

# Launchpad:

---

**TRANSFORMEZ LE DÉVELOPPEMENT DE PLUGIN  
WORDPRESS**

COQUARD Cyrille

27 September, 2024

```
class LimitLoginSubscriber implements DispatcherAwareInterface, PrefixAwareIn
{
    use DispatcherAwareTrait, PrefixAware;

    /**
     * @var State
     */
    protected $state;

    /**
     * @var TransientsInterface
     */
    7 usages
    protected $transients;

    /**
     * @param State $state
     */
    COQUARD Cyrille +1
    public function __construct(State $state, TransientsInterface $transients)
    {
        $this->state = $state;
        $this->transients = $transients;
    }

    /**
     * @hook wp_authenticate
     */
    1 usage   COQUARD Cyrille +1
    public function feed_state_empty_login($user, $password)
    {
        $this->state->set_empty_credentials(empty($user) && empty($password))
    }
}
```

# Qui utilise:

- Composer?

```
class LimitLoginSubscriber implements DispatcherAwareInterface, PrefixAwareIn
{
    use DispatcherAwareTrait, PrefixAware;

    /**
     * @var State
     */
    protected $state;

    /**
     * @var TransientsInterface
     */
    7 usages
    protected $transients;

    /**
     * @param State $state
     */
    COQUARD Cyrille +1
    public function __construct(State $state, TransientsInterface $transients)
    {
        $this->state = $state;
        $this->transients = $transients;
    }

    /**
     * @hook wp_authenticate
     */
    1 usage   COQUARD Cyrille +1
    public function feed_state_empty_login($user, $password)
    {
        $this->state->set_empty_credentials(empty($user) && empty($password))
    }
}
```

# Qui utilise:

- Composer?
- Namespace?

```

class LimitLoginSubscriber implements DispatcherAwareInterface, PrefixAwareIn
{
    use DispatcherAwareTrait, PrefixAware;

    /**
     * @var State
     */
    protected $state;

    /**
     * @var TransientsInterface
     */
    7 usages
    protected $transients;

    /**
     * @param State $state
     */
    ⚠ COQUARD Cyrille +1
    public function __construct(State $state, TransientsInterface $transients)
    {
        $this->state = $state;
        $this->transients = $transients;
    }

    /**
     * @hook wp_authenticate
     */
    1 usage ⚠ cyrille +1
    public function feed_state_empty_login($user, $password)
    {
        $this->state->set_empty_credentials(empty($user) && empty($password))
    }
}

```

## Qui utilise:

- Composer?
- Namespace?
- Injection de dépendences?

```

===== */
SOMMAIRE */
===== */

1= Chargement des fonctions pour le backoffice
2= Chargement des fonctions pour ACF
3= Chargement des fonctions pour Polylang
4= Chargement des fonctions pour le frontoffice
5= Chargement des custom post type
6= Supprimer les taxonomies par défaut
7= Chargement des menus dynamiques
8= Chargement du favicon
9= Configuration des images
10= Supprimer commentaire
11= Get category slug using category ID
12= Get 3 last post
13= initialiser les champs ACF pour Gutenberg

*/

===== */
1= CHARGEMENT DES FONCTIONS POUR LE BACKOFFICE */
===== */

( is_admin() ) {
get_template_part ( 'inc/functions/functions', 'backoffice' );

===== */
2= CHARGEMENT DES FONCTIONS POUR ACF */
===== */

get_template_part ( 'inc/functions/functions', 'acf' );

===== */
4= CHARGEMENT DES FONCTIONS POUR LE FRONTOFFICE */
===== */

( !is_admin() ) {
get_template_part ( 'inc/functions/functions', 'frontoffice' );

===== */
5= CHARGEMENT DES CUSTOM POST TYPE */
===== */

```

# Pourquoi pas comme ça dans les plugins ?

# PROBLÈMES

---



**Complexe**

**Investir**

**Morcelé**

## **Morcelé**

Pas de documentation centralisée

## **Investir**

Dilemme investissement initial

## **Complexe**

Besoin de tout connaître

Le PHP moderne est réservé à une élite.

# LAUNCHPAD

---

Baisser la barre pour rendre le développement moderne accessible.

## Méthodes :

- **Abstraire le maximum de notions**
- **Offrir une base**
- **Documenter les notions**



# COMPLEXE

Abstraire le maximum de notions:

- Subscriber
- Service provider

```
<?php

namespace NoobPress\Security\Debug;

use NoobPress\Dependencies\LaunchpadCore\Container\AbstractServiceProvider;

new *
class ServiceProvider extends AbstractServiceProvider
{

    new *
    public function get_common_subscribers(): array
    {
        return [
            Subscriber::class,
        ];
    }

    /**
     * @inheritDoc
     */
    new *
    protected function define()
    {
        $this->register_service(Subscriber::class);
    }
}
```



# MORCELÉ

## Documenter les notions:

- Documentation framework
- Bonnes pratiques
- Notions sous le capot

- Notions
- Framework concepts
  - Inversion of control
  - Subscribers
  - Dispatcher
- Good practices
  - Hooks
    - Preventing magic constants
    - Decouple features
    - Sanitize filters output
- Testing
  - Organize tests
- CLI
  - Commands
    - Creating a command
- TESTING
  - Unit test
  - Fixtures
  - Integration test
- CONTAINER
  - Architecture
  - Parameters
  - Providers
    - Auto wiring
    - Manual wiring
  - Activation/Deactivation
  - Inflectors
- MODULES

modification is added to that file.

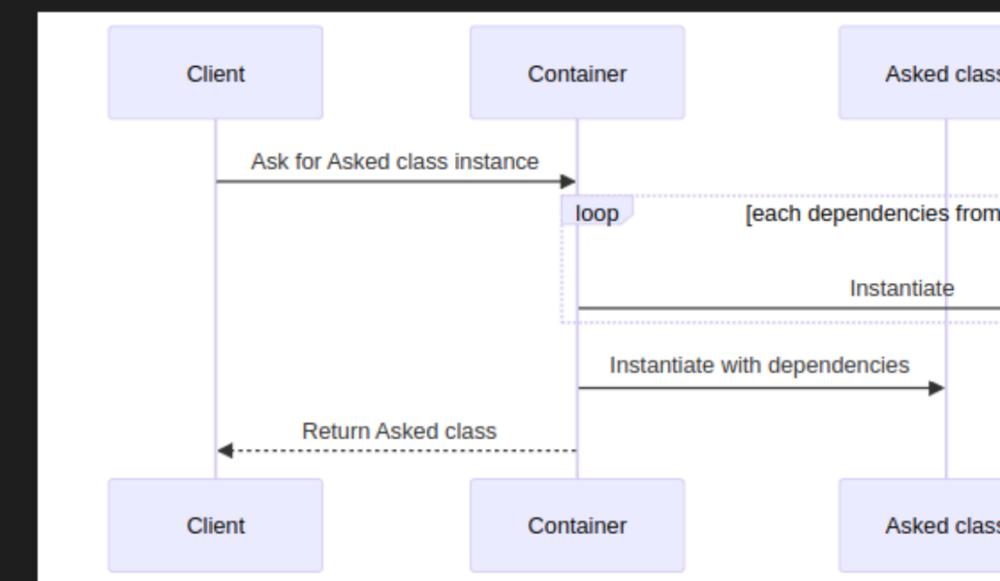
The second solution is to add classes wiring inside the constructor for t

This prevents conflicts as now classes organization is now split between  
main drawback is that now it is really hard to mock classes as it is not a  
classes.

Inversion of dependencies enters here. This solution provides us both b  
drawbacks.

Solution	Can test	Not conflict prone
Wiring inside the core	Yes	No
Wiring inside constructors	No	Yes
Inversion of dependencies	Yes	Yes

The inversion of dependencies is based on wiring inside the core but in  
rely on a container to make the wiring between classes.



This way the wiring logic is not anymore done by the core from the soft  
To tackle this issue two solution can be picked with each one their draw

Rely on the reflection to make the wiring which makes the wiring disapp  
slow and memory intensive.

Create another layer of abstraction to break down the wiring per feature

# UNE DISPARITÉ

---



## Débutant

Valorise:

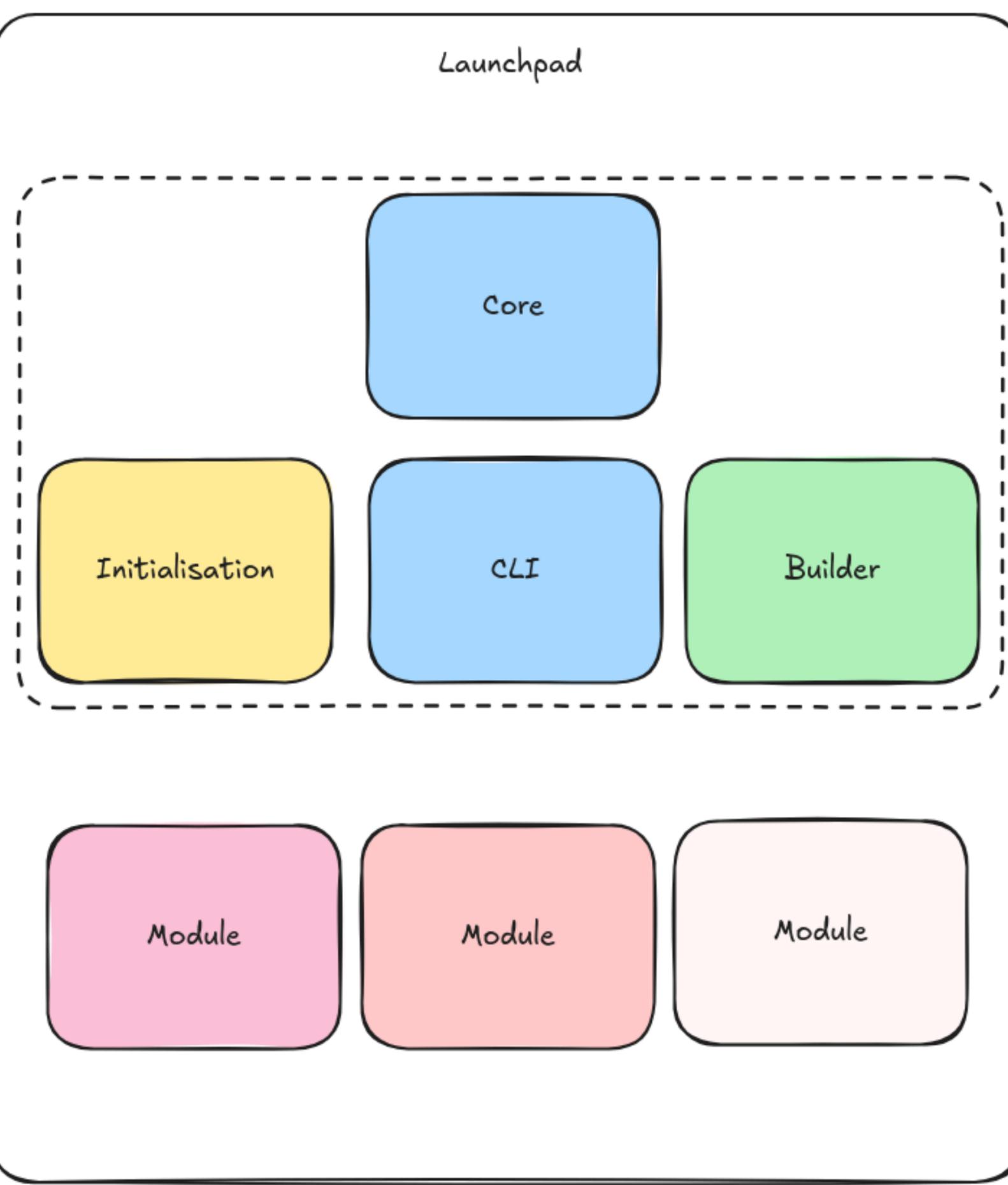
- Simplicité
- Rapidité



## Expert

Valorise:

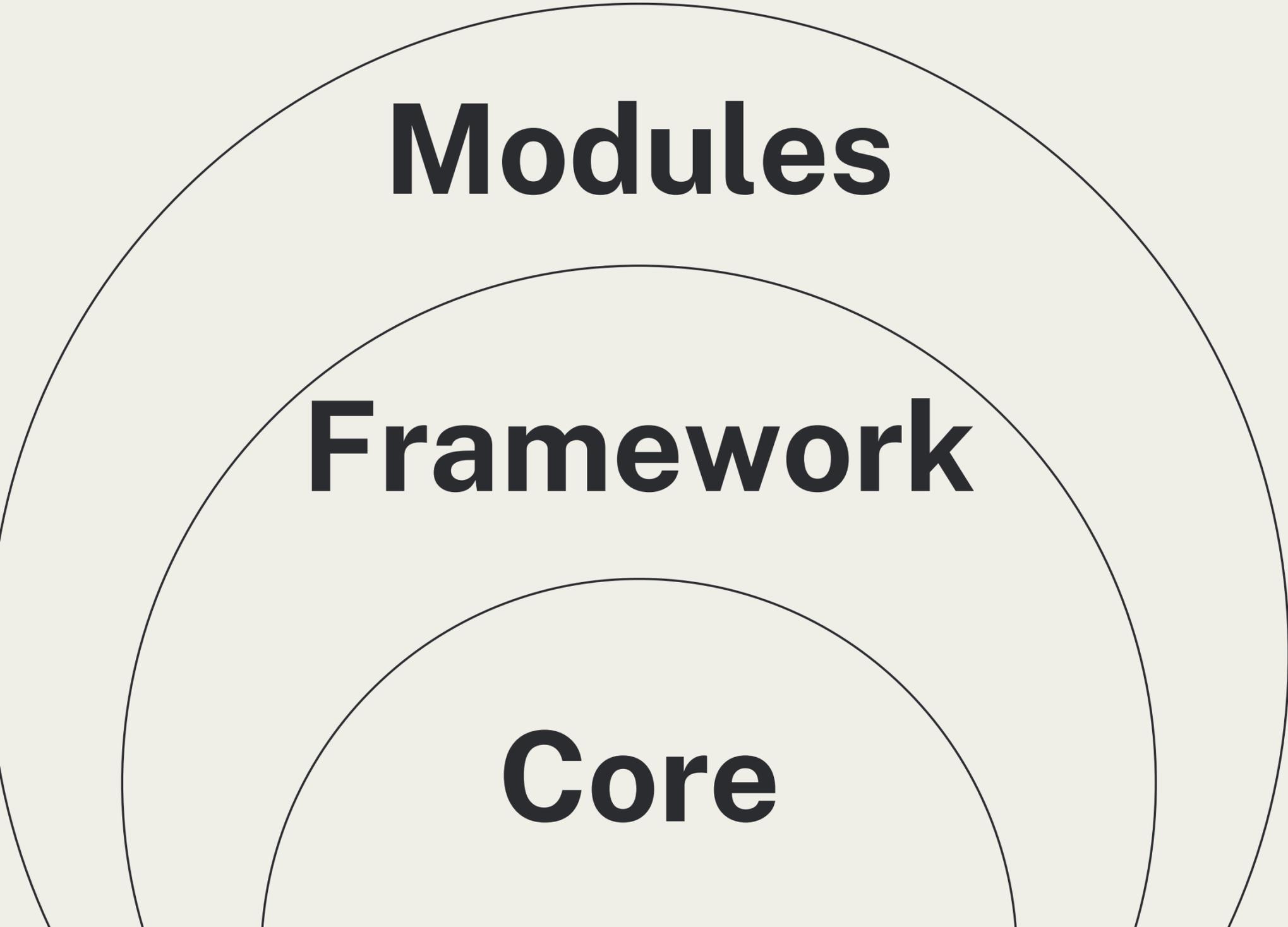
- Liberté



Une architecture  
modulaire

# SOLUTION

---



**Modules**

**Framework**

**Core**

## **Core**

Le strict minimum

## **Framework**

Simplification:

- Base
- Automatisation

## **Modules**

Une constellation de modules:

- Libertée de choix
- Adaptation niveau
- Adaptation besoins

# PROBLÈMES DE DX

---

## *Developer eXperience*

- Equivalent of UX mais pour le développeur
- Il y a toujours une solution lisible
- Réduire la charge mentale

---

*Annotations*

*CLI*

# PROBLÈME

Enregistrer hook:

- add\_action
- add\_filter



```
public static function get_subscribed_events(): array {
    $slug = rocket_get_constant( constant_name: 'WP_ROCKET_SLUG', default: 'wp_rocket_settings' );

    return [
        'update_option' . $slug => [
            [ 'clean_used_css_and_cache', 9, 2 ],
            [ 'maybe_set_processing_transient', 50, 2 ],
            [ 'maybe_unlock_preload', 9, 2 ],
            [ 'maybe_delete_transient', 10, 2 ],
        ],
        'switch_theme' => 'truncate_used_css',
        'permalink_structure_changed' => 'truncate_used_css',
        'rocket_domain_options_changed' => 'truncate_used_css',
        'wp_trash_post' => 'delete_used_css_on_update_or_delete',
        'delete_post' => 'delete_used_css_on_update_or_delete',
        'clean_post_cache' => 'delete_used_css_on_update_or_delete',
        'wp_update_comment_count' => 'delete_used_css_on_update_or_delete',
        'edit_term' => 'delete_term_used_css',
        'pre_delete_term' => 'delete_term_used_css',
        'admin_notices' => [
            [ 'display_no_table_notice' ],
            [ 'notice_write_permissions' ],
        ],
        'rocket_before_add_field_to_settings' => [
            [ 'set_optimize_css_delivery_value', 10, 1 ],
            [ 'set_optimize_css_delivery_method_value', 10, 1 ],
        ],
        'wp_rocket_upgrade' => [
            [ 'set_option_on_update', 14, 2 ],
            [ 'update_safelist_items', 15, 2 ],
            [ 'delete_used_css', 16, 2 ],
            [ 'cancel_pending_jobs_as', 16, 2 ],
            [ 'drop_resources_table', 18, 2 ],
        ],
        'wp_ajax_rocket_spawn_cron' => 'spawn_cron',
        'rocket_deactivation' => 'cancel_queues',
        'admin_head-tools_page_action-scheduler' => 'delete_as_tables_transient_on_tools_pag',
        'pre_get_rocket_option_remove_unused_css' => 'disable_russ_on_wrong_license',
        'rocket_before rollback' => 'cancel_queues'
```

# Une logique compliquée

```
use ...

/*
|-----
| API Routes
|-----
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "api" middleware group. Make something great!
|
*/

Route::middleware('auth:sanctum')->get('uri: '/user', function (Request $request) {
    return $request->user();
});

Route::get('uri: '/{product}/latest', [\App\Http\Controllers\CheckLastVersion::class, 'check'])
Route::get('uri: '/{product}/changelog', [\App\Http\Controllers\ListChangeLog::class, 'list'])
Route::post('uri: '/{product}/upload', [\App\Http\Controllers\UploadVersion::class, 'upload'])
Route::get('uri: '/{product}/{version}', [\App\Http\Controllers\FetchZip::class, 'fetch'])->mi
Route::post('uri: '/', [\App\Http\Controllers\CreateProduct::class, 'create'])->middleware(mi
Route::post('uri: '/{product}/licence', [\App\Http\Controllers\CreateLicence::class, 'create'])
Route::post('uri: '/{product}/licence/cancel', [\App\Http\Controllers\CancelLicence::class, 'c
```

Regarder ailleurs

```
src/Controller/DefaultController.php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;

class DefaultController extends AbstractController

#[Route(
    '/contact',
    name: 'contact',
    condition: "context.getMethod() in ['GET', 'HEAD'] and request.headers.get('User-Agent') matches '%app.allowed_browser'",
    // expressions can also include config parameters:
    // condition: "request.headers.get('User-Agent') matches '%app.allowed_browser'"
)]
public function contact(): Response
{
    // ...
}

#[Route(
    '/posts/{id}',
    name: 'post_show',
    // expressions can retrieve route parameter values using the "params" variable
    condition: "params['id'] < 1000"
)]
public function showPost(int $id): Response
{
    // ... return a JSON response with the post
}
```

Regarder encore ailleurs

```

/**
 * @param State $state
 */
COQUARD Cyrille +1
public function __construct(State $state, TransientsInterface $transients)
{
    $this->state = $state;
    $this->transients = $transients;
}

/**
 * @hook wp_authenticate
 */
1 usage COQUARD Cyrille +1
public function feed_state_empty_login($user, $password)
{
    $this->state->set_empty_credentials(empty($user) && empty($password));
}

/**
 * @hook $prefixis_empty_credentials
 */
1 usage COQUARD Cyrille
public function share_state_empty_credentials($empty) {
    return $this->state->is_empty_credentials();
}

/**
 * @hook wp_login_failed
 */
1 usage COQUARD Cyrille
public function increase_counter_failures()
{
    $ip = $this->dispatcher->apply_string_filters("${$this->prefix}ip_address", '');
    $max = $this->dispatcher->apply_int_filters("${$this->prefix}counter_max", 5);
    $expiration = $this->dispatcher->apply_int_filters("${$this->prefix}counter_ex");
    $empty = $this->dispatcher->apply_bool_filters("${$this->prefix}is_empty_crede

```

# Utiliser les annotations

```
remove_action( 'template_redirect', 'wp_redirect_admin_locations', 1000 );
}

/**
 * @hook wp_redirect
 * @hook site_url
 */
no usages  👤 COQUARD Cyrille
public function wp_redirect($location, $status)
{
    if( ! $this->is_active() ) {
        return $location;
    }

    if ( strpos( $location, needle: 'https://wordpress.com/wp-login.php' ) !== false )
        return $location;

    if ( strpos( $location, needle: 'wp-login.php?action=postpass' ) !== false ) {
        return $location;
    }

    $admin_slug = $this->dispatcher->apply_string_filters( "{$this->prefix}admin_sl

    if ( strpos( $location, needle: 'wp-login.php' ) !== false && strpos( wp_get_re

        $queries = wp_parse_url($location, PHP_URL_QUERY);

        if($queries) {
            $admin_slug .= "?$queries";
        }

        return get_site_url() . "/"$admin_slug";
    }

    return $location;
}
```

# Plusieurs événements?

# Plusieurs annotations



```
<?php
namespace {{ namespace }};
use {{ base_namespace }}Dependencies\LaunchpadCore\EventManagement\Opti

class {{ class_name }} implements OptimizedSubscriberInterface {

    /**
     * Returns an array of events that this subscriber wants to listen
     *
     * The array key is the event name. The value can be:
     *
     * * The method name
     * * An array with the method name and priority
     * * An array with the method name, priority and number of accepted
     *
     * For instance:
     *
     * * array('hook_name' => 'method_name')
     * * array('hook_name' => array('method_name', $priority))
     * * array('hook_name' => array('method_name', $priority, $accepted
     * * array('hook_name' => array(array('method_name_1', $priority_1
     *
     * @return array
     */
    public static function get_subscribed_events() {
        return [

        ];
    }
}
```

# Une CLI pour générer du code

```
"wp-launchpad/cli-installer": "^1.0", v1.1.0
"wp-launchpad/cli-installer": "^0.0", v0.0.5
"wp-launchpad/core": "^v3.0.0-alpha", v3.0.0-alpha
"wp-launchpad/take-off": "^1.0", v1.0.8
"wp-media/phpunit": "^3.0" v3.0.3
},
"autoload": {
    "psr-4": {
        "Launchpad\\": "inc/"
    }
},
"autoload-dev": {
    "psr-4": {
        "Launchpad\\Tests\\": "tests/"
    }
},
"extra": {
    "installer-paths": {
        "vendor/{$vendor}/{$name}/": [
            "type:wordpress-plugin"
        ]
    },
    "strauss": {
        "target_directory": "vendor-prefixed",
        "namespace_prefix": "Launchpad\\Dependencies\\",
        "classmap_prefix": "Launchpad",
        "constant_prefix": "Launchpad",
        "packages": [
            "league/container",
            "wp-launchpad/core",
            "wp-launchpad/autoresolver"
        ]
    }
}
```

# Une CLI pour abstraire la complexité

# Merci!

---

N'HÉSITÉZ PAS À CONTRIBUER



COQUARD Cyrille  
Slack : @coquardcyr

