# Leveraging the MVC pattern for enterprise WordPress development

WORDCAMP SYDNEY 2024

# Intro

💼 Head of Engineering @ The Code Co.

🏢 The Code Co. specialises in enterprise WordPress development

🏗️ Building sites with WordPress for over 14 years

💡 Passionate about solving problems with code

💪 Love the flexibility of the WordPress ecosystem

# What is MVC?

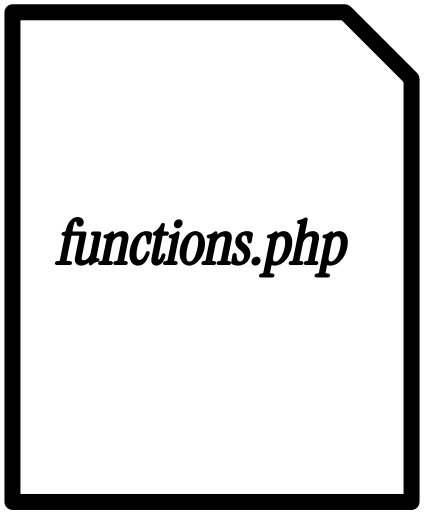
# WPMVC

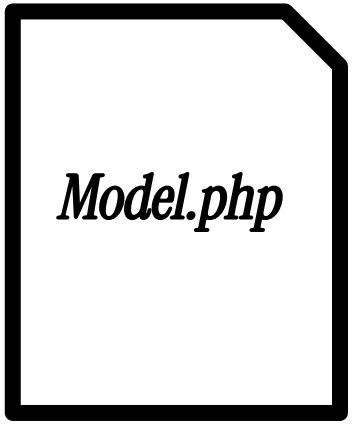
# Learnings

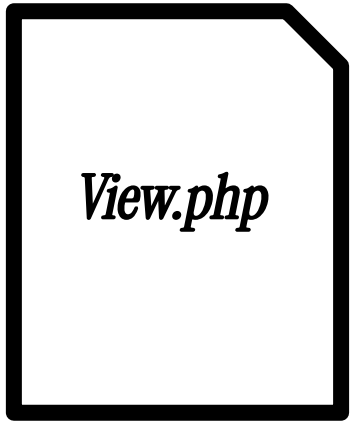
# What is MVC?

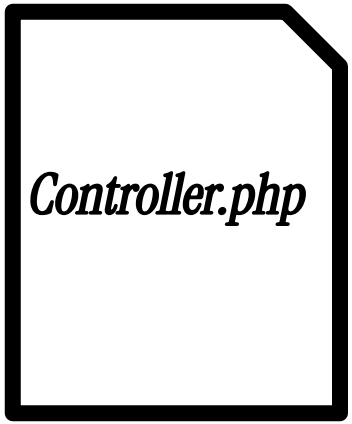
# Design pattern

# Structure


functions.php


Model.php


View.php


Controller.php

Model

View

Controller

# Model

- Representation of the data in your database
- Models can be used to access WP objects
- Think WP_Post, WP_Term, WP_User, WP_Comment
- Flexible

# Model/Movie.php

- Extension of WP_Post
- Properties: `$post->ID`
- Custom properties are usually stored as post meta
- Contains helper methods
- Use Factory classes to create Models

```
$movie_release_date = get_post_meta( $post_id, 'release_date', true );

$movie = new Movie();
$movie_release_date = $movie->get_release_date();
```

```php
<?php
/**
 * Movie.php
 */

namespace ExampleSite;

use \WPMVC\Model\GenericPostModel;

class Movie extends GenericPostModel {

    const RELEASE_DATE_META_KEY = 'release_date';

    public function get_release_date(): mixed {
        return $this->get_meta( self::RELEASE_DATE_META_KEY, true );
    }

}
```

```php
$movie_factory = new MovieFactory();
$movie         = $movie_factory->get_by_id( $post_id );
$release_date  = $movie->get_release_date();
```

# View

- UI layer, displays the data
- Frontend: WordPress Theme
- Use Cases:
  - Admin UI/templates
  - Transactional emails
  - Templates that don’t rely heavily on frontend styles/components

# template/emails/MovieAddedNotification.php

- Notification email sent to admins when a Movie is added
- Passed the Movie Title
- Registered via a Controller

```
$view = new ThemeableView(
    $this->config,
    'emails/MovieAddedNotification'
);

$view->set_param( 'movie_title', 'Interstellar' );

$view->render();
```

```php
<?php
/**
 * MovieAddedNotification.php
 */

?>
<h1>Movie added</h1>

<p>
This is an example of a parameter passed to the template:
</p>

<code>
Movie Title = "<?php echo esc_html( $this->get_param( 'movie_title' ) ); ?>"
</code>
```

# Controller

- Contains core application logic, linking the Models and Views
- Used to customise/add WordPress behaviour
- add_action()
- add_filter()
- Modify other plugins
- Groups related logic together

# Controller/MoviePostType.php

- Every controller in WPMVC has a set_up() method, called on the plugins_loaded hook
- Registers the post type
- Registers a save_post hook
- Easier to find related code

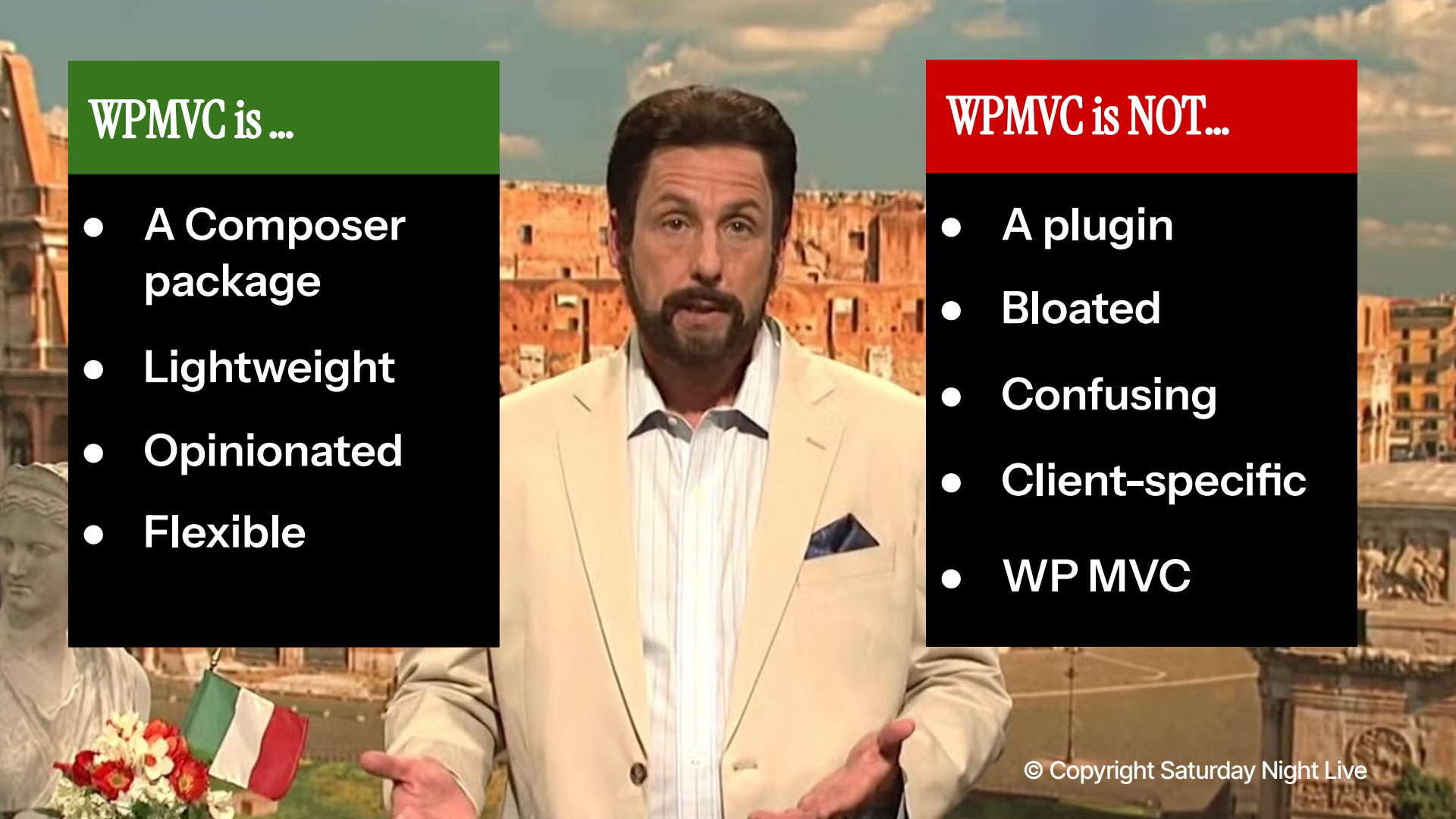
```php
use \WPMVC\Core\Controller;

class MoviePostType extends Controller {

    public function set_up() {
        add_action( 'init', [ $this, 'register_post_type' ] );
        add_action( 'save_post', [ $this, 'on_save' ] );
    }

    public function register_post_type() {
        // Register post type here.
    }

    public function on_save() {
        // Handle on save logic here.
    }
```

Model = *Data*

View = *Theme*

Controller = *Hooks*

# WPMVC*


WPMVC is ...
A Composer package
Lightweight
Opinionated
Flexible
WPMVC is NOT...
A plugin
Bloated
Confusing
Client-specific
WP MVC
© Copyright Saturday Night Live

# A typical project

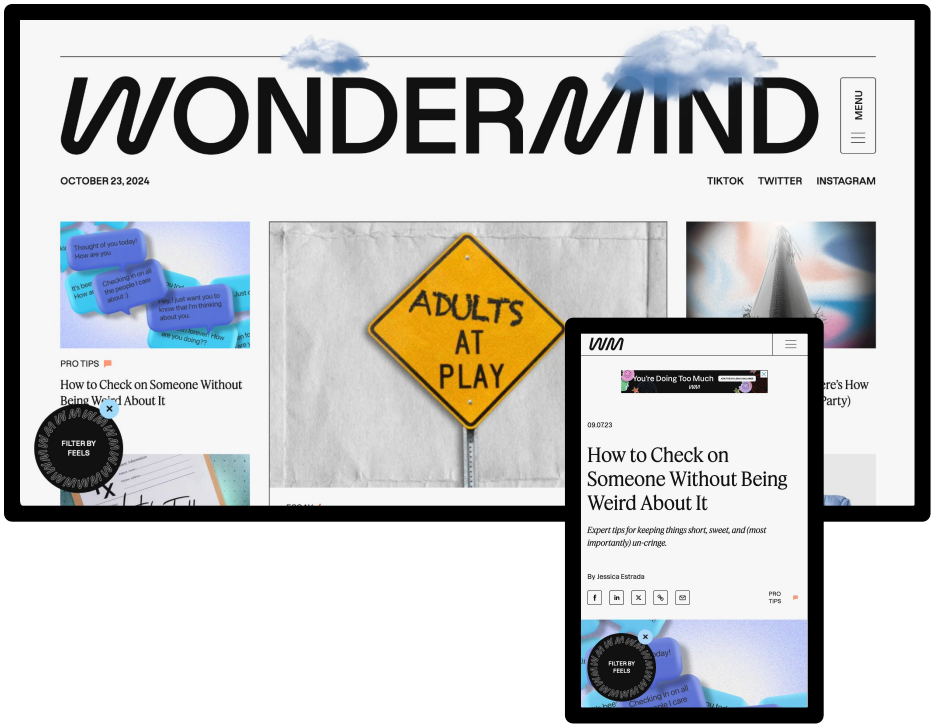
wp-content/mu-plugins

- *project-alpha.php*
- project-alpha/
  - *boot.php*
  - *composer.json*
  - config/
  - template/emails/*MovieAddedNotification.php*
  - src/
    - Controller/
      - *RewriteController.php*
    - Function/
      - *Autoload.php*
      - *Theme.php*
    - Library/
      - *PayPalAPI.php*
    - Model
      - Post/
        - *Movie.php*
        - *MovieFactory.ph*
      - Taxonomy/
        - *Genre.php*

# WPMVC for real life

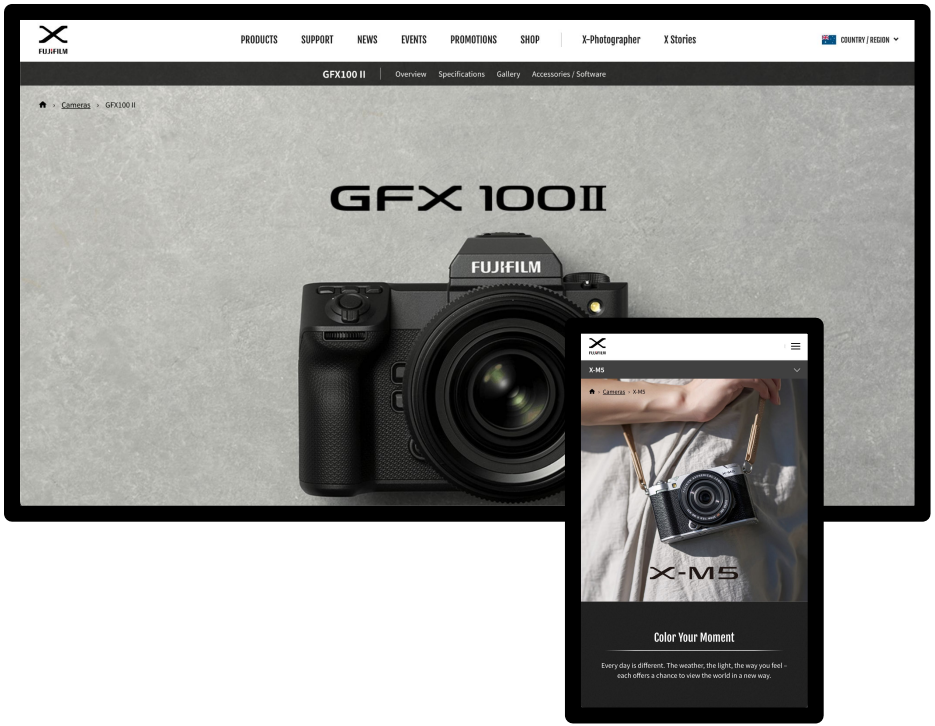
# Small-ish project

- Mental health startup
- Frontend-heavy
- Custom MU Plugin and theme
- ~10 controllers
- 3 custom post types
- 6 custom taxonomies



# Large project

- Global camera powerhouse
- 40+ languages
- ~100 controllers
- 5 custom MU Plugins
- 25 post types
- 10 taxonomies
- Combination of Multisite + Multilingual WPML
- 7+ developers



# Learnings & benefits

# Benefits

✅ Consistency across projects

✅ Easier to find code

✅ Easier to maintain and include

✅ Less mess and tech debt

# Learnings

- 💡 Keep it simple
- 💡 Structure = good

# What’s next?

Dependency injection/traits

CLI tooling & bootstrapping

Better docs

thecode.co/wpmvc