

— JAVASCRIPT —

MVC. Publisher-Subscriber

MVC

Model-View-Controller

Модель (Model)

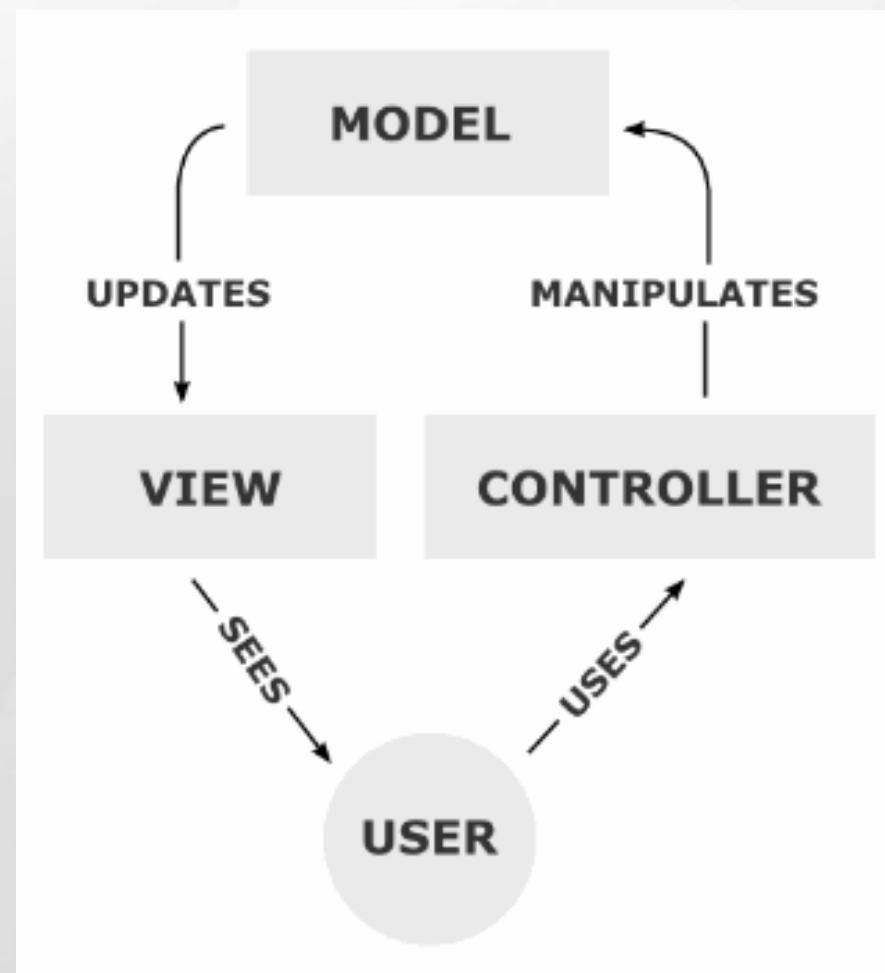
предоставляет данные и реагирует на команды контроллера, изменяя свое состояние.

Представление (View)

отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Контроллер (Controller)

интерпретирует действия пользователя, оповещая модель о необходимости изменений.

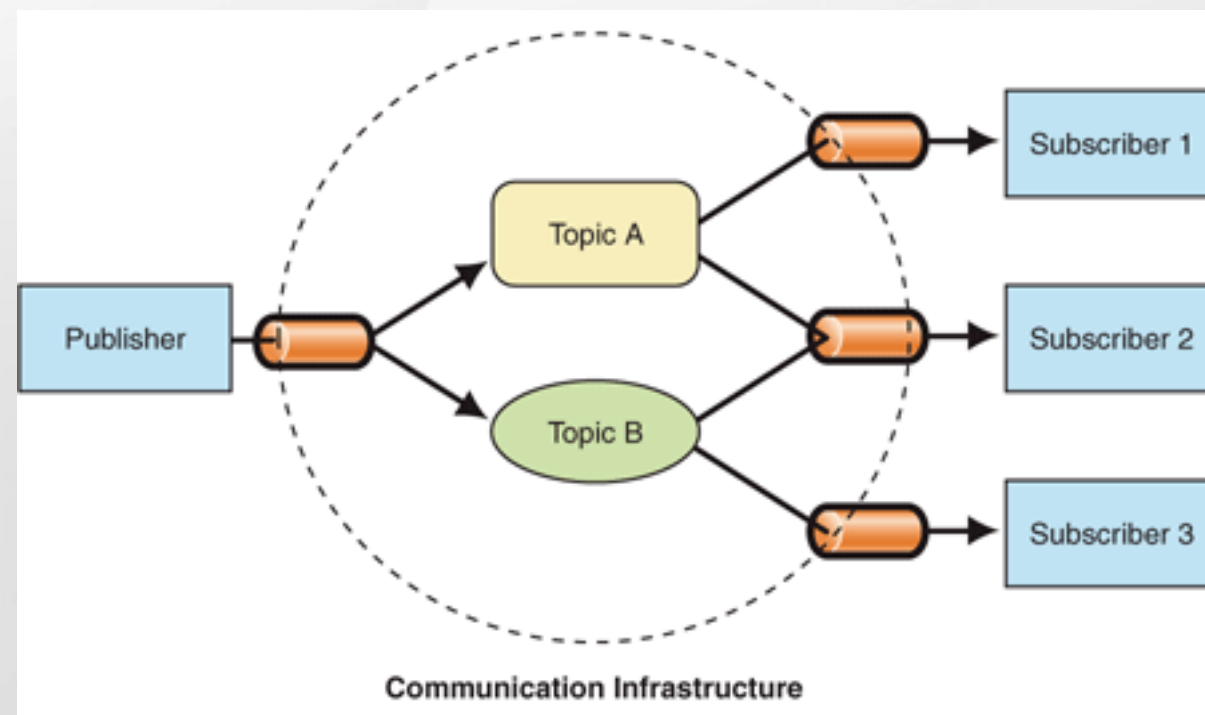


Publisher-Subscriber

Издатель-подписчик

(publisher-subscriber или pub/sub)

поведенческий шаблон проектирования передачи сообщений, в котором отправители сообщений, именуемые издателями (англ. publishers), напрямую не привязаны программным кодом отправки сообщений к подписчикам (англ. subscribers). Вместо этого сообщения делятся на классы и не содержат сведений о своих подписчиках, если таковые есть. Аналогичным образом подписчики имеют дело с одним или несколькими классами сообщений, абстрагируясь от конкретных издателей.



```
1
2▼ (function (win) {
3
4    var Radio = function () {
5        this.topics = {}
6    };
7
8▼    Radio.prototype = {
9▼        on: function (topic, listener) {
10            // create the topic if not yet created
11            if (!this.topics[topic]) {
12                this.topics[topic] = [];
13            }
14
15            // add the listener
16            this.topics[topic].push(listener);
17        },
18
19▼        trigger: function (topic, data) {
20            // return if the topic doesn't exist, or there are no listeners
21            if (!this.topics[topic] || this.topics[topic].length < 1) {
22                return;
23            }
24
25            // send the event to all listeners
26            this.topics[topic].map(function (listener) {
27                listener(data);
28            });
29        }
30    };
31
32    var radio = new Radio();
33    win.radio = radio;
34
35 }(window));
```

— JAVASCRIPT —

MVC. Publisher-Subscriber