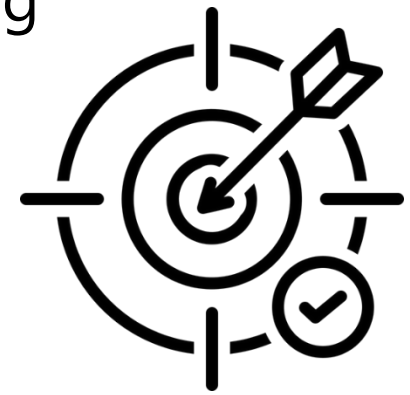
Web Framework: MVC Pattern

Web Programming and Testing

Mario Simaremare, S.Kom., M.Sc.
Program Studi Sarjana Sistem Informasi
Institut Teknologi Del

Objectives

- The objective of this session is the following:
  - The students are able to elaborate the benefit of developing solution on top of web framework.
  - The students are able to select and use contemporary web framework.



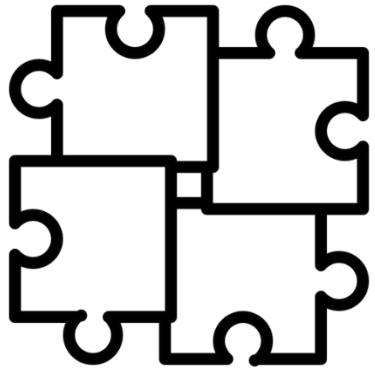
Outlines

1. Motivation
2. MVC architectural pattern.
3. Web framework.

Motivation

Web Dev in the Age of Dinosaur

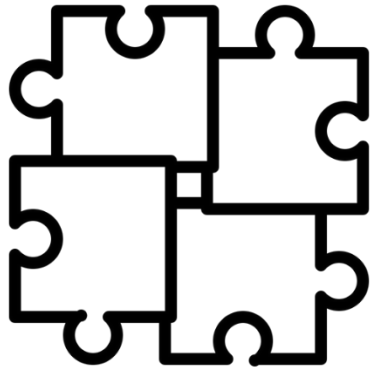
- Years ago, web developers tend to use in house components to construct their application.
  - Developed on top of experiences.
  - Components get polished through time.
  - Well tested, proven, and production ready,



- Problems:
  - Less standard, convention, rules, etc.
  - Component are reused only in the internal dev team.
  - New boy needs time to learn.

Driven by Community

- Community grows and working together to ease the development process.
  - Avoid reinventing the wheel.
  - Use the existing components.
  - Focus on the business.
- Specific language ecosystems.
  - e.g. https://www.php-fig.org/



MVC Architectural Pattern

Model-View-Controller Pattern

- MVC is a popular pattern that commonly applied to application with graphical user interface (GUI).
  - Segregating components based on their responsibilities.
  - Web and desktop applications.

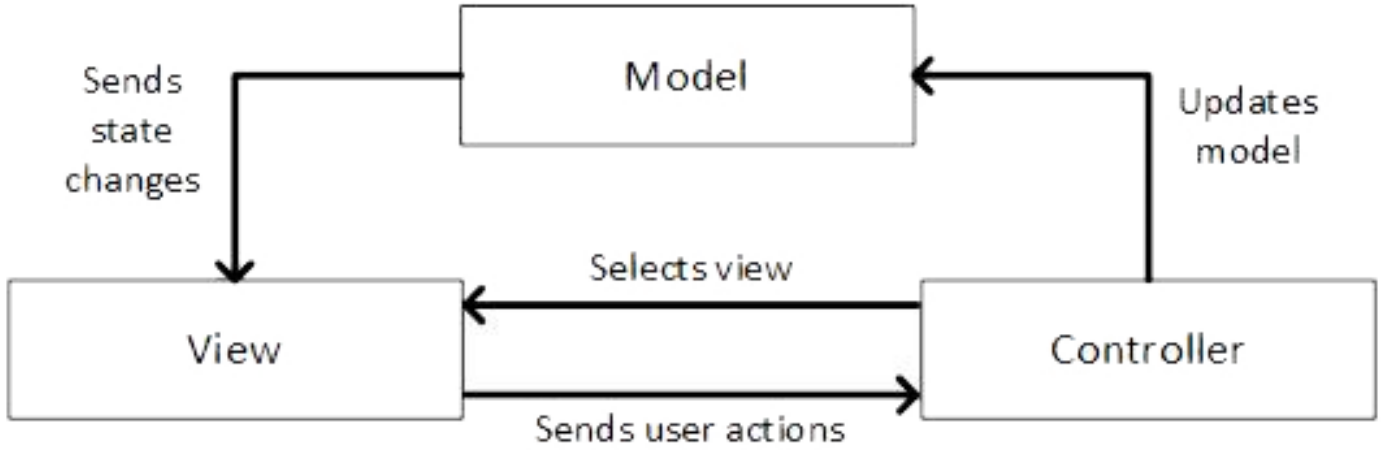
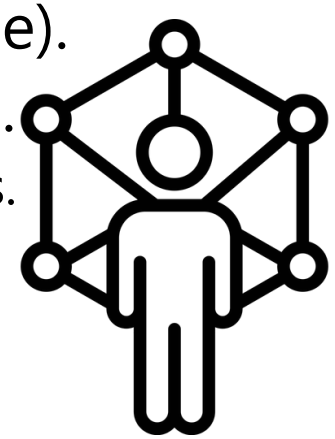


Image source: Ingeno, J. (2018). Software Architect's Handbook. Packt

Model-View-Controller Pattern

- **Model** represents business data, record, or states.
  - The actual data are stored in persistence storage (e.g. database).
  - Operations that related to business data are also defined here.
    - Incld. adding, modifying, removing data, and other specific operations.



- Patterns related to data:
  - Active Record, Data Mapper, Data Gateway

Model-View-Controller Pattern

- **View** takes care of how the state (data) should be presented.
  - Different audience might require different presentation.
    - structure, format, etc.
  - It directly interacts with the application users.
    - User interactions are received by view and forwarded to the controller.

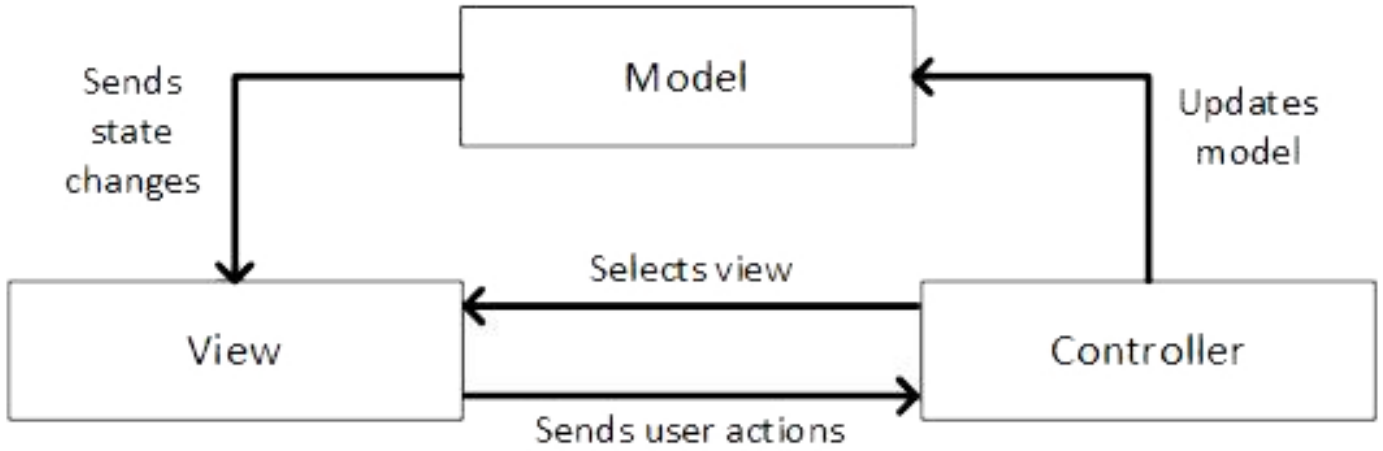


Image source: Ingeno, J. (2018). Software Architect's Handbook. Packt

Model-View-Controller Pattern

- **Controller** orchestrates the application behavior based on the events and the accompanying data.
  - Business logic is defined here.
  - It decides which models and views should be used.

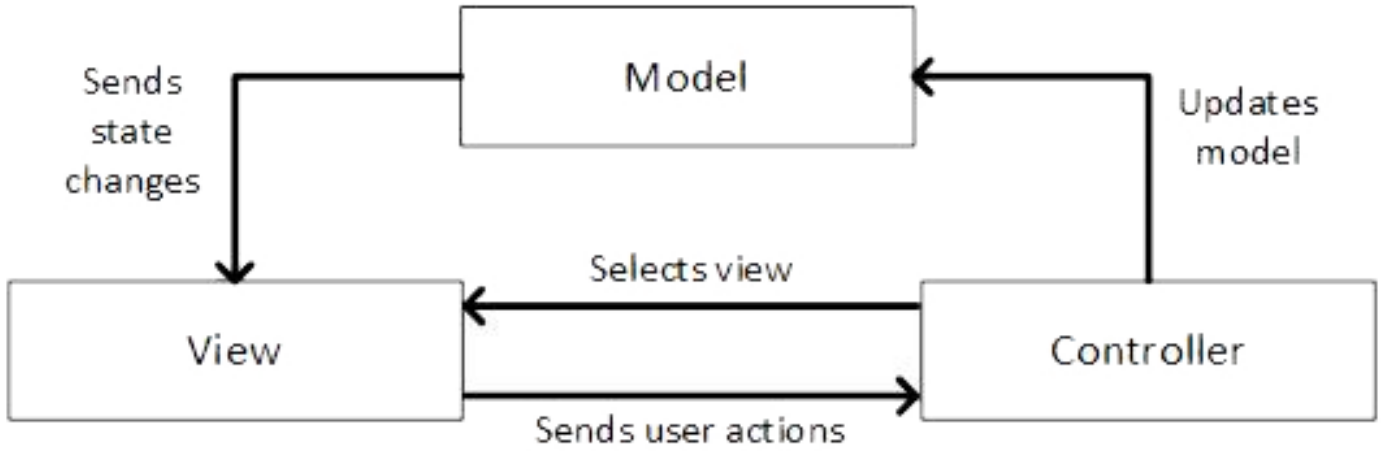
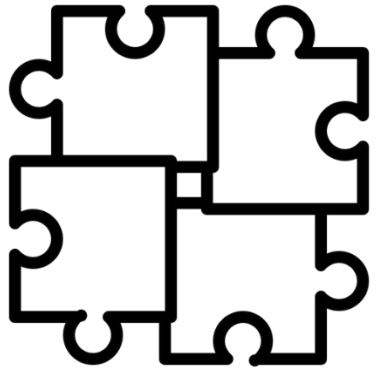


Image source: Ingeno, J. (2018). Software Architect's Handbook. Packt

Advantages and Disadvantages

- Advantages:
  - Modular solution with loosely coupled components.
  - Isolated problem.
  - Changes to particular aspect is way simpler.
  - Developer specialization.
    - Frontend and backend developers.
  - A chance for rapid development cycle.

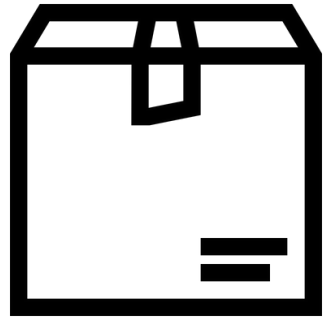


- Disadvantages:
  - In the case of simple project, MVC might be too much.

Web Framework

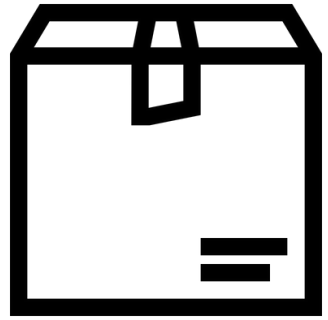
Web Framework

- It is a set of toolkit that provides a skeleton for web application.
  - Most of today's framework follows the MVC architectural pattern.
  - Either full-featured MVC or partial.



Advantages and Disadvantages

- Advantages:
  - Accelerate application development.
  - Solid structure, well tested, and community-baked.
  - Best practices and standards compliance (e.g. PSR).



- Disadvantages:
  - Might take some time to learn.
  - Too complex for simple problem.

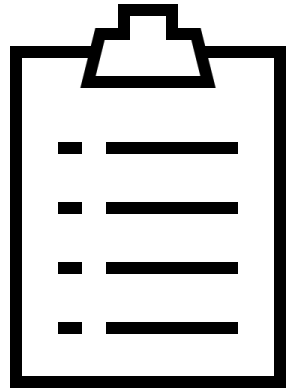
Partial vs. Fullstack

- Partial framework concentrates on a specific layer:
  - Focus on the controlling (Lumen, Silex, Slim).
  - Focus on the view aspect (Twig, Blade).
  - Focus on the data layer (Doctrine, CycleORM, Medoo).

- Fullstack handles every part of a web architecture.
  - Yii, Symfony, Laravel, Spring, Django, etc.

To-dos

1. Choose your web framework and use it in your project.
   - Consult with your TA for insights.
   - Read the documentation before choosing.



References

Srinivasan, M. (2012). Web Technology: Theory and Practice. Pearson.

Ingeno, J. (2018). Software Architect's Handbook. Packt

Thank you