

WordPress: Une cause perdue?

COQUARD Cyrille





**WordPress, c'est
de la merde !**

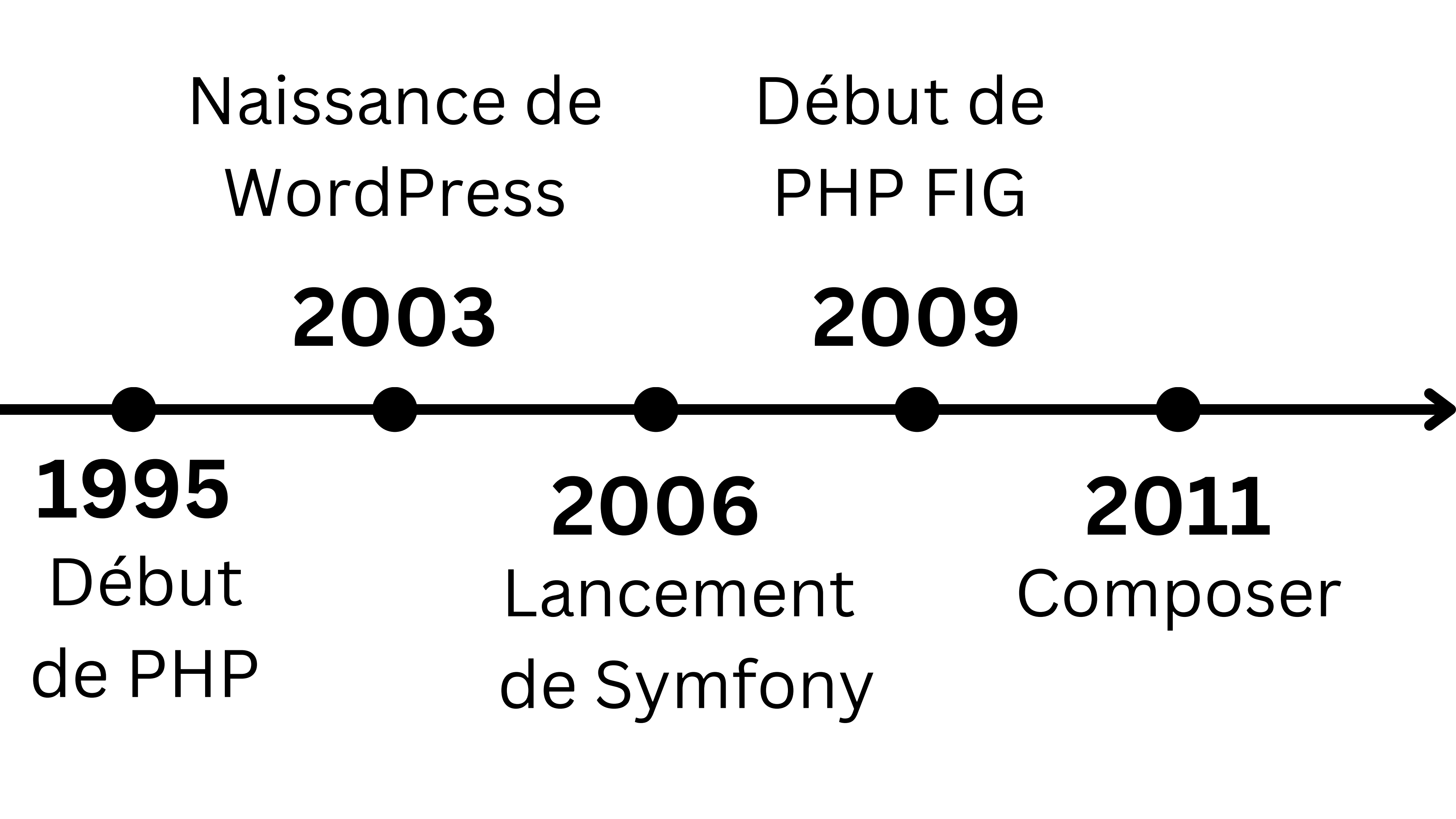


**“PHP, c’est de la
merde !”**

Grafikart



La quintessance de PHP



Naissance de
WordPress

2003

Début de
PHP FIG

2009

1995

Début
de PHP

2006

Lancement
de Symfony

2011

Composer

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

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

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

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

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

```
/* ===== */
/* 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 posts pour Gutenberg
*/
/* ===== */
/* 1= CHARGEMENT DES FONCTIONS POUR LE BACKOFFICE */
/* ===== */
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 */
/* ===== */
if ( !is_admin() ) {
    get_template_part ( 'inc/functions/functions', 'frontoffice' );
}
```

La liberté: épée de danocles

**WordPress,
c'est quoi?**

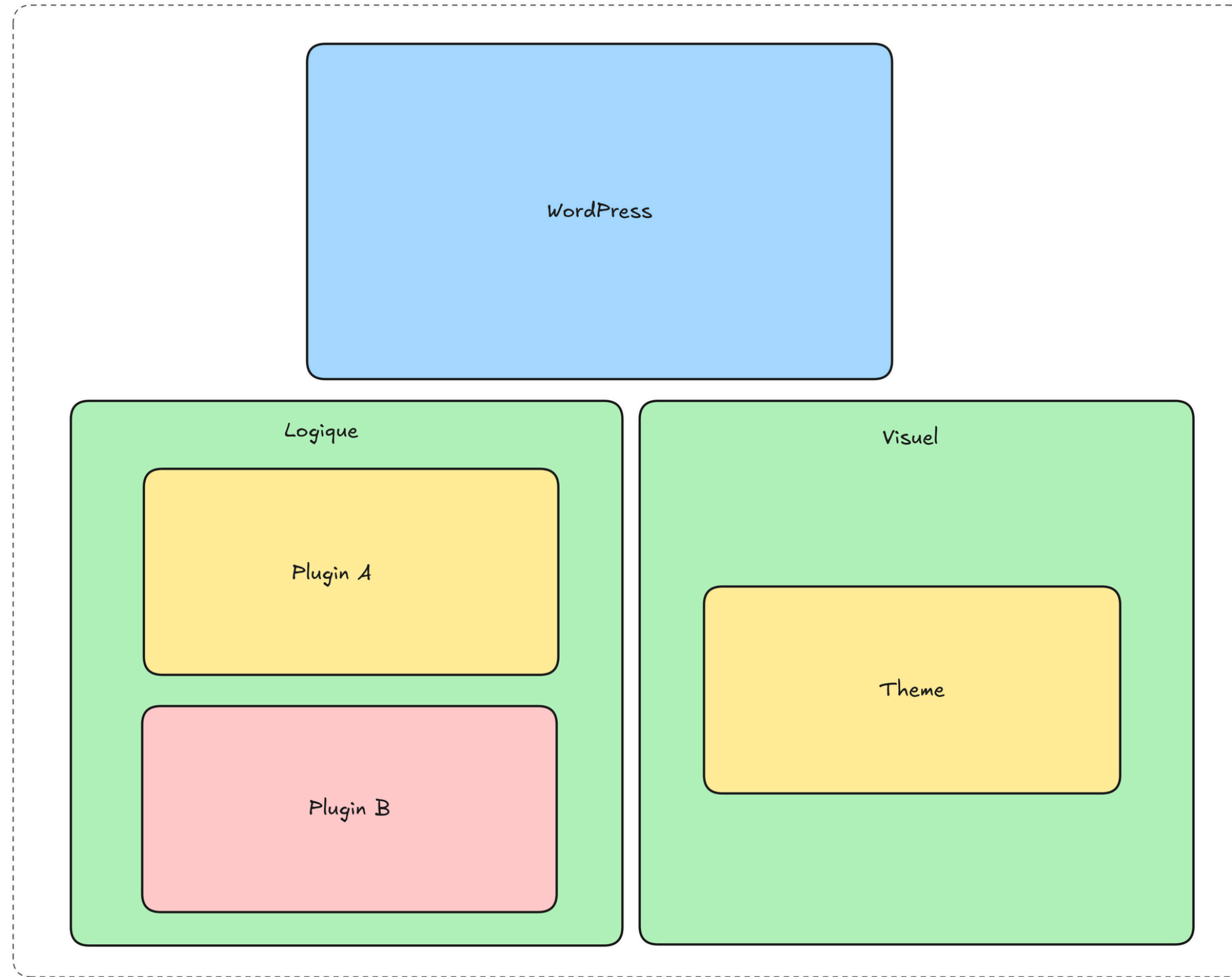
**Et si un site
était une voiture?**

A blue sports car is shown in motion on a road, with a blurred background suggesting speed. The car is viewed from a front-quarter perspective. A semi-transparent blue rectangular box is overlaid on the car's body, containing the word "Symfony" in white, bold, sans-serif font. The car's front grille features a small red and white logo. The road has a white line on the right side, and the background shows a rocky, hilly landscape under bright light.

Symfony

A blue Renault hatchback is parked on a gravel surface in a forest. The car is positioned in the lower half of the frame, facing right. In the background, there are many trees and a large stack of cut logs. A semi-transparent blue rectangular box is overlaid on the car's windshield and roof area, containing the text "WordPress" in white. The license plate is white with a blue border and contains the letter "F" on the left and "28" on the right.

WordPress



L'architecture modulaire



Thème

Gère le visuel



GRAVITY

mollie



UltimateMember

yoast

• A •



UpdraftPlus

Plugin

Gère la logique



TOTAL
CACHE



Wordfence
WORDPRESS SECURITY

WP ROCKET
Superior WordPress Performance



SUCURI



rafflepress

ACF

Une frontière

Créer son premier plugin

```
php mon-plugin.php ×
1  <?php
2
3  /*
4  Plugin Name: Mon plugin
5  Plugin URI: http://URI\_Of\_Page\_Describing\_Plugin\_and\_Updates
6  Description: A brief description of the Plugin.
7  Version: 1.0
8  Author: cyrille
9  Author URI: http://URI\_Of\_The\_Plugin\_Author
10 License: A "Slug" license name e.g. GPL2
11 */
```

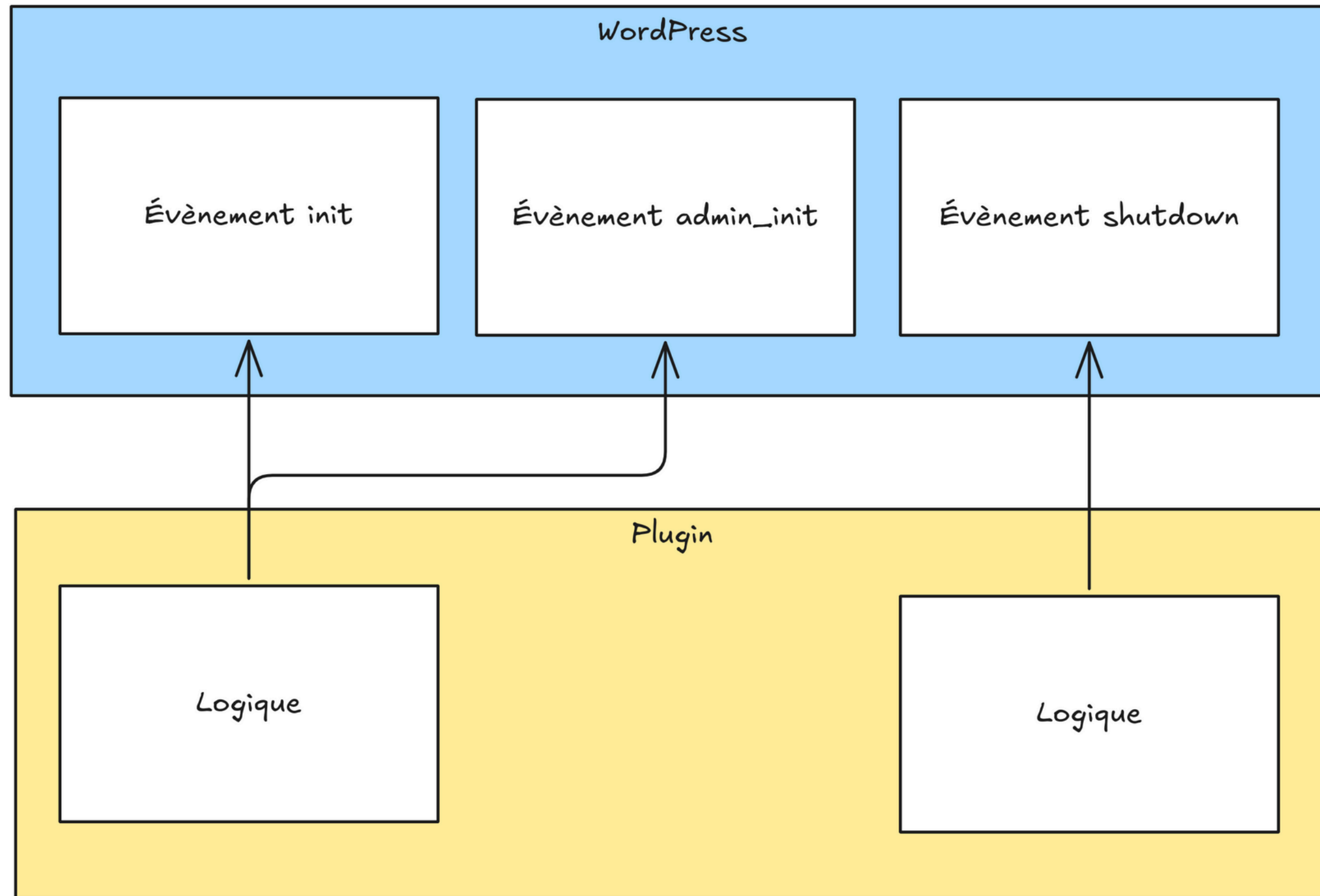
Créer un plugin

The screenshot shows the WordPress 'Plugins' management page. On the left is a dark sidebar with navigation links: Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins (highlighted), Installed Plugins, Add New Plugin, Users, Tools, Settings, and Collapse menu. The main content area is titled 'Plugins' and includes an 'Add New Plugin' button. Below the title, it shows filters: 'All (2) | Active (1) | Inactive (1) | Auto-updates Disabled (2)'. There are two sections of installed plugins, each with a 'Bulk actions' dropdown and an 'Apply' button. The first section contains two plugins: 'Hello Dolly' (Version 1.7.2 by Matt Mullenweg) and 'Mon plugin' (Version 1.0 by cyrille). The second section contains one plugin: 'Plugin'.

<input type="checkbox"/>	Plugin	Description
<input type="checkbox"/>	Hello Dolly Activate Delete	This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Ar Version 1.7.2 By Matt Mullenweg View details
<input type="checkbox"/>	Mon plugin Deactivate	A brief description of the Plugin. Version 1.0 By cyrille Visit plugin site
<input type="checkbox"/>	Plugin	Description

C'est aussi simple

Comment faire du code utile?



Basé sur les évènements

Action

```
no usages
function ma_logique_ajout_regle_redirection() {
    // ma logique a l'initialisation de WordPress
    add_rewrite_rule( '^sitemap\.xml$', 'index.php?sitemap=index', 'top' );
}

add_action('init', 'ma_logique_ajout_regle_redirection');
```

Filtre

```
no usages
function changement_url_sitemap($url, $path) {
    if ( '/wp-sitemap.xml' === $path ) {
        return str_replace( search: '/wp-sitemap.xml', replace: '/sitemap.xml', $url );
    }

    return $url;
}

add_filter('home_url', 'changement_url_sitemap', 10, 2);
```

Lier la logique aux évènements

XML Sitemap

This XML Sitemap is generated by W

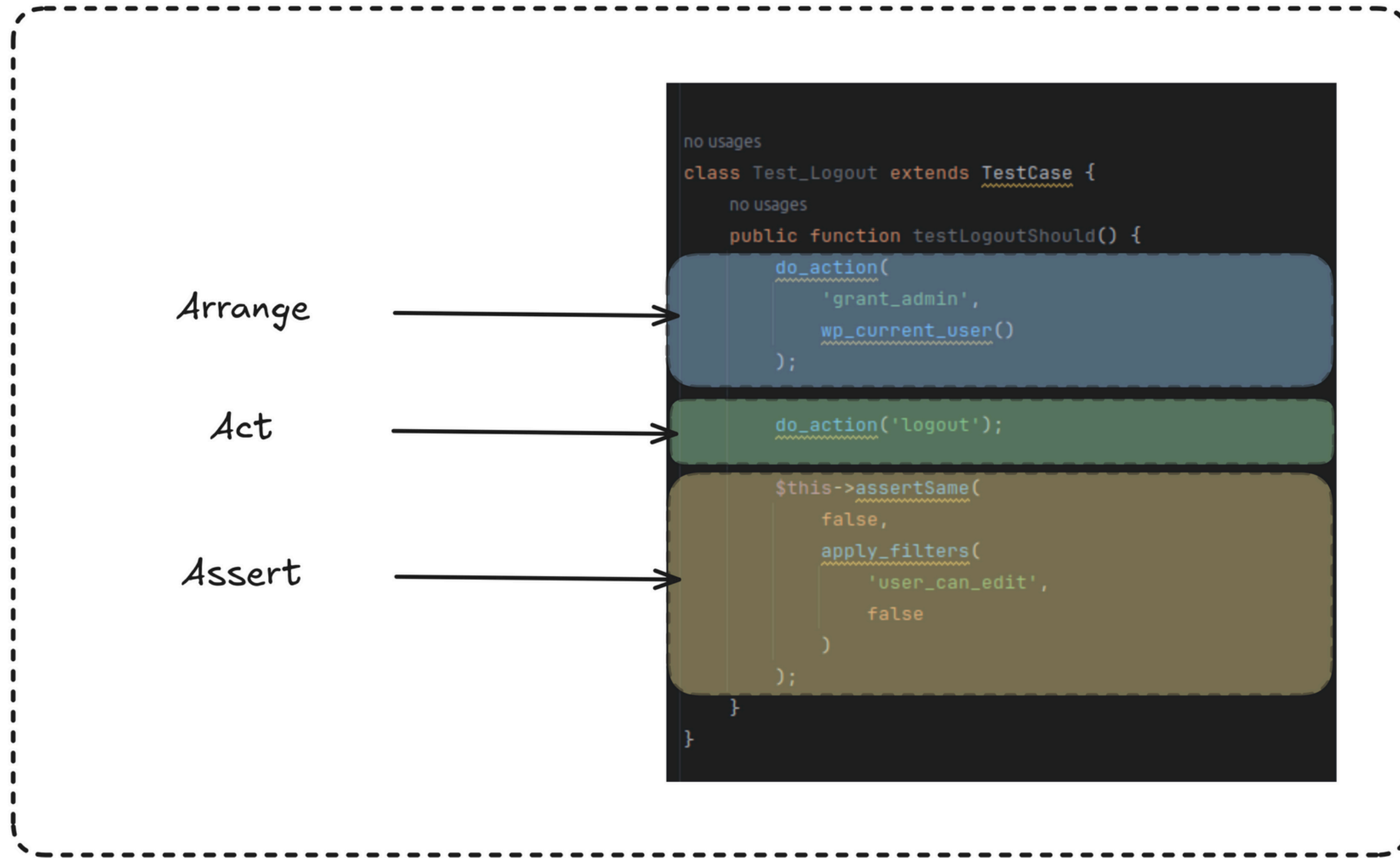
[Learn more about XML sitemaps.](#)

Number of URLs in this XML Sitemap

URL
http://localhost:8888/wp-sitemap-
http://localhost:8888/wp-sitemap-
http://localhost:8888/wp-sitemap-
http://localhost:8888/wp-sitemap-

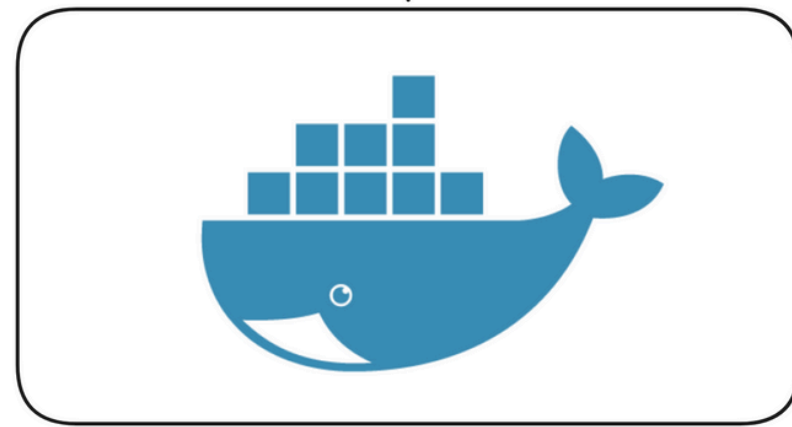
Changer le sitemap

Tester son code



Centré hook

Lancer dans le dossier avec le plugin



Installer WordPress

wp-media/phpunit

```
<?php
/** Initializes the wp-media/phpunit handler, which then calls the rocket integration test suite. ...*/

define( 'WPMEDIA_PHPUNIT_ROOT_DIR', dirname( dirname( path: __DIR__ ) ) . DIRECTORY_SEPARATOR );
define( 'WPMEDIA_PHPUNIT_ROOT_TEST_DIR', __DIR__ );

require_once WPMEDIA_PHPUNIT_ROOT_DIR . 'vendor/wp-media/phpunit/Integration/bootstrap.php';

define( 'WPMEDIA_IS_TESTING', true ); // Used by wp-media/{package}.
```

Mettre en place

```
no usages
public function testOldSitemapShouldRedirect() {

    $url = 'https://example.org/wp-sitemap.xml';

    $path = '/wp-sitemap.xml';

    $url = apply_filters( filter: 'home_url', $url, $path);

    $this->assertSame('https://example.org/sitemap.xml', $url);
}
```

```
no usages
public function testOtherUriShouldReturnSame() {

    $url = 'https://example.org/my-uri';

    $path = '/my-uri';

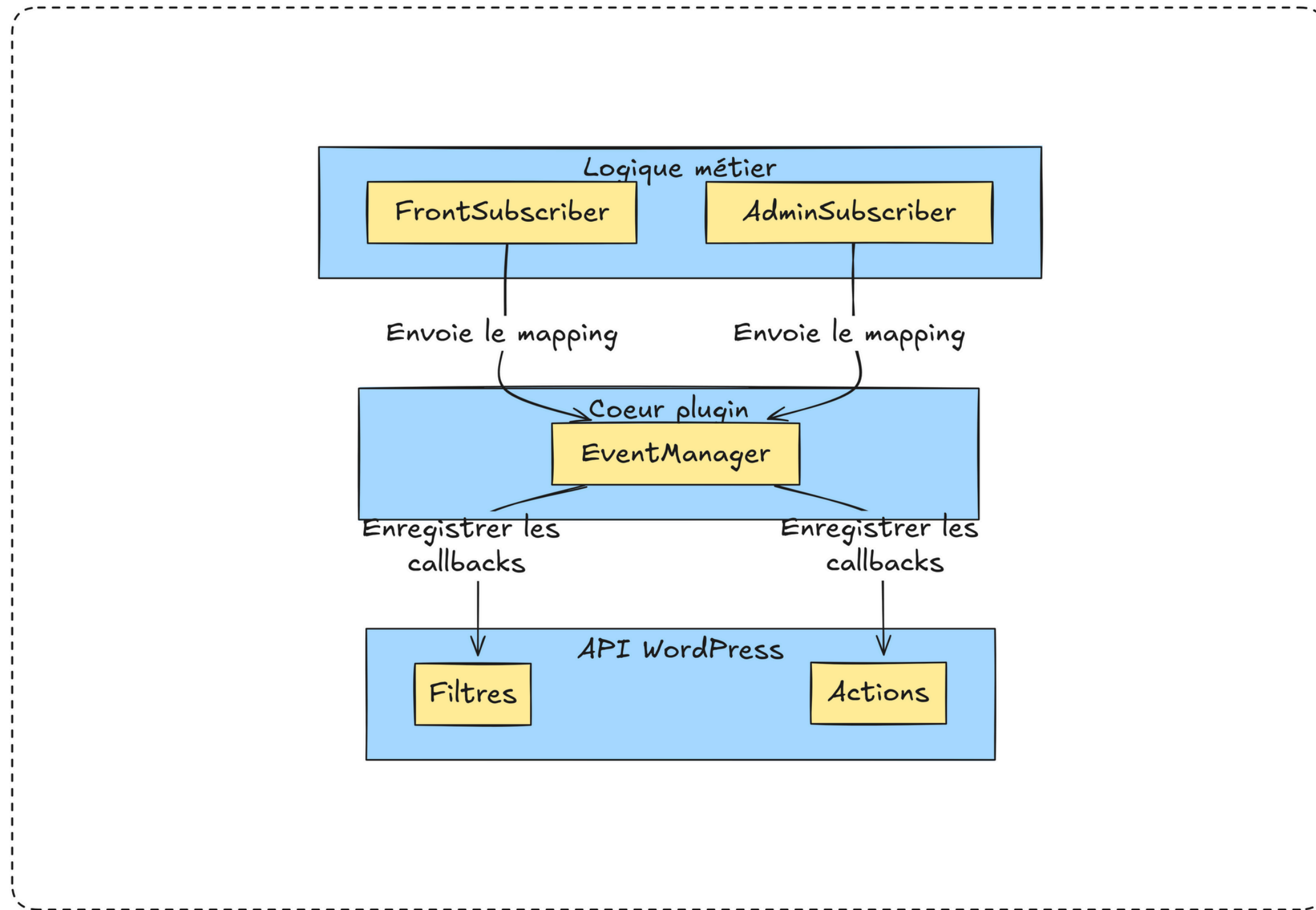
    $url = apply_filters( filter: 'home_url', $url, $path);

    $this->assertSame('https://example.org/my-uri', $url);
}
```

Tests du changement d'URL



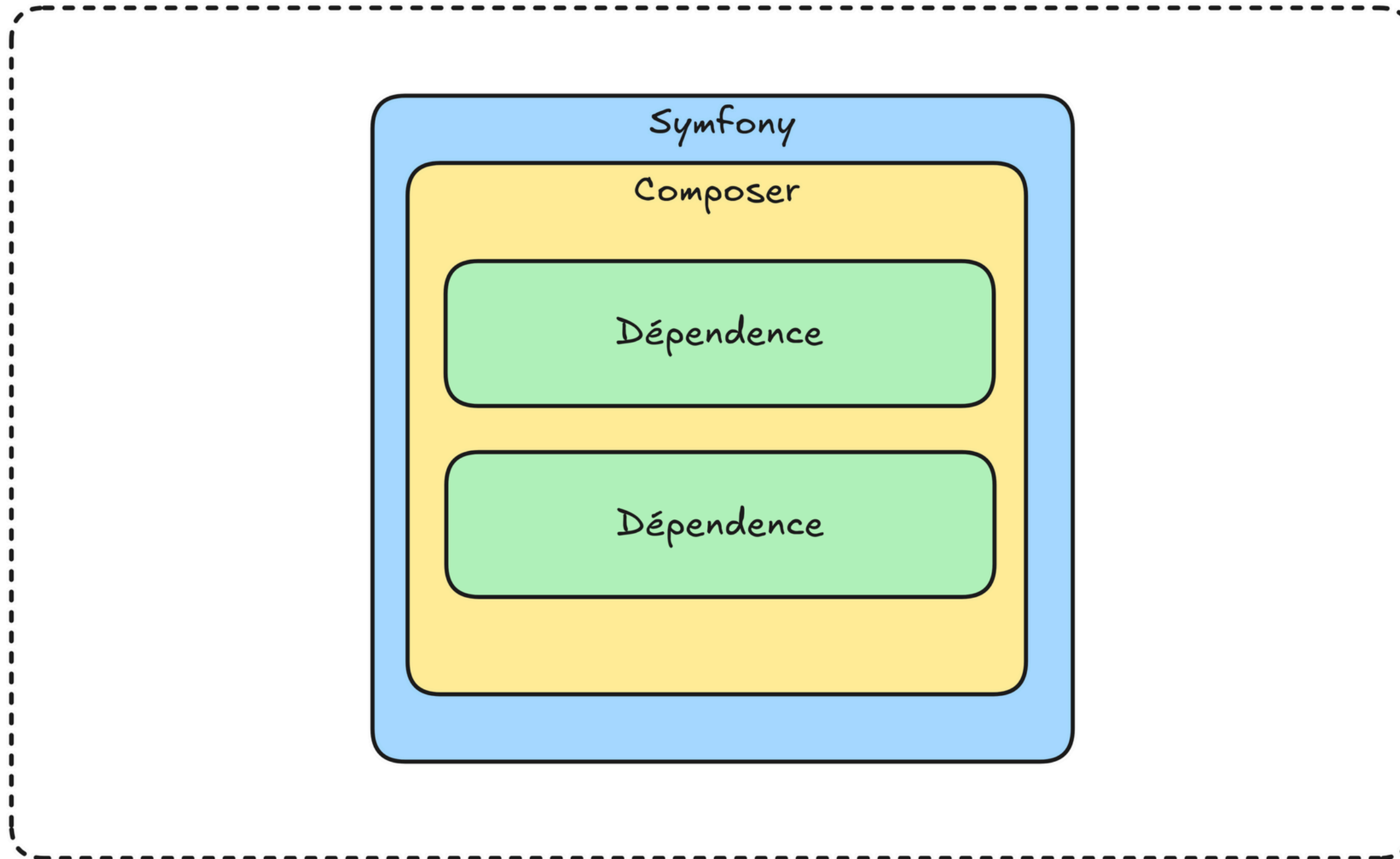
S'organiser dans le plugin



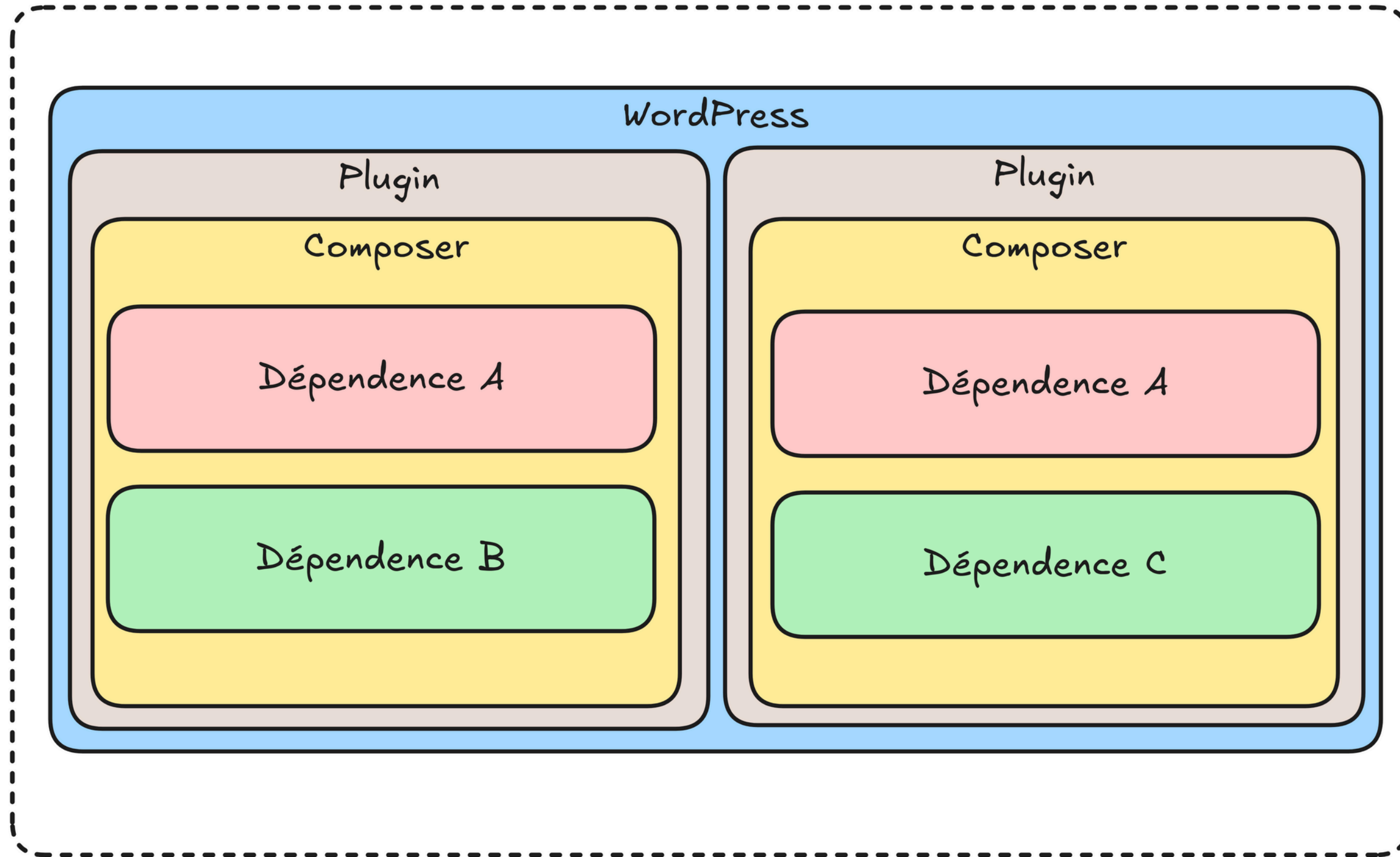
Délier l'API hook WordPress

**Et l'inversion de
dépendance?**

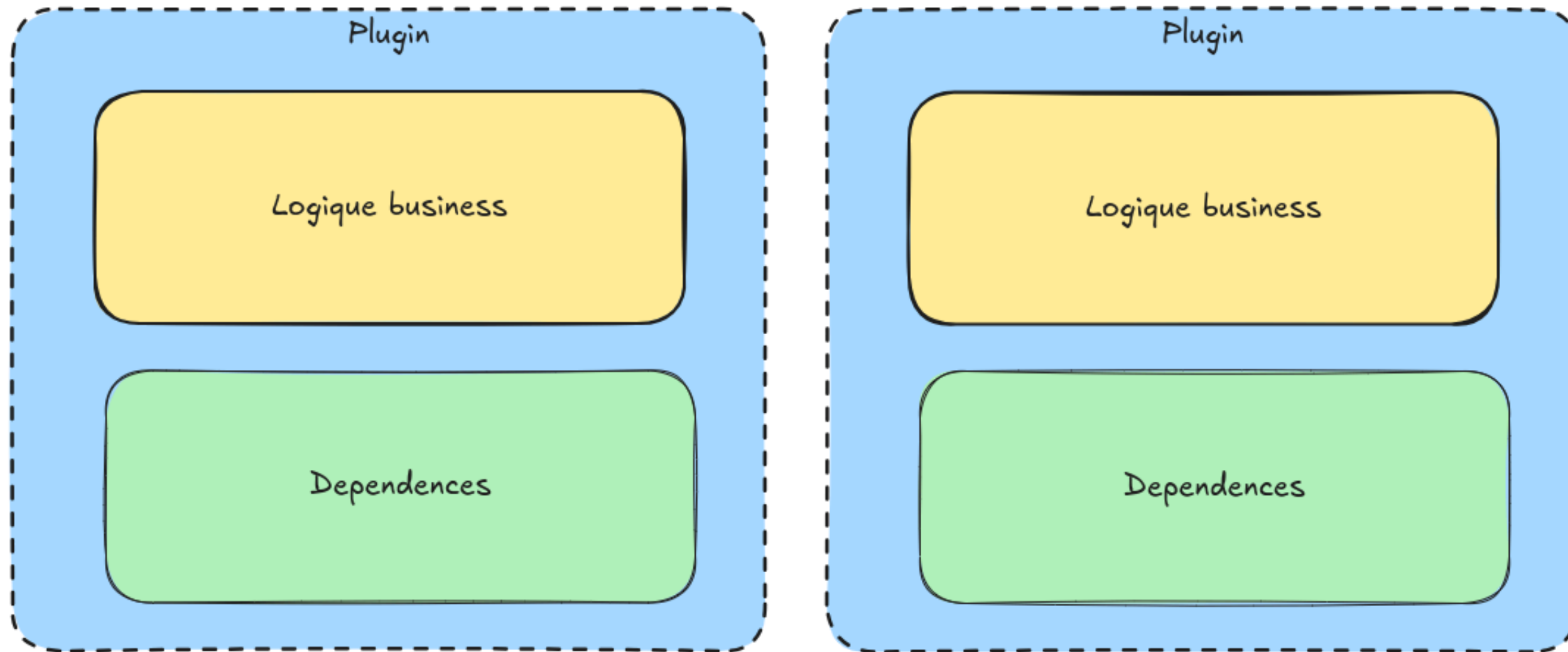
Et Composer dans tout ca?



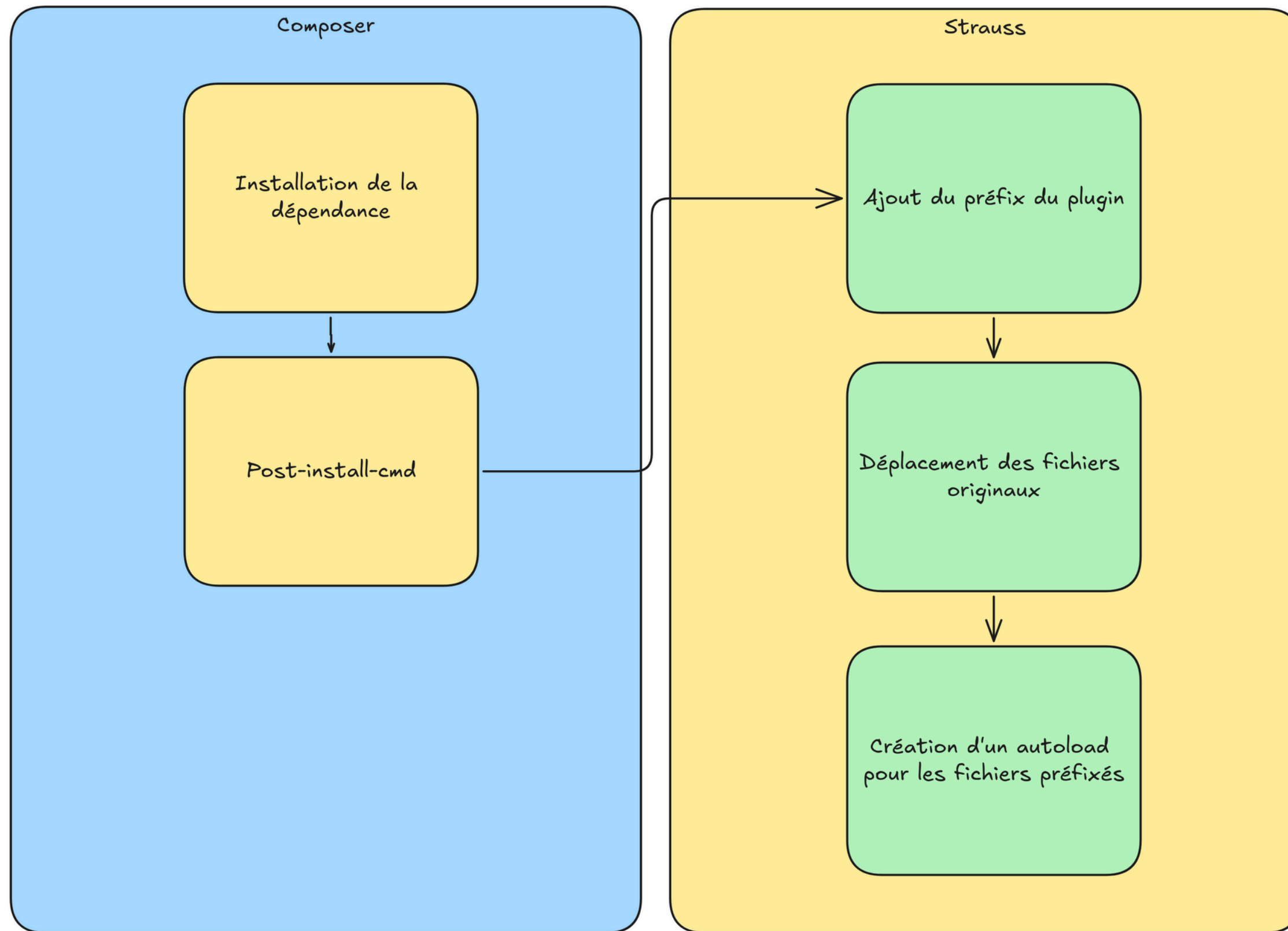
Symfony: un projet



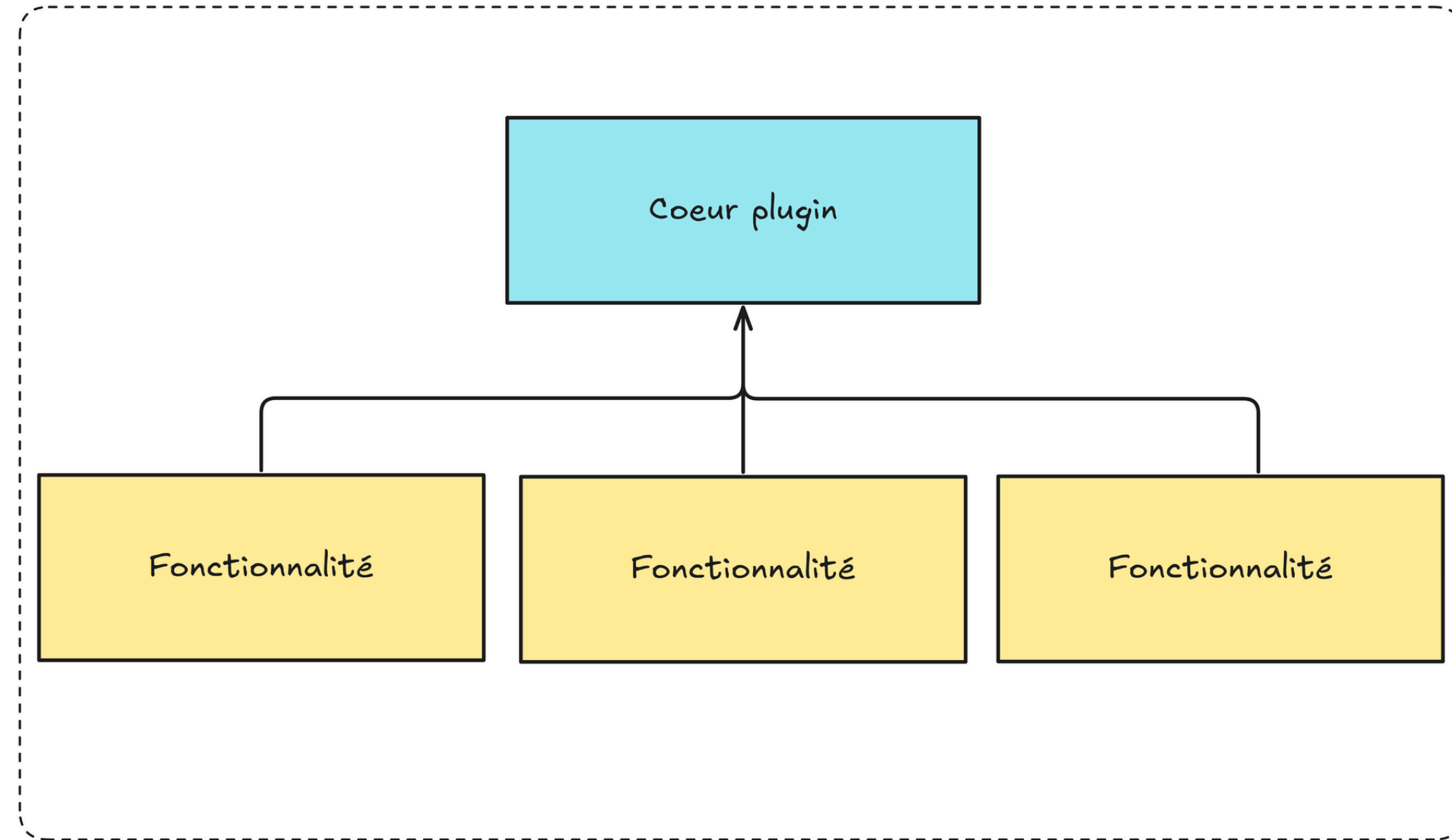
WordPress: une multitude



Utiliser un namespace local



Strauss



Architecture du plugin

Wordpress charge le plugin

Chargement de l'autoload
de la logique métier

Chargement de l'autoload
des dépendances

Initialisation du
container

Ajout de la logique
métier au container

Enregistrement des
subscribers

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

namespace MonPlugin;

if ( file_exists( filename: __DIR__ . 'vendor/autoload.php' ) ) {
    require __DIR__ . 'vendor/autoload.php';
}

if ( file_exists( filename: __DIR__ . 'vendor-prefixed/autoload.php' ) ) {
    require __DIR__ . 'vendor-prefixed/autoload.php';
}

add_action(
    'plugins_loaded',
    function () {

        $container = new Container();

        $providers = [
            \MonPlugin\Feature\ServiceProvider::class,
        ];

        foreach ( $providers as $provider ) {
            $container->addServiceProvider( $provider );
        }

        $event_manager = new EventManager();

        foreach ( $providers as $provider ) {
            $instance = new $provider;
            foreach ( $instance->get_subscribers() as $subscriber ) {
                $event_manager->add_subscriber( $container->get( $subscriber ) );
            }
        }

    }
);
```

Initialisation du plugin

Subscriber

```
namespace MonPlugin;

use MonPlugin\EventManagement\SubscriberInterface;

1 usage
class Subscriber implements SubscriberInterface {

    no usages
    public static function get_subscribed_events() {
        return [
            'init' => 'ajout_regle_sitemap',
            'home_url' => ['changement_url_sitemap', 10, 2],
        ];
    }

    no usages
    public function ajout_regle_sitemap() {
        add_rewrite_rule('^sitemap.xml', 'index.php?sitemap=index', 'top');
    }

    no usages
    public function changement_url_sitemap($url, $path) {
        if('/wp-sitemap.xml' === $path) {
            return str_replace(search: '/wp-sitemap.xml', replace: '/sitemap.xml', $url);
        }

        return $url;
    }
}
```

Provider

```
namespace MonPlugin;

class ServiceProvider extends AbstractServiceProvider
{
    no usages
    public function get_subscribers(): array {
        return [
            Subscriber::class,
        ];
    }
}
```

Retour à notre exemple

Le problème des bonnes pratiques

**Baisser la barre pour rendre les
bonnes pratiques accessible.**



Launchpad

Subscriber

```
COQUARD Cyrille
class Subscriber
{
    /**
     * @hook xmlrpc_enabled
     */
    1 usage COQUARD Cyrille
    public function disable_rpc()
    {
        return false;
    }

    /**
     * @hook wp_headers
     */
    no usages COQUARD Cyrille
    public function remove_pingback( $headers )
    {
        unset( $headers['X-Pingback'] );
        return $headers;
    }
}
```

Provider

```
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);
    }
}
```

Le minimum de notions

Framework

The screenshot shows a sidebar on the left with categories: TUTORIALS, CLI, TESTING, and CONTAINER. The 'Architecture' page is open, showing a 'General' section with text about understanding the project and its purposes, and a 'Subscriber loading' section with text about loading subscribers and to inside the `inc/plugin.php` file.

Notions

The screenshot shows a sidebar on the left with categories: Notions, Framework concepts, Good practices, Hooks, Testing, CLI, and TESTING. The 'Notions' page is open, showing a table with columns 'Solution' and 'Can test'. Below the table is a diagram showing a 'Client' box and a 'Container' box. An arrow labeled 'Ask for Asked class instance' points from the Client to the Container, which has a 'loop' label below it.

Solution	Can test
Wiring inside the core	Yes
Wiring inside constructors	No
Inversion of dependencies	Yes

Documenter les notions

Début

```
no usages
function ma_logique_ajout_regle_redirection() {
    // ma logique a l'initialisation de WordPress
    add_rewrite_rule( '^sitemap\.xml$', 'index.php?sitem
}

add_action('init', 'ma_logique_initilisation');
```

Fin

```
namespace MonPlugin;

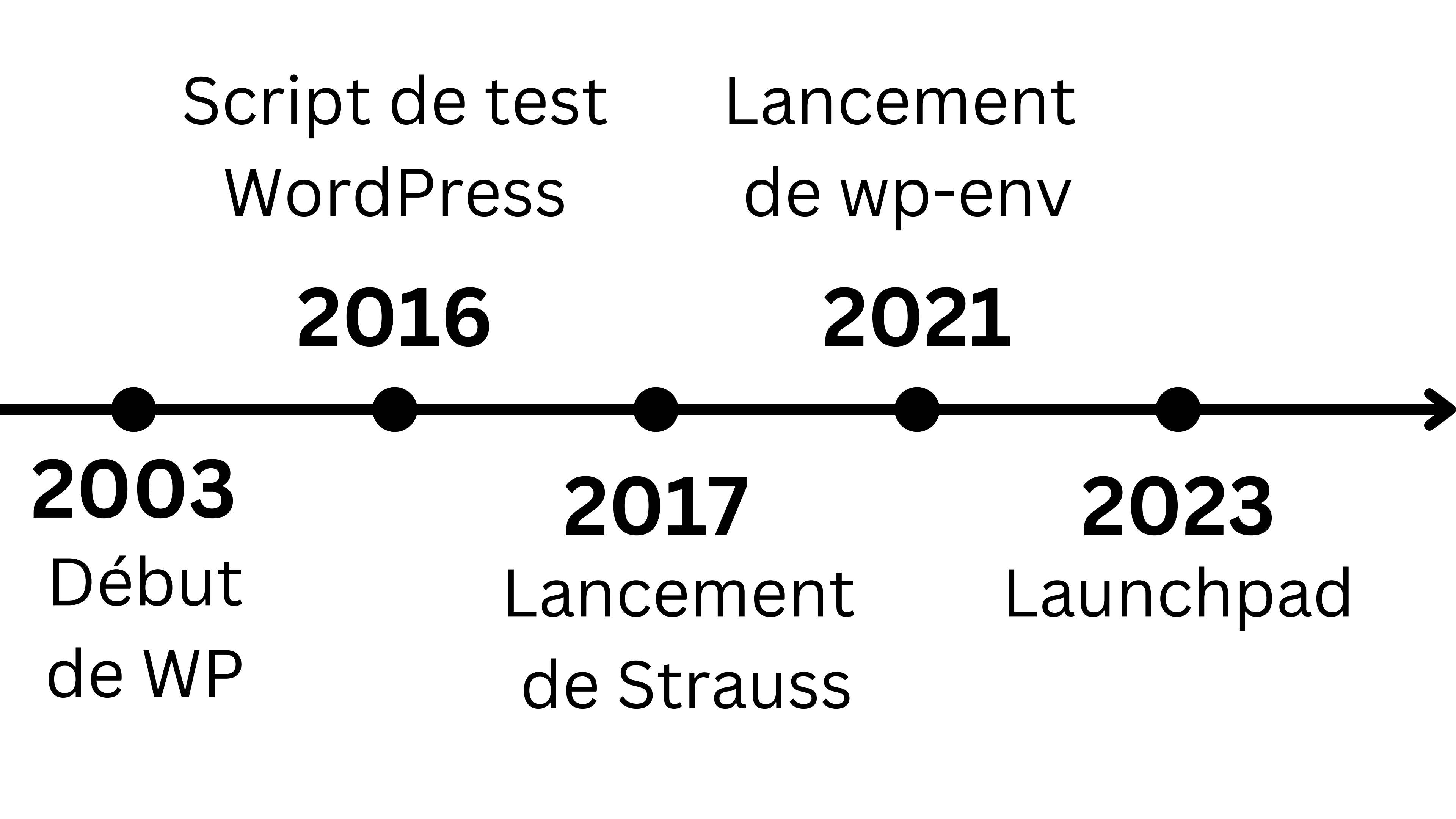
1 usage
class Subscriber {

    /**
     * @hook init
     */
    no usages
    public function ajout_regle_sitemap() {
        add_rewrite_rule('^sitemap\.xml', 'index.php?sitemap=index', 'top');
    }

    /**
     * @hook home_url
     */
    no usages
    public function changement_url_sitemap($url, $path) {
        if('/wp-sitemap.xml' === $path) {
            return str_replace( search: '/wp-sitemap.xml', replace: '/sitemap.xml', $url);
        }

        return $url;
    }
}
```

Évolution de la syntaxe



Script de test
WordPress

Lancement
de wp-env

2016

2021

2003

2017

2023

Début
de WP

Lancement
de Strauss

Launchpad

Merci !



COQUARD Cyrille
Ingénieur logiciel
chez WP Media

