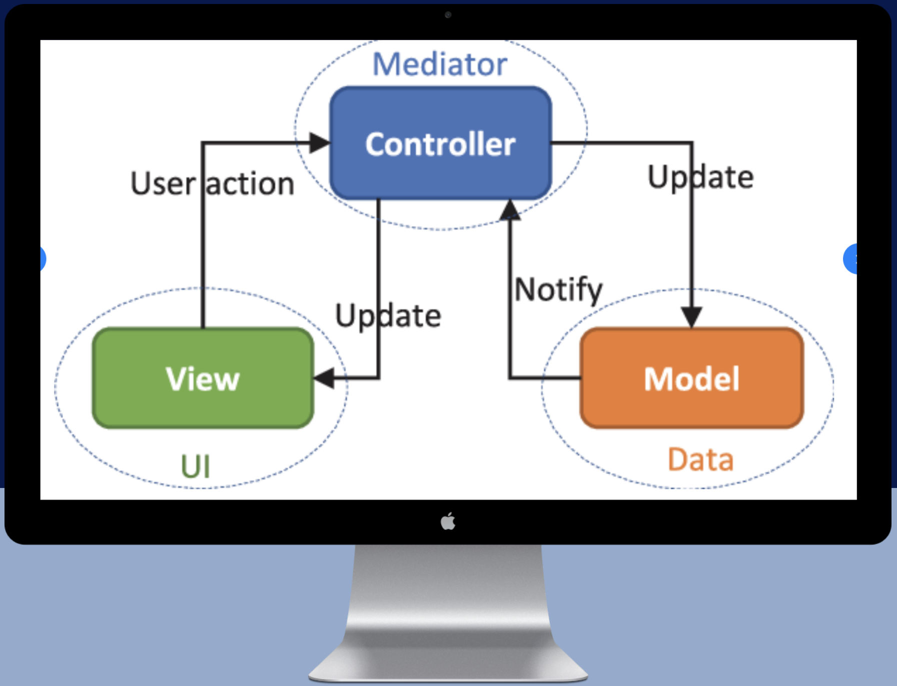


Model View Controller (MVC)

Alberto Cruz Luis

alu0101217734@ull.edu.es

Jeremy Manuel Luis León

alu0101244587@ull.edu.es

Content Table

- What is Model View Controller?
- History
- Skeleton of MVC
- Model
- View
- Controller
- Tic-Tac-Toe in MVC
- Advantages & Disadvantages
- Alternatives To MVC
- Bibliography

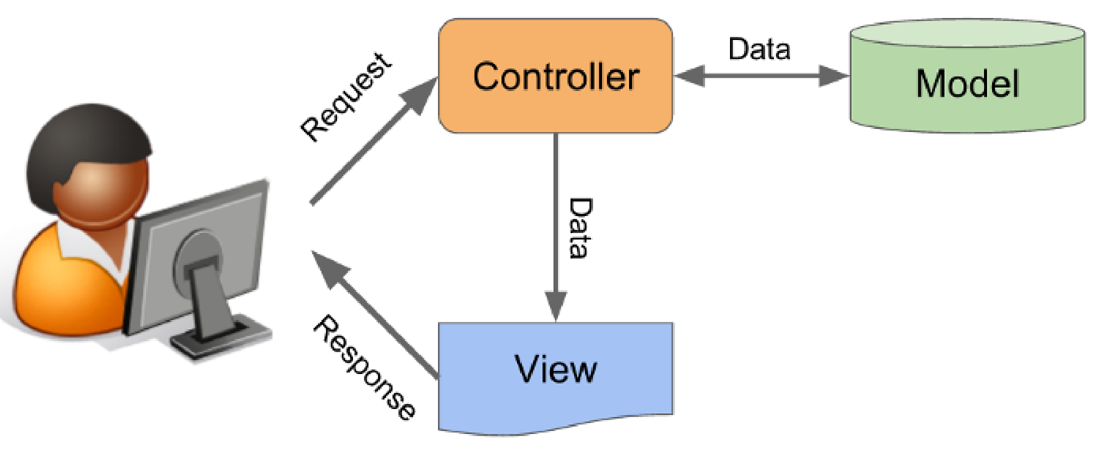
What is Model View Controller?

It is a software architecture pattern, which separates the data and mainly the business logic of an application from its representation and the module in charge of managing events and communications.

The **model** represents the data, and does nothing else.

The **view** displays the model data, and sends user actions to the controller.

The **controller** provides model data to the view, and interprets user actions such as button clicks.

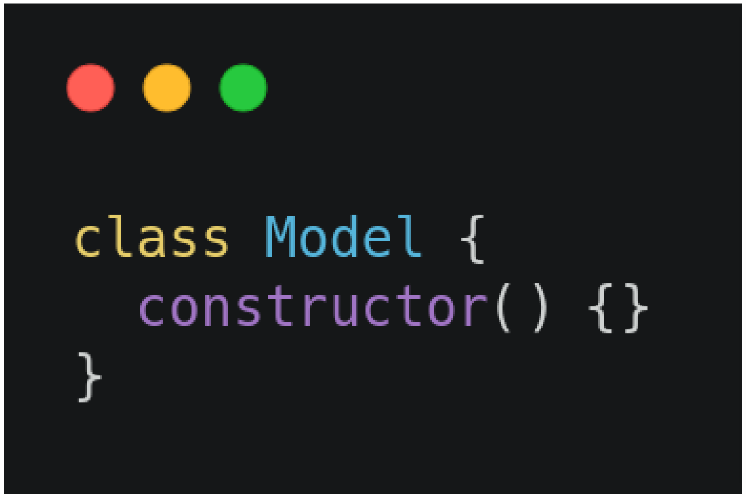


History

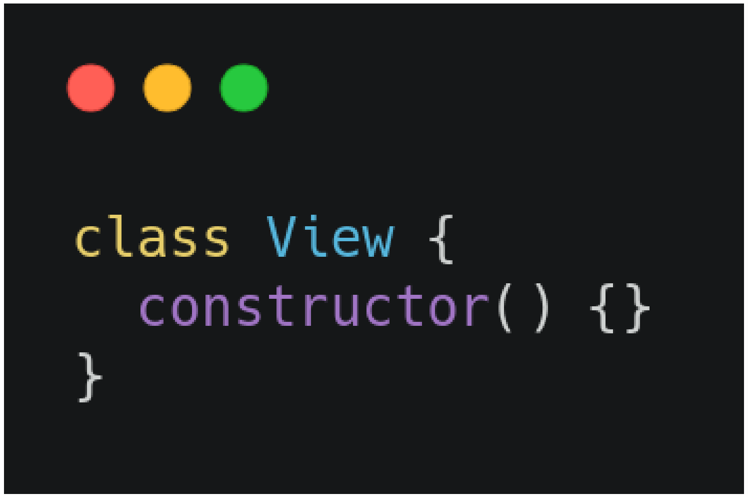
1 Introduced in 1979 by Trygve Reenskaug into Smalltalk-79

2 The MVC pattern has subsequently variants such as hierarchical model–view–controller (HMVC), model–view–adapter (MVA), model–view–presenter (MVP), model–view–viewmodel (MVVM)

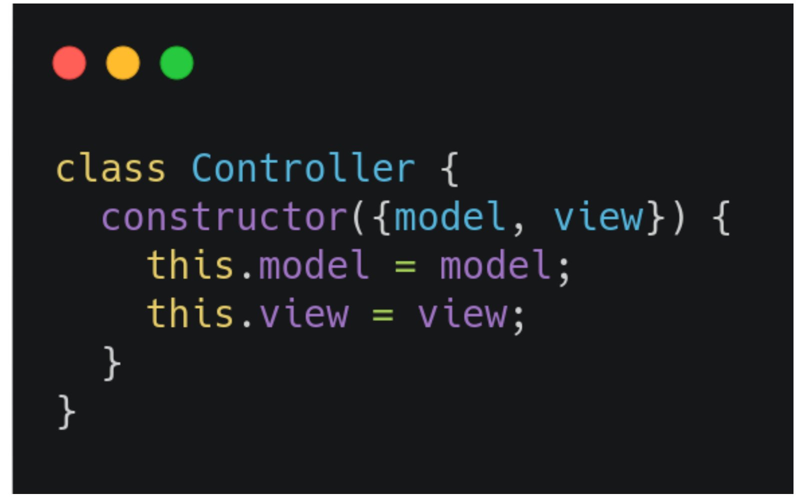
Skeleton of MVC



```
class Model {
  constructor() {}
}
```

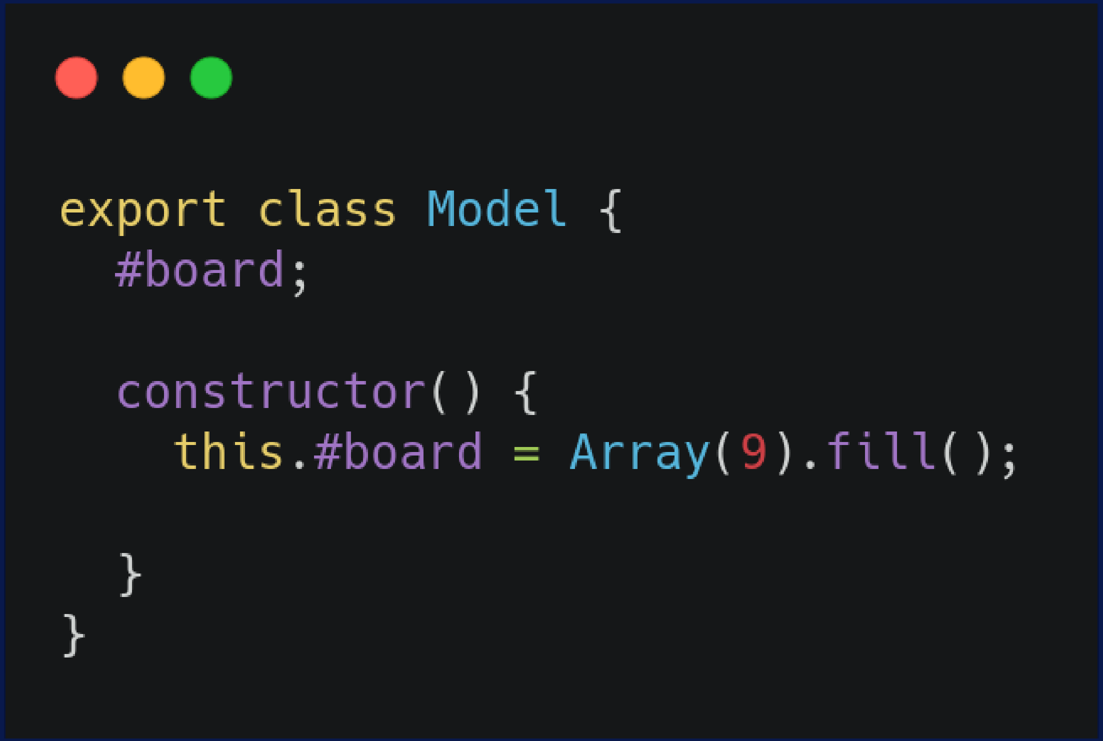


```
class View {
  constructor() {}
}
```



```
class Controller {
  constructor({model, view}) {
    this.model = model;
    this.view = view;
  }
}
```

Model

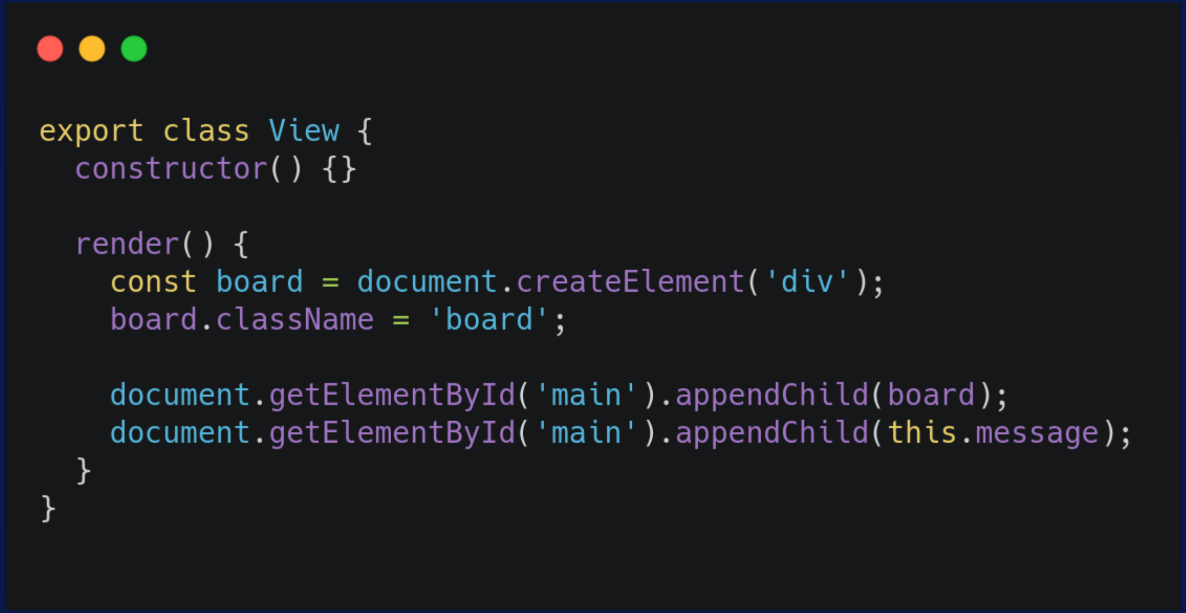


```
export class Model {
  #board;

  constructor() {
    this.#board = Array(9).fill();

  }
}
```

- Is responsible for managing the data of the application
- It responds to request from the view and it also responds to instructions from the controller to update itself
- It is the lowest level of the pattern which is responsible for maintaining data
- The model represents the application core.
- It is also called the domain layer

View

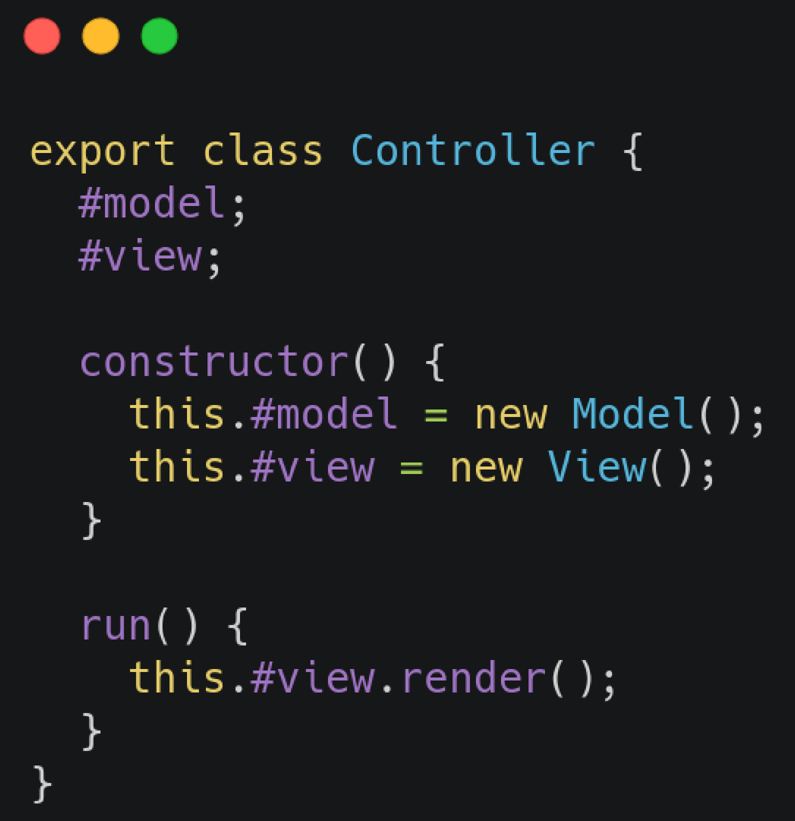
The View, or user interface, is made up of the information that is sent to the client and the interaction mechanisms with it.



```
export class View {
  constructor() {}

  render() {
    const board = document.createElement('div');
    board.className = 'board';

    document.getElementById('main').appendChild(board);
    document.getElementById('main').appendChild(this.message);
  }
}
```

- Receive data from the model and display it to the user.
- It is solely responsible for the graphical interface and how the end user will see the data

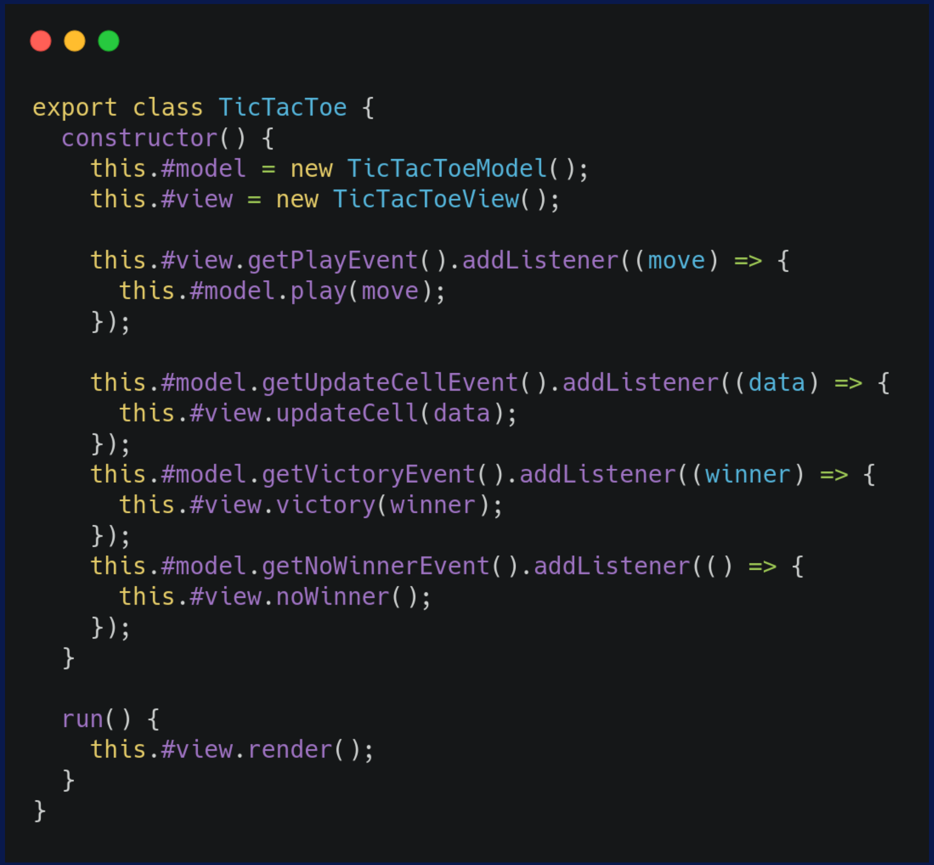
Controller



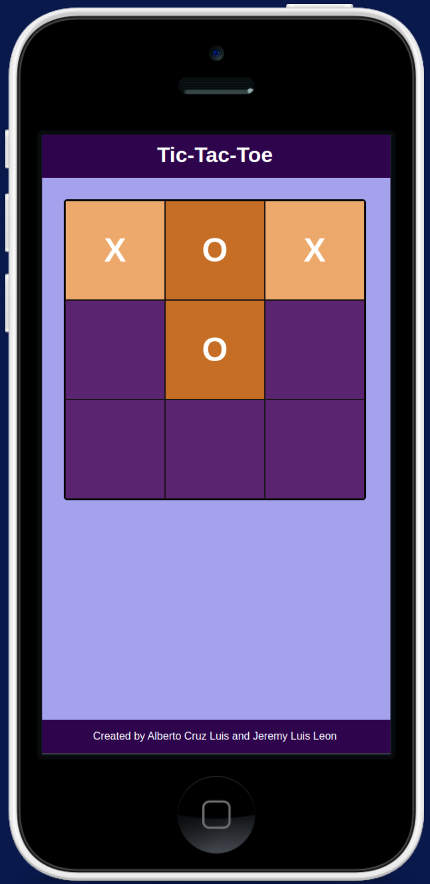
```
export class Controller {
  #model;
  #view;

  constructor() {
    this.#model = new Model();
    this.#view = new View();
  }

  run() {
    this.#view.render();
  }
}
```

- It is the link between the view and the model
- Is responsible for receiving and responding to events

Tic-Tac-Toe in MVC



```
export class TicTacToe {
  constructor() {
    this.#model = new TicTacToeModel();
    this.#view = new TicTacToeView();

    this.#view.getPlayEvent().addListener((move) => {
      this.#model.play(move);
    });

    this.#model.getUpdateCellEvent().addListener((data) => {
      this.#view.updateCell(data);
    });
    this.#model.getVictoryEvent().addListener((winner) => {
      this.#view.victory(winner);
    });
    this.#model.getNoWinnerEvent().addListener(() => {
      this.#view.noWinner();
    });
  }

  run() {
    this.#view.render();
  }
}
```



Advantages & Disadvantages

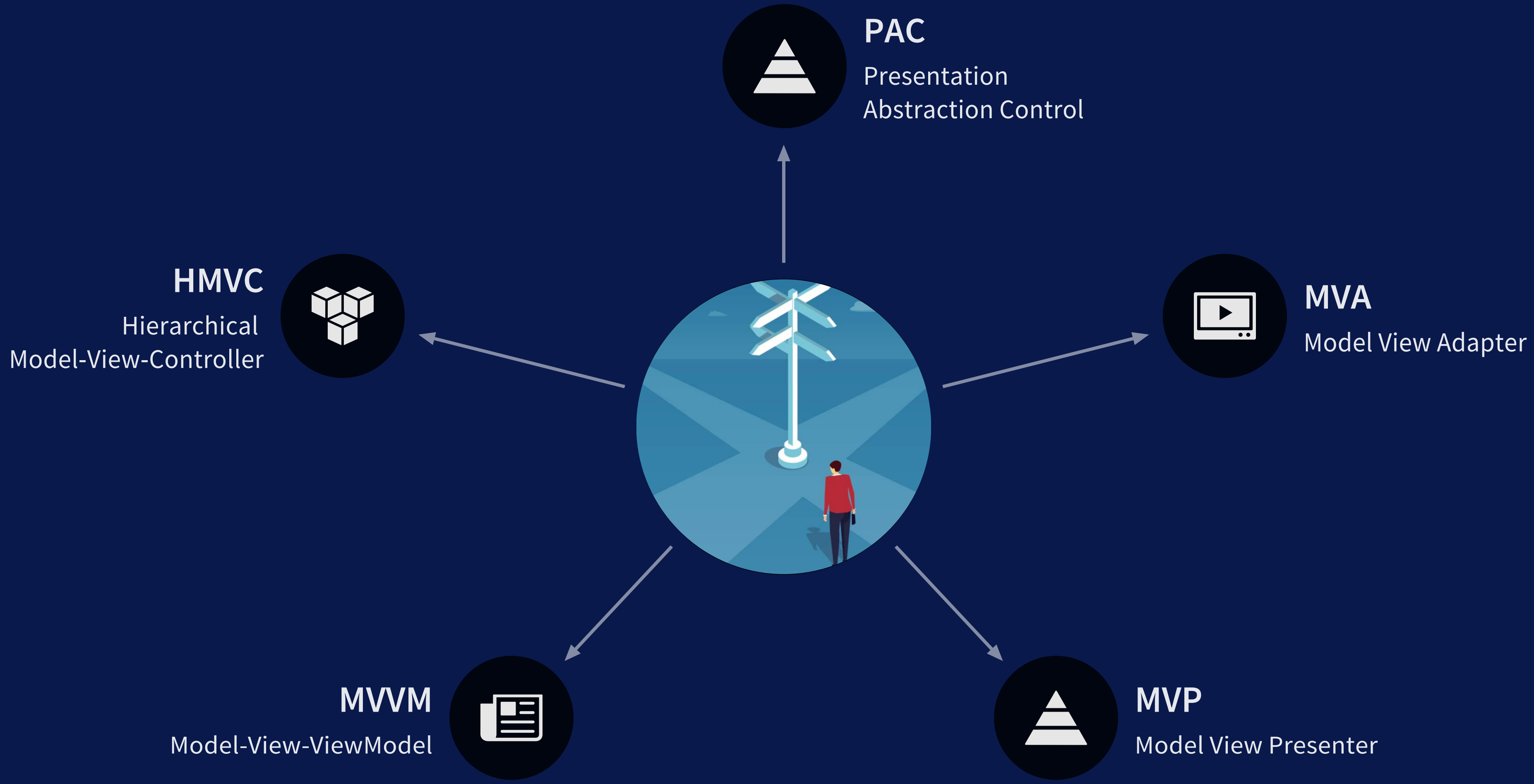
Advantages

- Clear separation between presentation logic and business logic
- Single responsibility principle
- Modifications does not affect the entire model
- Fast development
- Modular implementation
- The developer does not have to worry about requesting that the views be updated

Disadvantages

- Hard to understand the architecture
- Must have strict rules on methods
- Increased complexity
- MVC design pattern requires greater dedication in the early stages of development.

Alternatives to MVC



PAC

Presentation
Abstraction Control

HMVC

Hierarchical
Model-View-Controller

MVA

Model View Adapter

MVVM

Model-View-ViewModel

MVP

Model View Presenter

Bibliography

https://www.taniarascia.com/javascript-mvc-todo-app/

https://solucionesci.wordpress.com/2016/07/12/patrones-de-diseno-web-alternativas-a-mvc/

https://programmerclick.com/article/388054092/

https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm

https://www.tomdalling.com/blog/software-design/model-view-controller-explained/

https://www.tutorialspoint.com/mvc_framework/mvc_framework_controllers.htm

https://www.codeproject.com/Articles/858310/Controller-in-MVC

https://www.oreilly.com/library/view/learning-javascript-design/9781449334840/ch10s02.html

Thanks for your Attention

Alberto Cruz Luis

alu0101217734@ull.edu.es

Jeremy Manuel Luis Leon

alu0101244587@ull.edu.es