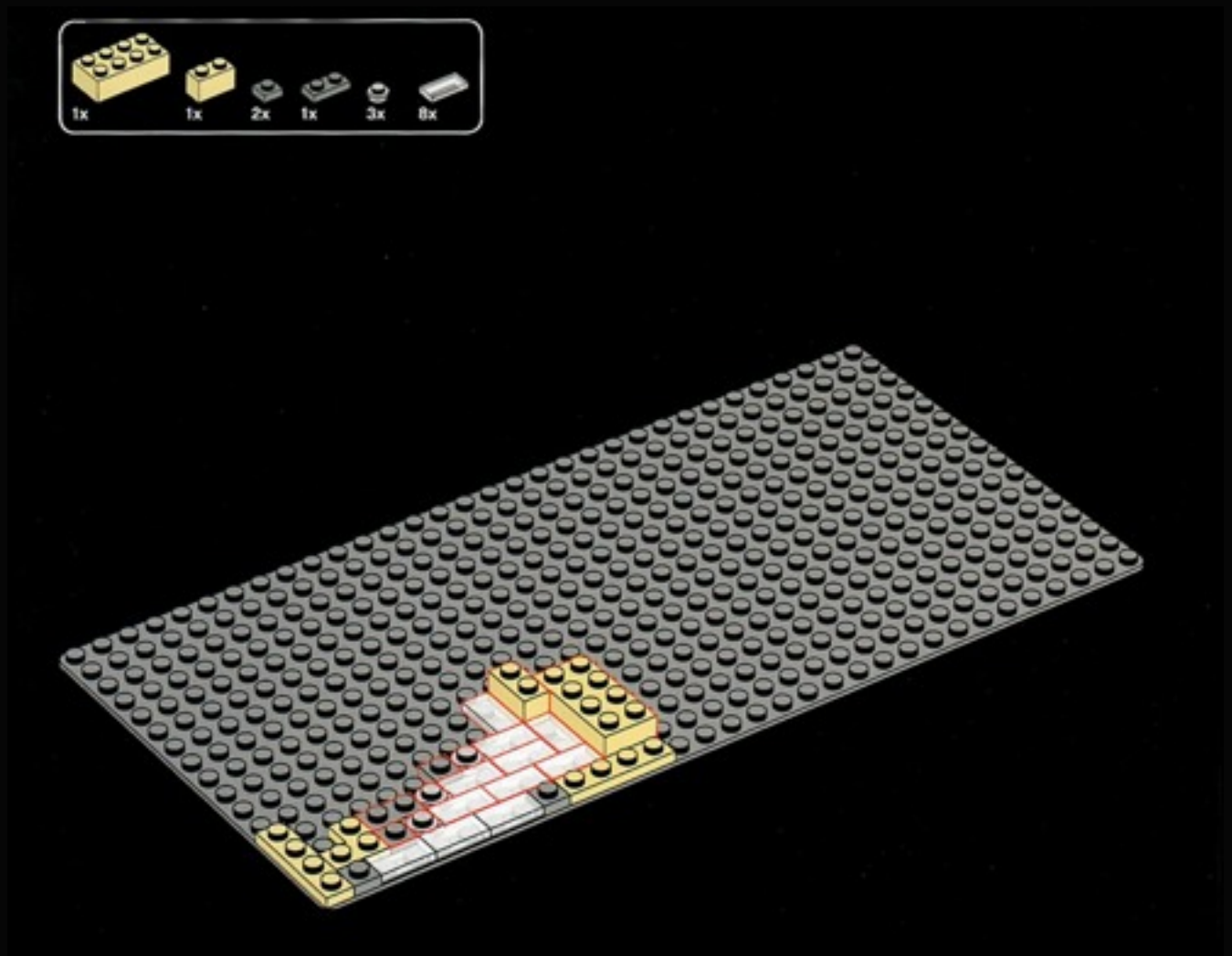


Controller

Handles application and user events



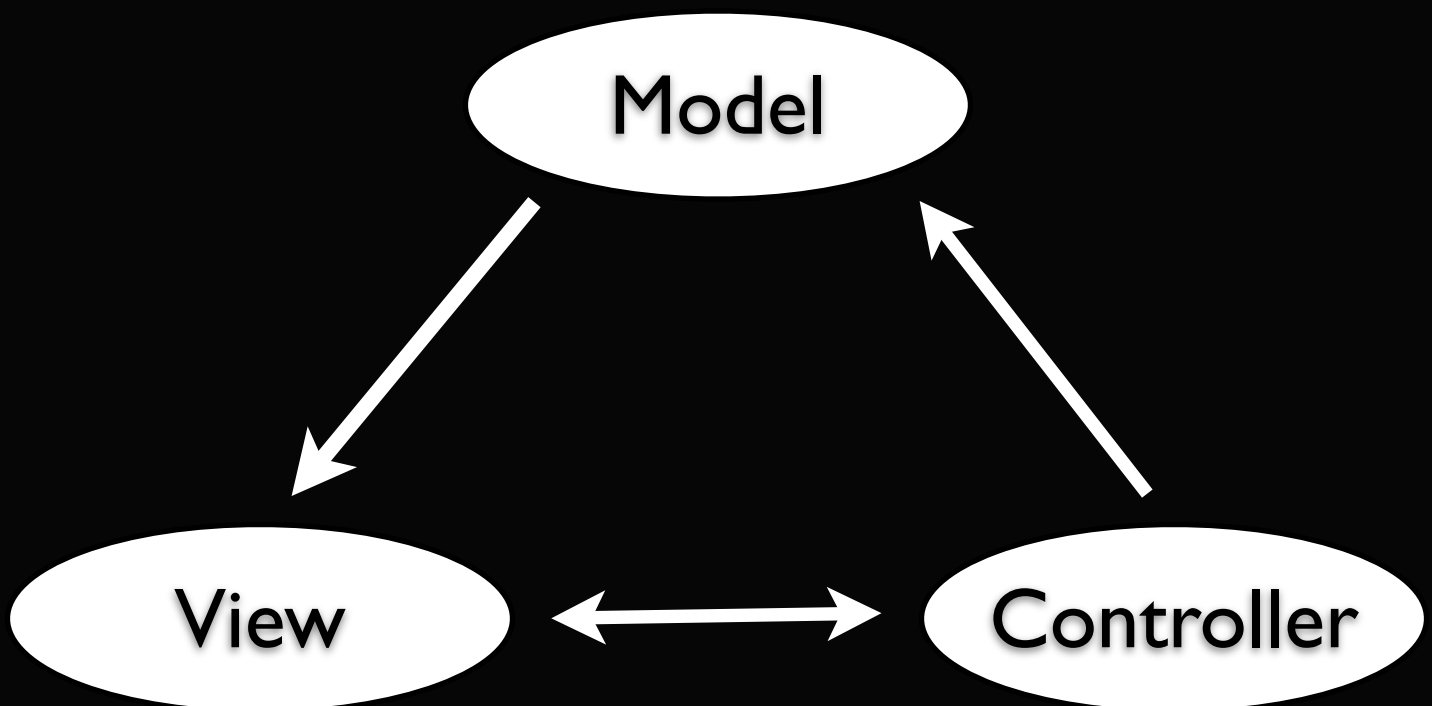
Controller



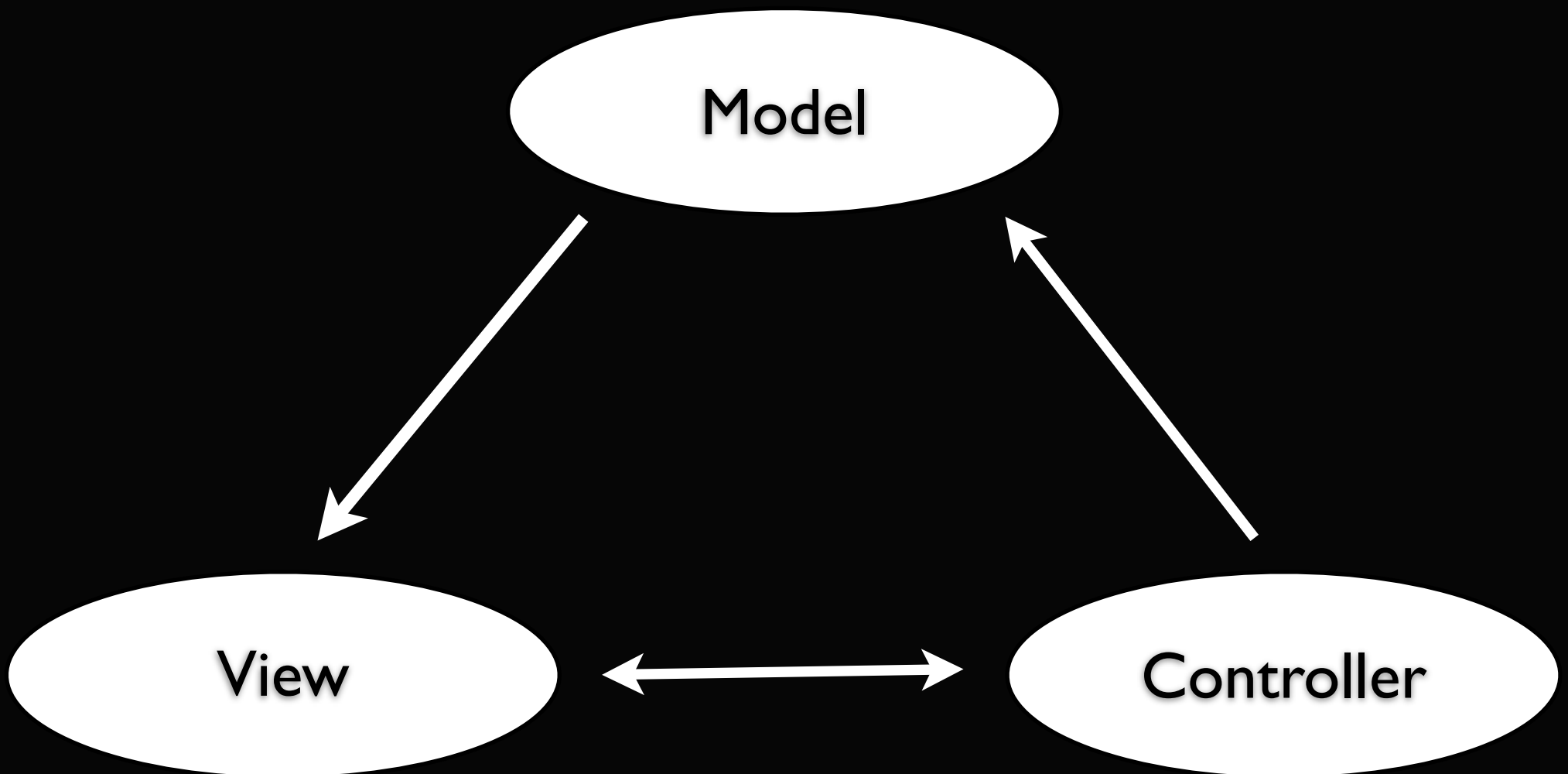
Controller

Handles application and user events

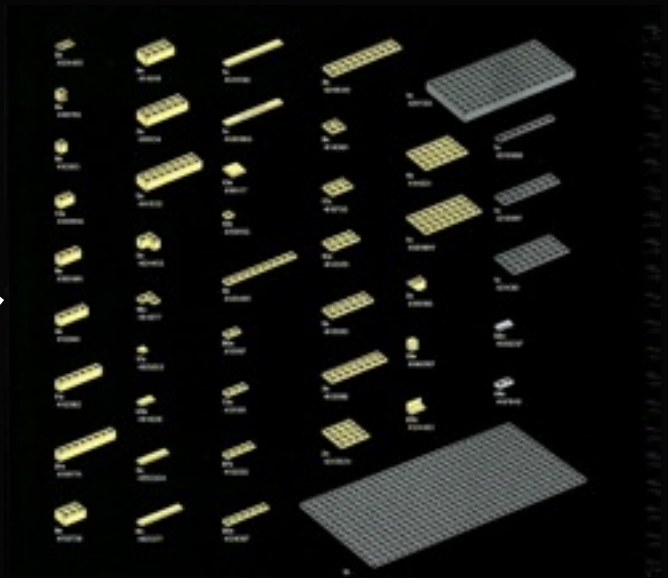
Manipulates the model

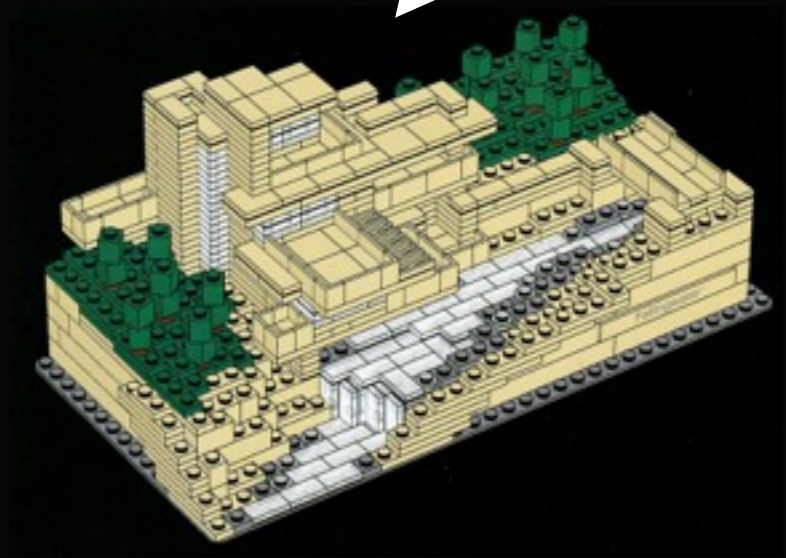


MVC



MVC





MVP & MVVM

Model (Domain Objects)

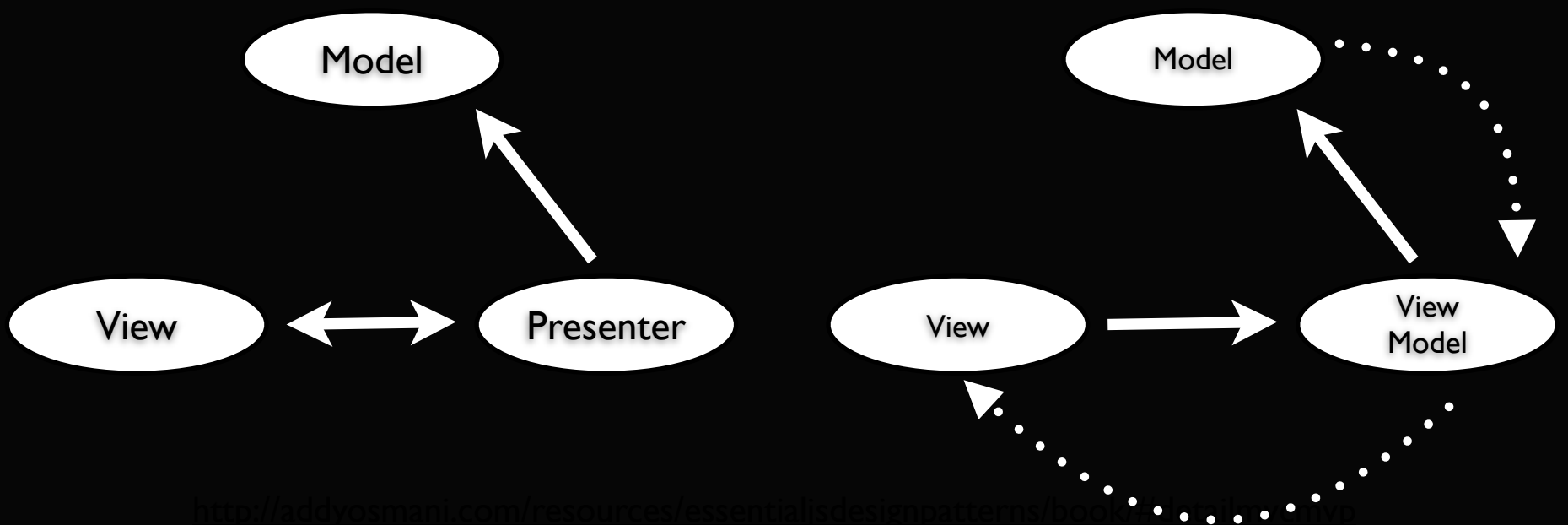
View (UI)

Presenter (Application Logic)

Model (Domain Objects)

View (UI)

View Model (Mediates between the Model and the View via data bindings)



http://addyosmani.com/resources/essentialjsdesignpatterns/book/#detailmvp

MV* is not (just) DOM Manipulation

MV* is much much more than DOM querying & manipulation framework

Some MV* have DOM frameworks baked in

Others have dependancies and/or extensions

JavaScript MV*

Models

Views

Collections

Events

Routers

Controllers

Services

Factories

Filters

Providers

Directives

Objects

Keep Your Truth Out of the DOM

When to use MV*

When building an Single Page Application that communicates with an API or backend data service, where much of the heavy lifting is done on the client

gmail, git, dropbox

When not to use MV*

When building and application that relies heavily on the server for view rendering, and javascript is used mostly for interactive behaviors (lightbox, sliders, galleries, etc)

Benefits of MVC

Better Code Organization & Reusability

Predictability avoids Spaghetti Code

Better for unit testing

Help write large JS applications

Useful for working with API services (RESTful or Otherwise)

Javascript MVC Crippled by choice?

Backbone

Ember

Angular

Knockout

Batman

Dojo

Yui

Spine

+ More every day

http://todomvc.com/

Some Criteria for Selecting a Framework

What is the framework really capable of?

Has the framework been proved in production?

Is the framework mature?

Is the framework flexible or opinionated?

Have you really played with the framework?

Does the framework have a comprehensive set of documentation?

What is the total size of the framework?

What dependencies does the framework have?

Have you reviewed the community around the framework?

Backbone

Models	Atomic data
Views	Controller + View
Collections	Sets of Models + Server connection
Routers	Connects URLs to actions

http://backbonejs.org/

Angular

Modules	Sets of Models, views & Controllers
Models	Atomic data
Views	View
Controllers	Controller

Services, Factories, Filters, Providers, Directives

angularjs.org/

Setting up Backbone

Backbone dependancies:

DOM query & manipulation: jquery - zepto

Templates : Underscore, lowdash

Setting up Angular

Angular dependancies:

None.

The Twitter app

Twitter Search API

dev.twitter.com/docs/api/1.1/get/search/tweets

te.chni.ca/twitter.api/tweet.php

Freebase API

https://developers.google.com/freebase/

Search:

https://www.googleapis.com/freebase/v1/search?filter=(all type:/film/film)&query=

Topic:

https://www.googleapis.com/freebase/v1/topic/m/

Explorer:

http://www.freebase.com/