

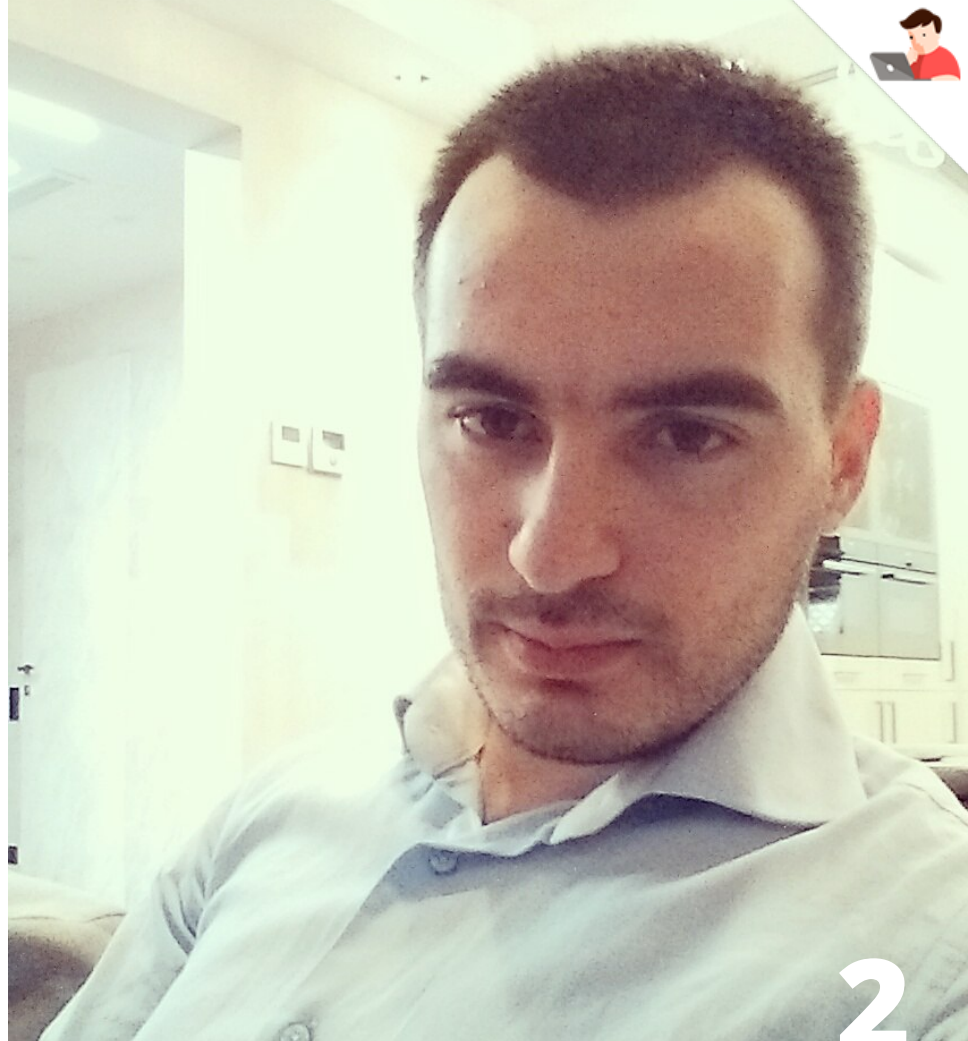
Model-View-*

Ведущий вебинара

Сергей Мелюков

Круг интересов: Backend, Frontend, GameDev, MobileDev

Место работы: Frontend разработчик профессиональных инструментов Avito



Содержание



1. Паттерны семейства Model-View-*
2. MVC
3. MVP
4. MVVM

Паттерны семейства Model-View-*

Паттерны семейства Model-View-*

Паттерны семейства MV*

предназначены для решения задач по разделению логики приложения и его внешнего представления.

Разделяют три основных MV-паттерна:

MVC - Model-View-Controller

MVP - Model-View-Presenter

MVVM - Model-View-ViewModel



The background features a minimalist illustration of a mountain range with several peaks in shades of gray. A single white cloud is positioned on the left side. The entire scene is set against a light gray background. In the top right corner, there is a yellow L-shaped line. In the bottom left corner, there is a yellow L-shaped line. At the bottom of the slide, there is a solid yellow horizontal bar.

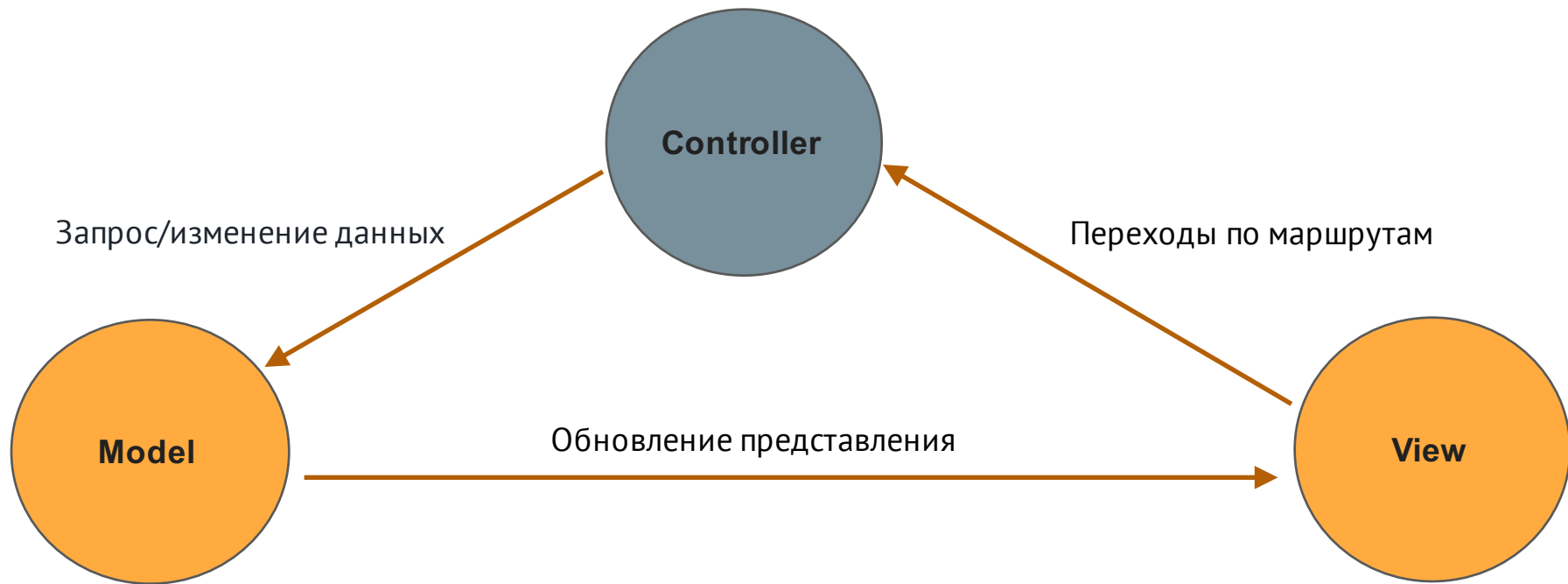
MVC

MVC



Model	бизнес-логика приложения
View	визуальное представление модели
Controller	прослойка между моделью и представлением. Осуществляет взаимодействие между ними

MVC - упрощенно



MVC в web



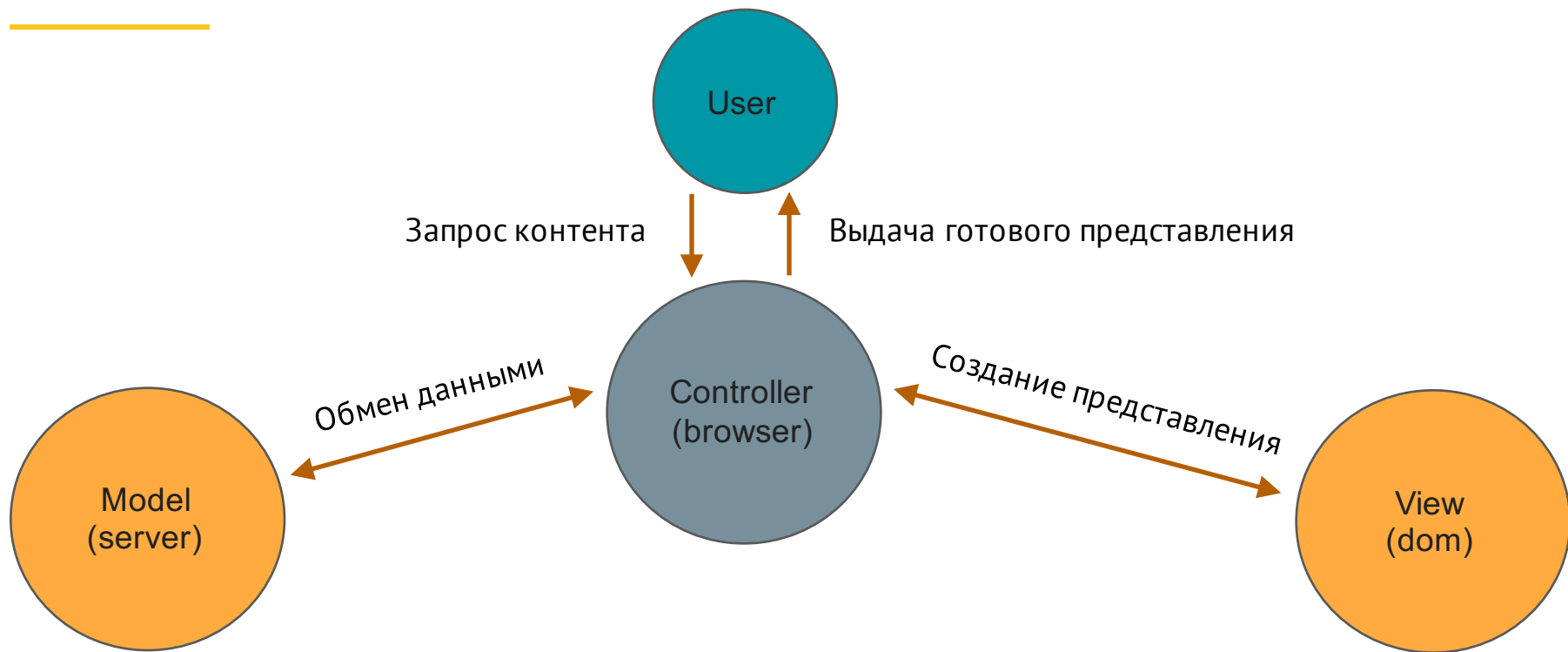
Если попробовать переложить концепцию MVC на работу с сайтом, то можно провести следующие аналогии:

Пользователь просит браузер открыть страницу. Браузер посылает запрос на сервер. Сервер отдает данные в виде html-разметки. Браузер превращает разметку в визуальное dom-дерево. Следовательно:

Model	сервер
View	dom-дерево
Controller	веб-браузер

Это лишь один вариантов абстракции.

MVC - детально



The background features a minimalist illustration of a mountain range with several peaks in shades of gray. A single white cloud is positioned on the left side. The entire scene is set against a light gray background. In the top right corner, there is a yellow L-shaped line. In the bottom left corner, there is another yellow L-shaped line. At the bottom right, the number 11 is displayed in a large, bold, yellow font, partially overlapping a solid yellow horizontal bar that spans the width of the slide.

MVP

MVP

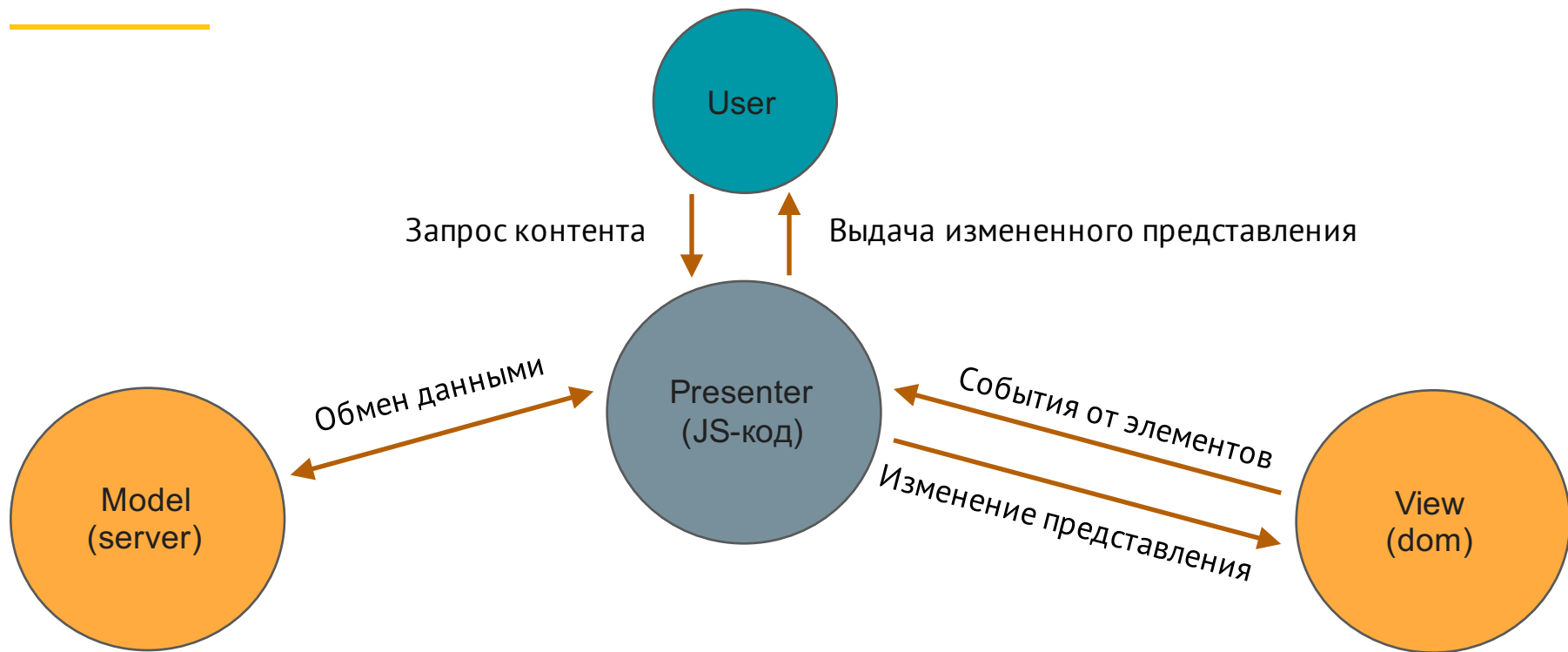


Model	бизнес-логика приложения
View	визуальное представление модели
Presenter	прослойка между моделью и представлением. Осуществляет взаимодействие между ними

Presenter в MVP отличается от Controller в MVC.

Основное отличие в том, что Presenter и View могут общаться между собой. Presenter подписывается на события от View(например клики мышкой) и при поступлении таких событий - обновляет модель. При поступлении ответа от модели - обновляет представление. Таким образом, в отличие от MVC, осуществляется двусторонняя связь между presenter и view.

MVP



The background features a stylized illustration of a mountain range with several peaks. A white cloud is positioned on the left side, partially obscuring one of the mountain peaks. The mountains are rendered in shades of gray, and the sky is a light gray. In the bottom right corner, there is a large yellow number '14'. There are also yellow L-shaped decorative elements in the top right and bottom left corners.

MVVM



Model	бизнес-логика приложения
View	визуальное представление модели
ViewModel	прослойка между моделью и представлением. Осуществляет взаимодействие между ними.

ViewModel отвечает за связывание динамическое данных между моделью и представлением.

На самом деле, грань между MVVM и MVP довольно тонка, т.к. MVVM - это усовершенствованный MVP. Схематически, взаимодействие у MVVM и MVP одинаково. Разница в реализации: MVVM реализует более гибкий слушатель событий от представления.

Как результат, MVVM позволяет реализовать динамическое связывание данных (data binding), используя декларативный подход.

Время ваших вопросов