WordCamp Europe

Kraków 2026

WordCamp Europe
Kraków 2026

A HANDS-ON GUIDE TO FSE & BLOCK THEMES

BUILD YOUR DEVELOPER PORTFOLIO

Dejan Rudic Vranic

WordCamp Europe
Kraków 2026

DEJAN RUDIC VRANIC

Senior Backend Engineer and founder of Studio Agnis with 10+ years of expertise in the WordPress ecosystem. I specialize in backend development, custom solutions, and complex enterprise-grade integrations.

WordCamp Europe
Kraków 2026

FSE IS NOT FOR SERIOUS DEVELOPERS.

You've heard it. Maybe you've said it.

- "theme.json feels like XML soup"
- "I lose precision"
- "My clients will break it"
- "Where is my template hierarchy?"

WordCamp Europe
Kraków 2026

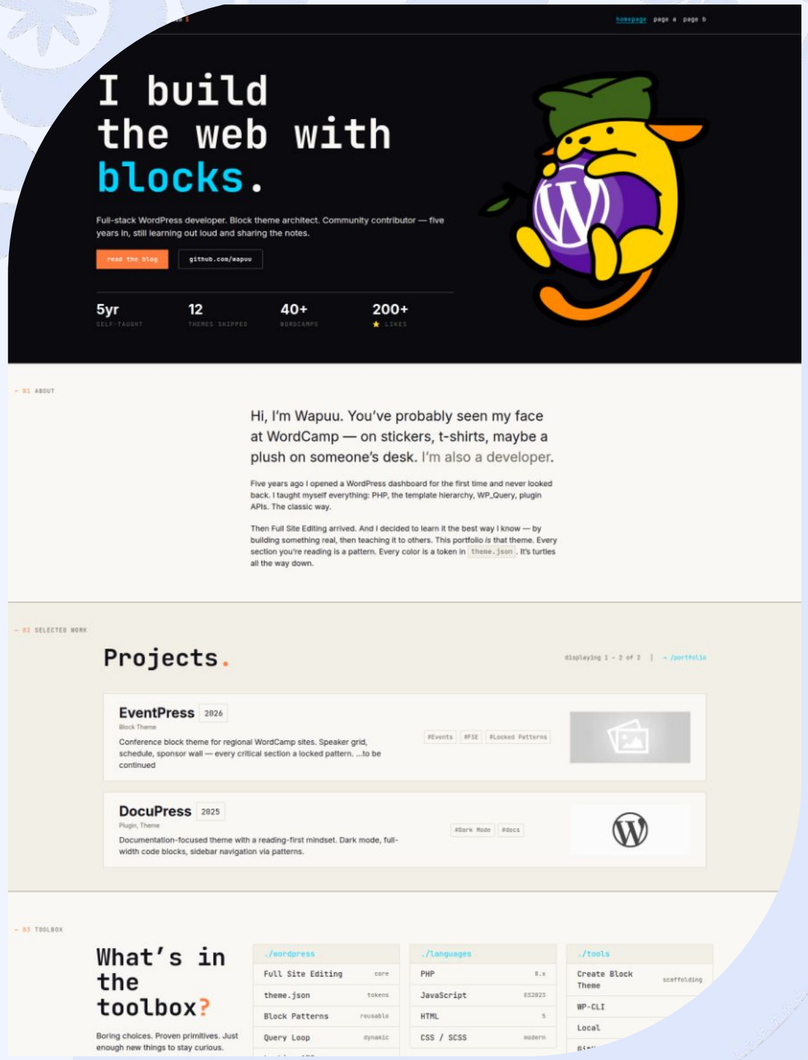
THE COST OF WAITING

- Block themes are now the default. Twenty Twenty-Two onward.
- Classic theme reviews on wp.org are slowing.
- Clients ask for editor flexibility. Builders provide it. So can you — natively.
- The longer you wait, the bigger the gap.

WordCamp Europe
Kraków 2026

A REAL THEME. FROM SCRATCH. IN 75 MINUTES.

- A working block theme
- Dark mode developer portfolio
- 5 portfolio projects in a Query Loop
- Locked, client-proof layout



WordCamp Europe
Kraków 2026

THE FICTIONAL CLIENT: WAPUU

- Persona: self-taught WP dev, 5 years
- Wants: portfolio, blog, project showcase
- Doesn't want: a designer to maintain it
- This is who we're building for

WordCamp Europe
Kraków 2026

You wrote	Now it lives in
header.php	parts/header.html
footer.php	parts/footer.html
WP_Query loop	Query Loop block
get_template_part()	Pattern
wp_enqueue_style()	theme.json
page.php, single.php	templates/*.html

WordCamp Europe
Kraków 2026

WHAT'S ACTUALLY NEW

1. theme.json — design tokens as data, not code
2. Block templates — HTML files instead of PHP
3. Patterns — composable UI you can register
4. Query Loop block — WP_Query with a UI
5. Locking APIs — your client governance layer

WordCamp Europe
Kraków 2026

75 MINUTES, 5 GIT BRANCHES

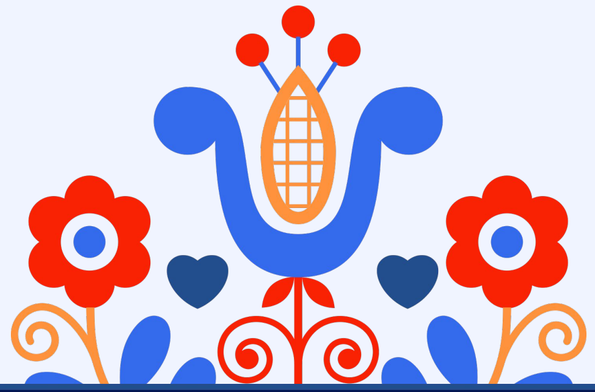
Checkpoint	Branch	What's on it
~10 min	01-start	Empty theme via plugin
~25 min	02-styles	theme.json design system
~35 min	03-patterns	Patterns
~50 min	04-portfolio-archive	Query Loop
~65 min	05-governance	Locking, content-only

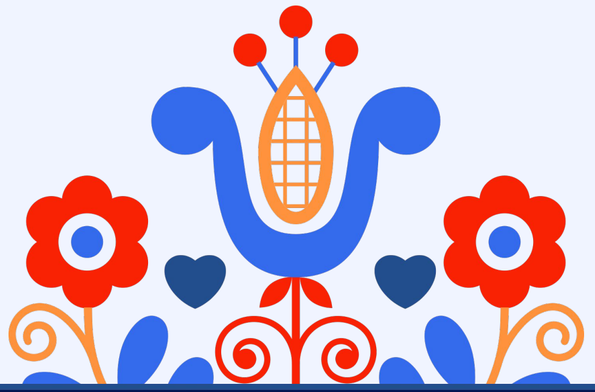
https://github.com/dvranic-code/portfolio-fse

WordCamp Europe
Kraków 2026

01_start

Scaffolding without writing a single file



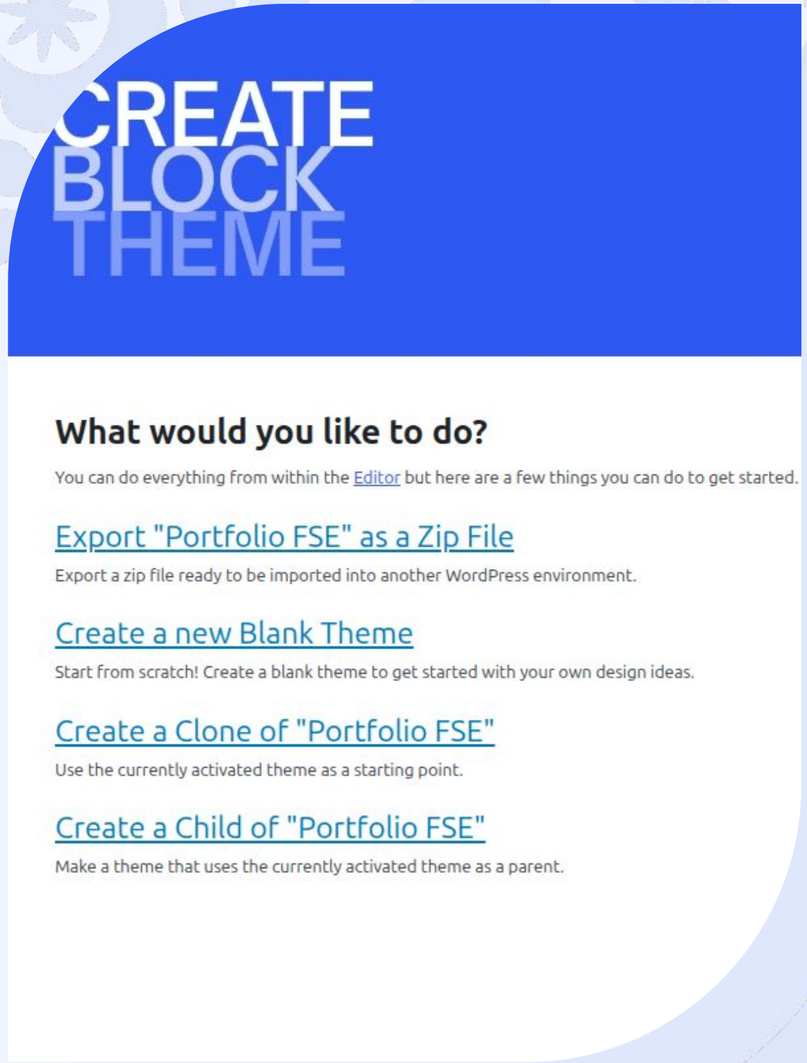


WordCamp Europe
Kraków 2026

CREATE BLOCK THEME

- Maintained by the WordPress core team
- Free on wp.org
- Generates, clones, exports block themes from the editor
- We'll use "Create a new Blank Theme"

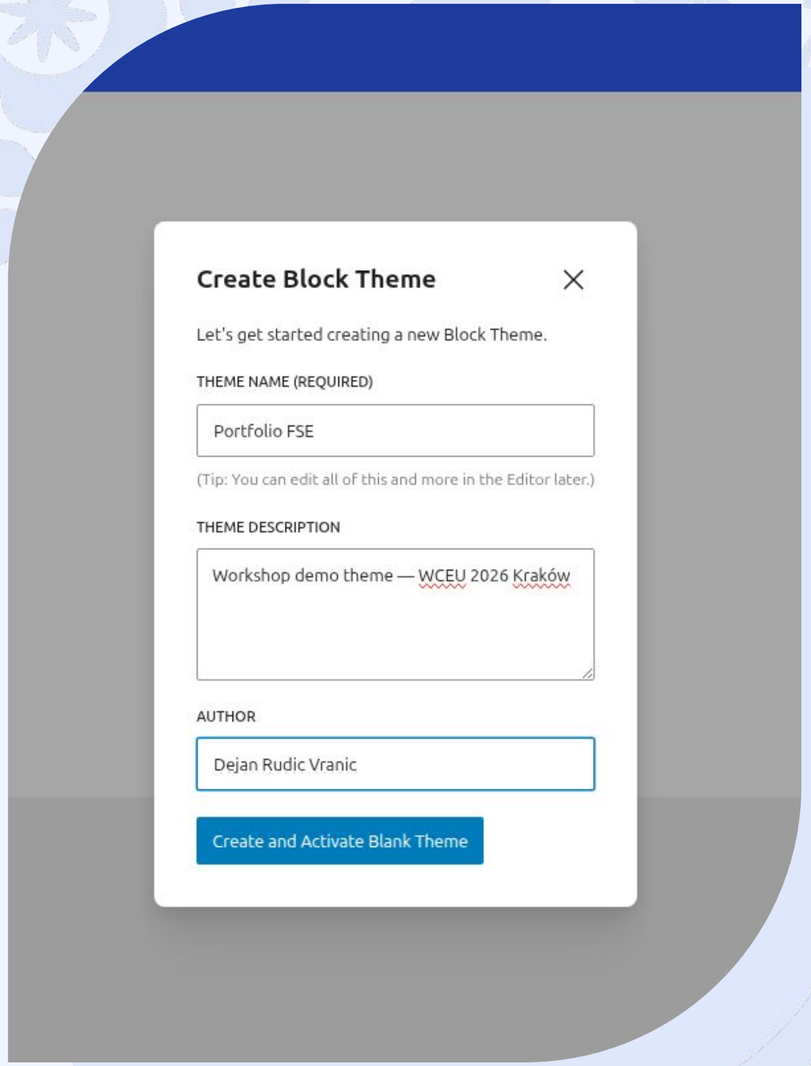
https://wordpress.org/plugins/create-block-theme/



WordCamp Europe
Kraków 2026

CREATE BLOCK THEME

- **Theme Name:** Portfolio FSE
- **Description:** Workshop demo theme — WCEU 2026 Kraków
- **Author:** your name



WordCamp Europe
Kraków 2026

WHAT THE PLUGIN GENERATED

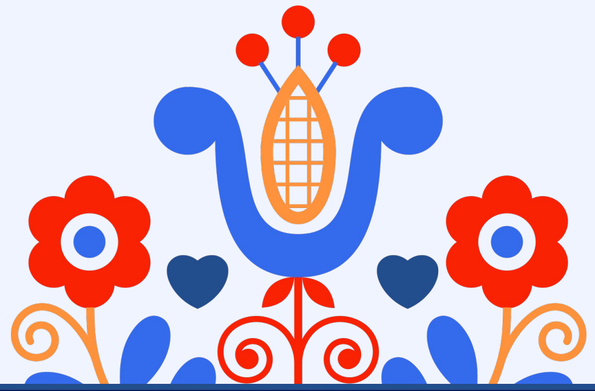
```
portfolio-fse/
├── style.css
├── theme.json
├── parts/
│   ├── header.html
│   └── footer.html
├── templates/
│   └── index.html
├── readme.txt
└── screenshot.png
```

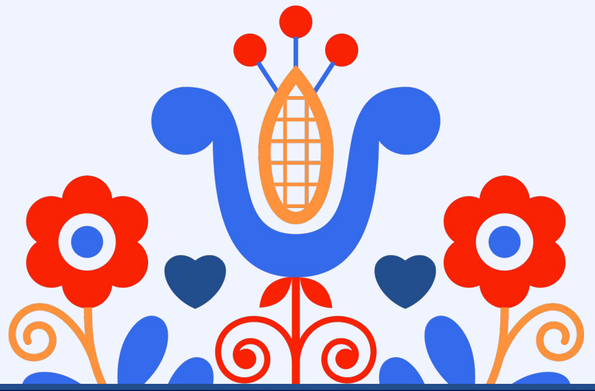
- 7 files
- ~57 KB
- 0 lines of PHP

WordCamp Europe
Kraków 2026

02_styles

The design system in one file





WordCamp Europe
Kraków 2026

WHAT IS THEME.JSON?

- Single source of truth for design tokens
- Replaces dozens of wp_enqueue_* calls in functions.php
- WordPress auto-generates CSS variables from it
- Editors read from it. Frontend reads from it. Same source.

```
portfolio-fse/
├── style.css      ← theme metadata only
├── theme.json     ← the entire design system
├── parts/
│   ├── header.html
│   └── footer.html
├── templates/
│   └── index.html
├── readme.txt
└── screenshot.png
```

WordCamp Europe
Kraków 2026

TWO HALVES. DON'T CONFUSE THEM.

SETTINGS

```
"settings": {
   "color": { "palette": [...] },
   "typography": { "fontSizes": [...] }
}
```

STYLES

```
"styles": {
   "typography": {
      "fontFamily": "var:preset|font-family|sans"
   }
}
```

Settings = the menu. Styles = what you ordered.

WordCamp Europe
Kraków 2026

COLORS - SEMANTIC, NOT VISUAL

- 12 colors mapped from the design reference
- Slugs are semantic (`accent-primary`), not visual (`cyan`)
- Why: swap `accent-primary` from cyan to orange without touching block markup
- `defaultPalette: false` removes WP's built-in colors

```
"color": {
  "defaultPalette": false,
  "palette": [
    {
      "slug": "background",
      "color": "#0a0a0f",
      "name": "Background"
    },
    {
      "slug": "accent-primary",
      "color": "#00d4ff",
      "name": "Accent Primary"
    },
    ...
  ]
}
```

WordCamp Europe
Kraków 2026

TYPOGRAPHY - SELF-HOSTED, TWO SLUGS

- Two families: Inter (sans), JetBrains Mono (mono)
- 4 weights each, woff2 only
- Self-hosted in `assets/fonts/`
- No Google Fonts at runtime
- Slugs are semantic again: `sans` and `mono`

8 woff2 files. ~320 KB total. Loaded by WordPress automatically.

```
"fontFamilies": [
  {
    "name": "Inter",
    "slug": "sans",
    "fontFamily": "Inter, -apple-system, ...",
    "fontFace": [
      {
        "fontFamily": "Inter",
        "fontWeight": "400",
        "fontStyle": "normal",
        "src": ["file:./assets/fonts/inter/inter-400.woff2"]
      },
      ...
    ]
  },
  ...
]
```

WordCamp Europe
Kraków 2026

FLUID TYPOGRAPHY VIA `clamp()`

- One toggle: `"fluid": true`
- Each size: min + max
- WordPress generates the `clamp()` math
- No media queries needed

```
"typography": {
  "fluid": true,
  "fontSizes": [
    {
      "slug": "huge",
      "name": "Huge",
      "size": "6rem",
      "fluid": {
        "min": "4rem",
        "max": "6rem"
      }
    }
  ]
}
```

WordCamp Europe
Kraków 2026

SPACING - ONE LINE, SIX TOKENS

- Geometric scale: each step = previous × 1.5
- Six steps from `~0.3rem` to `~2.25rem`
- Editor shows them in every block's spacing dropdown
- Replaces Tailwind's `p-1`, `p-2`, `p-4`, `p-8`...

```
"spacing": {
  "spacingScale": {
    "operator": "*",
    "increment": 1.5,
    "steps": 6,
    "mediumStep": 1,
    "unit": "rem"
  }
}
```

WordCamp Europe
Kraków 2026

THE 4 TOGGLES YOU'LL FORGET

Without these, WordPress core defaults pollute your design system.

Toggle	What it hides
`defaultPalette: false`	WP's built-in colors
`defaultGradients: false`	WP's built-in gradients
`defaultDuotone: false`	WP's duotone filters
`defaultFontSizes: false`	WP's `S/M/L/XL` font sizes

All four go inside their respective `settings` block.

WordCamp Europe
Kraków 2026

REFERENCE, DON'T REPEAT

- `styles` should never contain literal CSS
- Reference `settings` slugs via `var:preset|...`
- WordPress compiles to `var(--wp--preset--*)`

```
var:preset|font-family|sans
───  ──────  ───────────  ────
 │     │         │          │
 │     │         │          └─ slug from settings
 │     │         └─ category
 │     └─ "I'm a preset reference"
 └─ "I'm a CSS variable reference"
```

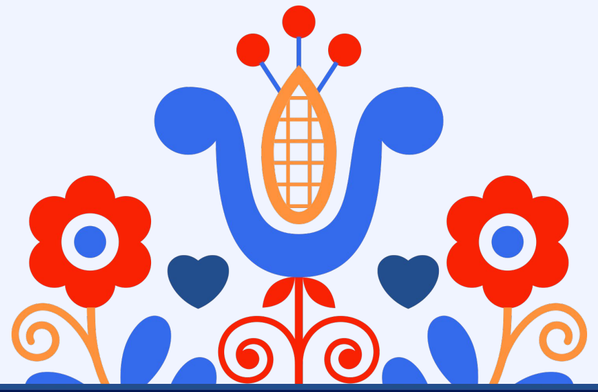
```
"styles": {
  "typography": {
    "fontFamily":
"var:preset|font-family|sans"
  },
  "elements": {
    "h1": {
      "typography": {
        "fontFamily":
"var:preset|font-family|mono"
      }
    }
  }
}
```

WordCamp Europe
Kraków 2026

03_patterns

From `get_template_part()` to

`<!-- wp:pattern /-->`



WordCamp Europe
Kraków 2026

WHAT IS A PATTERN?

```php
// Classic PHP era

<?php
   get_template_part('parts/hero');
?>
```

- File include, runtime
- HTML hardcoded inside PHP
- Every change = code deploy
- Reuse tied to file location

```
// FSE era


<!-- wp:pattern {"slug":"portfolio-fse/hero"} /-->
```

- Block reference, parsed at render
- HTML lives as block markup
- Editable in Site Editor
- Reuse via slug (location-agnostic)

Patterns are **reusable block compositions**, not includes.

WordCamp Europe
Kraków 2026

WHAT IS A PATTERN?

```
<?php
/**
 * Title: Hero                              ← what users see
 * Slug: portfolio-fse/hero                 ← unique ID, theme-prefixed
 * Categories: portfolio-fse                ← inserter grouping
 * Description: Homepage hero section.      ← preview hover text
 * Keywords: hero, intro, landing           ← search terms
 * Block Types: core/post-content           ← starter for which contexts
 */
?>
<!-- wp:group ... -->
   <!-- block markup body -->
<!-- /wp:group -->
```

Header docblock = control panel. | Body = block markup.

WordCamp Europe
Kraków 2026

AUTO-REGISTRATION. NO PHP.

WordPress scans `patterns/*.php` on `init` hook. Header docblock is read. Pattern auto-registered.

```
themes/portfolio-fse/
├── functions.php
└── patterns/
    ├── hero.php                ← auto-registered
    ├── about-section.php       ← auto-registered
    └── work-list.php           ← auto-registered
```

No `register_block_pattern()` calls needed. Drop file, refresh editor, pattern exists.

Exception: Pattern **categories** must be registered in functions.php. WP doesn't infer them.

WordCamp Europe
Kraków 2026

WHERE DOES PHP STILL LIVE?

What	Why
`register_block_pattern_category()`	Custom inserter category
`register_post_type()`	Portfolio CPT
`register_taxonomy()`	Technologies + Project Types
`register_block_style()`	Primary/Secondary button variants
`wp_enqueue_style()`	Block themes don't auto-enqueue `style.css`

Five hooks, ~80 lines. Classic theme `functions.php` is ~400. **80% reduction is real.**

WordCamp Europe
Kraków 2026

show in rest: true-
THE FSE HANDSHAKE

```
register_post_type('portfolio', [
    'public'       => true,
    'has_archive'  => true,
    'show_in_rest' => true,    ← critical
    // ...
]);
```

Without it:

- ❌ Block Editor falls back to classic editor
- ❌ Query Loop block dropdown does NOT show your CPT
- ❌ Site Editor can’t create CPT templates

With it: Everything just works.

WordCamp Europe
Kraków 2026

YOU DON'T WRITE BLOCK MARKUP. YOU CONFIGURE IT.

1. Create new pattern HERO with Create Block Theme plugin
2. Open it, build visually in Site Editor
3. Save → "Save Changes to Theme"
4. Plugin writes block markup back to hero.php

Site Editor **IS** the IDE for blocks. Code editors only for: header metadata, lock attributes, namespaced class additions.

WordCamp Europe
Kraków 2026

WHEN `theme.json` IS NOT ENOUGH

```
Need to style something?
     │
     ├─ Closed elements set (button, link, h1–h6)?
     │    → theme.json styles.elements
     │
     ├─ Core block (paragraph, code, group)?
     │    → theme.json styles.blocks
     │
     └─ Anything else (code, mark, kbd, custom selectors)?
          → style.css with var(--wp--preset--*) tokens
```

`style.css` is **not auto-enqueued** in block themes. Manual `wp_enqueue_style` in `functions.php` is required. One of the few classic-era ceremonies that survives.

WordCamp Europe
Kraków 2026

QUERY LOOP IS MADE VISUAL

WP_Query

Classic PHP era:

```
$q = new WP_Query([
    'post_type'      => 'portfolio',
    'posts_per_page' => 5,
    'orderby'        => 'date',
    'order'          => 'DESC',
]);
while ($q->have_posts()) :
$q->the_post();

get_template_part('parts/project-card')
;
endwhile;
```

FSE era:

```
<!-- wp:query {"query":{
  "postType":"portfolio",
  "perPage":5,
  "orderBy":"date",
  "order":"desc"
}} -->
  <!-- wp:post-template -->
    <!-- wp:pattern
{"slug":"portfolio-fse/project-card"} /-->
  <!-- /wp:post-template -->
<!-- /wp:query -->
```

Same query. Same loop. Different syntax. **Configurable in UI.**

WordCamp Europe
Kraków 2026

CONTEXT FLOWS FROM POST TEMPLATE, NOT QUERY LOOP

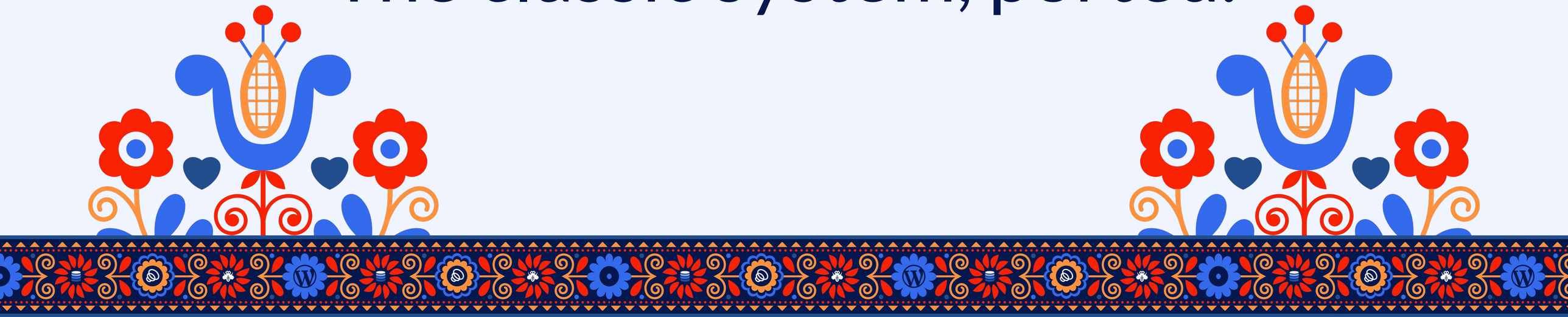
```
Query Loop                          ← runs WP_Query
  └ Post Template                   ← iterates results
      └ ✓ post-title                 ← knows current post
      └ ✓ post-date                  ← knows current post
      └ ✓ post-excerpt               ← knows current post

Query Loop
  └ ✗ post-title                    ← OUTSIDE Post Template
                                       → shows sample data
  └ Post Template
```

`core/post-*` blocks must be **descendants of Post Template**. Siblings of Post Template do not get loop context.

WordCamp Europe
Kraków 2026

04_portfollo-archive

Templates & Hierarchy.
The classic system, ported.



WordCamp Europe
Kraków 2026

TEMPLATE HIERARCHY DIDN'T DIE. IT MOVED.

Classic PHP (root folder)	FSE (templates/ folder)
front-page.php	front-page.html
home.php	home.html
single.php	single.html
single-{post_type}.php	single-{post_type}.html
archive.php	archive.html
archive-{post_type}.php	archive-{post_type}.html
category.php	category.html
tag.php	tag.html
search.php	search.html
404.php	404.html
index.php	index.html

Same rules. New extension.
New folder.

WordCamp Europe
Kraków 2026

HOW WORDPRESS FINDS THE RIGHT TEMPLATE

Single project URL:

```
URL: /projects/wapuu-blocks/

WordPress checks (in order):

   single-portfolio.html        ← match. STOP.
   single-{post-slug}.html
   single.html
   singular.html
   index.html                   ← terminal fallback
```

Archive URL:

```
URL: /projects/

WordPress checks (in order):

   archive-portfolio.html       ← match. STOP.
   archive.html
   index.html                   ← terminal fallback
```

First match wins. `index.html` is the safety net.

WordCamp Europe
Kraków 2026

SINGLE TEMPLATE = NO QUERY LOOP

single.php (classic)

```php
<?php
while ( have_posts() ) :
    the_post();
    the_title();
    the_content();
    the_post_thumbnail();
endwhile;
?>
```

single.html (FSE)

```
<!-- wp:post-title /-->
<!-- wp:post-content /-->
<!-- wp:post-featured-image /-->
```

Post is implicit from URL.

No Query Loop. No Post Template wrapper.

`core/post-*` blocks work directly at template level.

WordCamp Europe
Kraków 2026

05_governance

Locking APIs.
Where FSE earns "**serious developer** " status.

WordCamp Europe
Kraków 2026

WHERE WE ARE

✓ "theme.json feels like XML soup"

→ answered in 02_styles

✓ "I lose precision"

→ answered in 02_styles

✓ "Where is my template hierarchy?"

→ answered in 04_portfolio-archive

→ **"My clients will break it"**

→ **we solve this now.**

WordCamp Europe
Kraków 2026

FOUR LOCK CONTROLS. THREE JOBS.

Control	Where it goes	What it does
`templateLock: "all"`	Template root or Group block	Editor cannot add or remove children
`templateLock: "insert"`	Template or Group block	Editor cannot add new blocks (can remove existing)
`lock: {"move": true, "remove": true}`	Per block	One block can't be moved or deleted
`templateLock: "contentOnly"`	Group block	Layout frozen. Text and media editable.

All four are JSON attributes. Zero PHP. Zero plugins.

WordCamp Europe
Kraków 2026

LAYER 1: LOCK THE STRUCTURE

```
templates/archive-portfolio.html

<!-- wp:group {"templateLock":"all"} -->

    <!-- wp:pattern {"slug":"portfolio-fse/archive-work-list"} /-->
    <!-- wp:pattern {"slug":"portfolio-fse/cta-banner"} /-->

<!-- /wp:group -->
```

✓ Editor can edit pattern content inside

✗ Editor cannot add new blocks at this level

✗ Editor cannot delete the patterns

✓ Page structure preserved

WordCamp Europe
Kraków 2026

LAYER 2: LOCK INDIVIDUAL BLOCKS.

```
patterns/hero.php

<!-- wp:group {
    "lock": {"move": true, "remove": true},
    "className": "hero-outer"
} -->

    <!-- hero content stays here -->

<!-- /wp:group -->
```

✓ Block stays in place

✗ Cannot be moved

✗ Cannot be deleted

✓ Other blocks around it remain editable

WordCamp Europe
Kraków 2026

LAYER 3: CONTENTONLY — THE KILLER FEATURE

Editor **SEES** the layout.

Can edit: text · images · links · post content

Can't edit: structure · layout · spacing · colors

Designer locks the design.

Editor owns the content.

Finally done right.

```
patterns/project-card.php

<!-- wp:group {"templateLock":"contentOnly"} -->

    <!-- wp:post-title /-->          ← editable text
    <!-- wp:post-excerpt /-->        ← editable text
    <!-- wp:post-featured-image /--> ← swappable image
    <!-- wp:post-terms /-->          ← editable taxonomy

<!-- /wp:group -->
```

WordCamp Europe
Kraków 2026

WHERE TO GO FROM HERE

Repository: github.com/dvranic-code/portfolio-fse

Official documentation:

- developer.wordpress.org/themes/
- developer.wordpress.org/themes/templates/template-hierarchy/
- developer.wordpress.org/themes/global-settings-and-styles/

Tools: Create Block Theme plugin

WordCamp Europe
Kraków 2026

THANK YOU!

studioagnis.com

@archiplace2015 on Make WP Slack

www.linkedin.com/in/dejan-rudic-vranic-nis/