Documentation développeur

PUBLICUM Essentiel

Sommaire

Introduction

Publicum Essentiel est une base complète pour sites WordPress professionnels, éco-conçue et pensée pour l'accessibilité. Il fournit une structure de pages standardisée, des gabarits de contenu prêts à l'emploi et un système de design configurable, afin de limiter l'empreinte technique tout en respectant les principaux critères d'accessibilité courante.

L'éco-conception se traduit par une feuille de style unique, un volume de JavaScript très réduit, l'absence de frameworks externes et la désactivation de fonctionnalités WordPress non nécessaires. Ces choix visent à limiter la quantité de ressources chargées, simplifier le comportement côté navigateur et faciliter la maintenance du site dans la durée.

L'accessibilité est prise en compte dans la structure HTML, l'organisation des pages, la navigation, la hiérarchie typographique et les composants de base. Les couleurs, polices et bordures sont pilotées par un système de variables de design ; les réglages sont accessibles dans l'administration, sans modification du code. Des vérifications restent possibles selon les combinaisons de couleurs retenues.

Publicum Essentiel intègre un module de gestion de site simplifié, qui rassemble les fonctions courantes : création et modification des pages, actualités, réglages de design et opérations de base liées à l'entretien du site. L'objectif est que le client final puisse gérer le contenu et l'apparence du site en autonomie, sur une structure technique stable.

Ce document présente l'architecture interne du thème, l'organisation des fichiers, les composants principaux et les choix techniques liés à la performance, à la sobriété et à l'accessibilité, afin de permettre à un développeur ou à une agence d'évaluer rapidement le fonctionnement de Publicum Essentiel.

0. Métadonnées

0.1 Nom et version

Nom du thème : Publicum Essentiel
Thème enfant basé sur Astra.

Version documentée : 0.1.2025
(Convention de versionnement recommandée : *année.version* — exemple : `2025.1`, `2025.2`)

0.2 Auteur

Développement :
La Petite Fabrique Digitale — Sophie Boutemy
Rennes, France.

0.3 Licence et droits d’utilisation

Code du thème : GPL v2
Structure, organisation, nomenclature ACF et documentation : CC BY-NC-ND 4.0
La redistribution, la revente et l’usage commercial hors cadre de la licence sont interdits sans autorisation écrite.

0.4 Compatibilité

- WordPress : compatible avec les versions récentes de WordPress (version minimale à préciser selon déploiement)
- PHP : version recommandée ≥ 8.1
- Thème parent requis : Astra (version minimale à préciser
- Plugins requis : Advanced Custom Fields (ACF)
- Plugins optionnels : aucun

0.5 Objectif du thème

Framework WordPress destiné aux sites institutionnels simples, orienté performance, accessibilité et maintenance longue durée. Publicum fournit ses propres mises en page, un design system dynamique via ACF, un ensemble réduit de fonctionnalités et un socle technique minimaliste, sans dépendance au block editor.

1. Architecture générale

1.1 Positionnement technique du thème

Publicum Essentiel fournit une structure complète de pages prête à l'emploi, destinée à servir de base à la création de sites professionnels. Le thème s'appuie sur des modèles prédéfinis, une organisation standardisée des contenus et un système de design paramétrable depuis l'administration. Il est utilisable tel quel ou comme fondation technique pour des développements spécifiques.

L'approche privilégie une empreinte technique réduite : feuille de style unique, volume limité de JavaScript, absence de frameworks externes et désactivation de fonctionnalités non nécessaires. Les réglages d'apparence reposent sur des variables injectées dynamiquement à partir des paramètres saisis dans l'administration.

Publicum Essentiel vise à permettre au client final de gérer le site en autonomie sans intervenir dans les fichiers du thème. La structure technique reste stable et modifiable par le développeur si nécessaire.

ChatGPT a dit :

1.2 Thème parent

Publicum Essentiel repose sur un thème parent basé sur Astra. Le parent est requis pour assurer la structure minimale de fonctionnement du thème dans WordPress : chargement initial, hiérarchie par défaut des fichiers de template et compatibilité générale avec les versions courantes de WordPress.

Publicum ne s'appuie pas sur les composants visuels d'Astra. Les styles, scripts et options front-end du parent ne sont pas utilisés ; le rendu du site est assuré par les fichiers du thème enfant. Le thème parent sert de socle technique et de point d'ancrage pour le chargement du thème dans WordPress.

Le maintien du parent garantit une base stable tant que l'ensemble des gabarits n'est pas complètement réimplémenté côté enfant. Il reste possible d'adapter la structure pour se passer du parent si une reconstruction intégrale du socle est effectuée. Ce cas n'entre pas dans le périmètre de cette version.

contact@lapetitefabriquedigitale – lapetitefabriquedigitale.fr - 71 B avenue Gros Malhon 35000 Rennes - SIREN : 990 410 946

1.3 Structure interne du thème

Le thème est organisé autour de fichiers de gabarits situés à la racine et de répertoires dédiés aux composants techniques, aux modèles partiels, aux champs ACF et aux ressources.

Fichiers principaux à la racine :

- `functions.php` : déclarations principales du thème, chargement des ressources, désactivation de composants WordPress et initialisation des options.
- `style.css` : feuille de style unique du thème.
- `header.php`, `footer.php`, `page.php`, `index.php`, `home.php` : gabarits de structure.
- `404.php`, `403.php` : pages système.
- `search.php`, `searchform.php` : pages et formulaires de recherche.
- `archive.php`, `archive-actualites.php` : gabarits d'archives.
- `single-actualite.php` : affichage des actualités.
- modèles de pages spécifiques :
    - `template-publicum.php`
    - `template-home.php`
    - `template-contact.php`
    - `template-plan.php`
    - `template-footer.php`
    - `template-cartes.php`
    - `template-agenda.php`
    - `template-accordeon.php`

Dossiers principaux :

- `assets/`
  Contient les fontes utilisées par le thème. Aucune image n'est chargée depuis ce dossier dans la version documentée.
- `inc/`
  Composants techniques du thème, organisés en sous-dossiers :

    - `acf-fields/` : groupes de champs utilisés par les pages et modèles.
    - `cpt/` : types de contenus déclarés, dont `cpt-actualites.php`.
    - `sitemap.php` : génération du plan du site.
- `js/`
  Contient `scripts.js`, chargé en front-end pour les interactions nécessaires.
- `partials/`
  Contient des éléments réutilisables, dont le sprite SVG utilisé dans l'administration.
- `template-parts/`
  Sections appelées par les modèles de pages. Chaque section correspond à une partie spécifique du rendu et est incluse dans les templates correspondants.

L'ensemble des fichiers de présentation est situé dans le thème enfant. Les gabarits du parent ne sont pas utilisés pour l'affichage.

1.4 Système de contenu

Publicum Essentiel organise le contenu autour d'un ensemble limité de modèles de pages, complétés par un type de contenu dédié aux actualités. Chaque modèle correspond à une structure prédéfinie du front-end et à un ensemble de champs configurés dans l'administration.

Modèles de pages principaux :

- `template-publicum.php` : structure générale utilisée comme base.
- `template-home.php` : page d'accueil.
- `template-contact.php` : formulaire de contact.
- `template-cartes.php` : présentation sous forme de blocs visuels.
- `template-accordeon.php` : contenu structuré en panneaux déroulants.
- `template-agenda.php` : page dédiée à la prise de rendez-vous.
- `template-plan.php` : plan du site.
- `template-footer.php` : gestion du pied de page.

Les pages sont liées à leurs modèles par leur identifiant ou leur slug. Le choix du modèle ne se fait pas via l'attribut "Modèle de page" de WordPress ; il est déterminé automatiquement selon la structure du site.

Actualités :
Un type de contenu spécifique est utilisé pour les actualités. Il dispose de son gabarit d'archive et de son gabarit individuel. Le système repose sur un type dédié sans utiliser le mécanisme de blog natif de WordPress, ce qui permet d'ajouter un blog distinct si nécessaire.

Organisation des champs :
Les champs de contenu sont déclarés dans le thème via Advanced Custom Fields, dans le dossier `inc/acf-fields/`. Chaque groupe de champs correspond à un modèle ou à une section de page. Les valeurs définies dans l'administration alimentent directement les templates.

Comportement général :

- contenu structuré via des champs spécifiques plutôt que via un éditeur libre ;
- affichage conditionnel selon la présence ou l'absence de champs ;
- aucun contenu critique n'est dépendant du JavaScript pour l'affichage.

Cette organisation vise à fournir une structure de contenu stable, prévisible et adaptée à une utilisation autonome du back-office.

1.5 Système de design

Le design du site repose sur une feuille de style unique associée à des variables CSS générées depuis l'administration. L'ensemble des règles de mise en forme est centralisé dans `style.css`, qui contient la typographie, la mise en page, les composants, les formulaires et la navigation.

L'usage d'une feuille de style unique répond aux contraintes d'éco-conception :

- réduction du poids total des fichiers ;
- absence de feuilles redondantes ;
- traitement limité lors du rendu par le navigateur.
  Cette approche permet de diminuer la consommation de ressources et de simplifier la maintenance des styles.

Les réglages graphiques sont contrôlés par des variables CSS. Les paramètres définis dans l'administration (couleurs, polices, bordures, proportions typographiques) sont injectés dans le <head> puis utilisés par `style.css`. Les modifications d'apparence ne nécessitent pas de modification des fichiers du thème.

Le système est organisé autour du module *Design Publicum* dans l'administration. Il regroupe les réglages accessibles à l'éditeur du site :

- palette de couleurs ;
- choix des polices ;
- taille du texte et titres ;
- bordures des boutons et cartes.

Ce fonctionnement permet :

- une gestion autonome de l'apparence par l'utilisateur non technique ;
- une base cohérente pour l'accessibilité (structure stable, styles contrôlés) ;
- un contrôle direct sur l'impact des choix graphiques sur la charge front-end.

Organisation :

- règles principales dans `style.css` ;
- champs d'options dans `inc/acf-fields/` ;
- génération des variables CSS dans `functions.php`.

1.6 Scripts et comportements

Le fichier JavaScript principal est `scripts.js`, situé dans le dossier `js/` et chargé en différé. Le script ne dépend pas de jQuery ou de frameworks externes. Il intervient uniquement sur des fonctionnalités interactives nécessaires au fonctionnement du site.

Rôles principaux du script :

- gestion du menu mobile (ouverture, fermeture, focus, navigation clavier) ;
- réglages d'accessibilité liés à la navigation (états ARIA, gestion du focus, masquage conditionnel) ;

contact@lapetitefabriquedigitale – lapetitefabriquedigitale.fr - 71 B avenue Gros Malhon 35000 Rennes - SIREN : 990 410 946

- bascule typographique "Luciole" avec mémorisation locale ;
- surlignage de termes dans le contenu via le paramètre `?sk`.

Les comportements de navigation et d’affichage critiques restent fonctionnels sans exécution du script ; il intervient pour améliorer l’ergonomie et l’accessibilité lorsque JavaScript est disponible.

Le script est chargé uniquement en front-end et ne modifie pas les contenus structurants. Il n’introduit pas d'animations destinées à l’habillage graphique.

Ce choix limite la quantité de code exécuté par le navigateur, facilite l’audit d’accessibilité et réduit la maintenance associée à des dépendances tierces.

2. Fonctionnement structurel

2.1 Templates associés automatiquement via slug

Certaines pages clés sont reliées à des templates spécifiques selon leur *slug*. Le choix du modèle ne repose pas sur les réglages WordPress visibles dans l’édition, mais sur une correspondance automatique définie dans `functions.php`. Cela évite les erreurs de sélection, garantit l’homogénéité graphique et limite les manipulations côté client.

Correspondances natives

- `accueil` → `template-home.php`
- `plan-du-site` → `template-plan.php`
- `pied-de-page` → `template-footer.php`
- `nous-ecrire` → `template-contact.php`
- `prendre-rdv` → `template-agenda.php`

Cette règle est prioritaire sur les choix manuels dans l’interface ; si un modèle est défini automatiquement, l’option de changement de template est masquée côté admin.

Objectifs

- cohérence visuelle et structurelle
- réduction des risques de casse par modification de modèle
- simplification de l’administration côté client (aucun choix de template)

Fallback et sécurité

Si le template attendu est introuvable, WordPress utilise le modèle par défaut sans bloquer le rendu. Cela permet d’ajouter ou retirer des templates sans erreur critique.

Impact sur l’intégration

contact@lapetitefabriquedigitale – lapetitefabriquedigitale.fr - 71 B avenue Gros Malhon 35000 Rennes - SIREN : 990 410 946

Toute nouvelle page destinée à remplacer l'un de ces rôles doit utiliser un slug identique ou modifier la table de correspondance dans `functions.php`.

2.2 CPT activés

Le thème active un seul type de contenu personnalisé fonctionnel par défaut : `actualites`.

Ce CPT remplace l'usage natif des *posts* (désactivé dans le thème) et fournit une base éditoriale structurée sans activer de logique de blog complète.
Objectif : permettre une publication d'articles institutionnels (communiqués, actualités, événements, informations publiques) tout en laissant la possibilité d'ajouter un vrai blog ultérieurement via un autre CPT ou plugin.

Spécifications principales

- Nom interne : `actualites`
- Visibilité front : oui
- Administration visible dans les sous-menus personnalisés, mais la page statique portant le slug `actualites` est masquée dans la liste des pages (évite les erreurs d'édition)
- Aucun template de single répliquant les conventions blog WordPress (structure simplifiée, pas de taxonomies publiques)

Raisons du choix

- pas de blog natif WordPress → cohérence fonctionnelle avec un site institutionnel
- cloisonnement entre contenu éditorial et pages système
- intégration plus simple dans les modules de page (ex. actualités sur la home)

2.3 ACF liés

Les contenus et réglages du thème sont pilotés par une série de groupes ACF, chacun correspondant à un module fonctionnel précis.
Chaque groupe a son fichier dédié dans `/inc/acf-fields/`.
On documente uniquement le rôle global de chaque groupe, pas le détail des champs.

Liste des groupes ACF utilisés

- `acf-accueil-fields`
  Groupe central pour la page d'accueil.
  Il contrôle notamment l'activation ou non des différentes sections (CTA, accès rapides, actualités, alertes, brèves, infos, logos, etc.) ainsi que leurs titres via un module placé en bas de la page d'accueil.
- `accordeon-fields`
  Configuration des blocs en accordéons (titres, contenus, ordre) pour les pages utilisant le template accordéon.
- `breves-fields`
  Gestion des brèves (courtes informations mises en avant, par exemple sur la home).
- `cartes-fields`
  Champs pour les pages en cartes (éléments affichés sous forme de blocs visuels structurés).
- `contact-fields`
  Champs liés à la page de contact (textes, informations complémentaires, encadrés, éventuellement mentions RGPD côté contenu).
- `cta-fields`
  Blocs d'appel à l'action (CTA) utilisés sur la page d'accueil et/ou dans d'autres sections.
- `design-fields`
  Ensemble des réglages de design (palette de couleurs, options de bordures, tailles de textes, etc.) exposés dans la page d'options "Design PUBLICUM".
- `footer-fields`
  Contenus et réglages du pied de page (textes institutionnels, liens, éléments éditoriaux).
- `logos-fields`
  Gestion des logos partenaires / institutions (liste, ordre, liens éventuels).

contact@lapetitefabriquedigitale – lapetitefabriquedigitale.fr - 71 B avenue Gros Malhon 35000 Rennes - SIREN : 990 410 946

- `publications-fields`
  Champs spécifiques liés au CPT `actualites` (ou équivalent) pour enrichir les contenus d'actualités.
- `seo-fields`
  Champs SEO (titre, meta description) utilisés pour surcharger le `<title>` et la meta description par page.
- `template-page-fields`
  Champs communs aux pages structurées (titres alternatifs, intros, options de mise en page, éléments génériques de gabarit).

Principes d'organisation

- un fichier PHP par groupe, dans `/inc/acf-fields/`
- nom de fichier aligné sur le groupe (ex. `cta-fields.php`, `design-fields.php`, etc.)
- les options globales (design, favicon, etc.) sont stockées en page d'options et injectées ensuite dans le front (CSS, `<head>`)

2.4 Pages système

Ces pages ont un rôle fonctionnel ou structurel. Elles ne sont pas destinées à être supprimées ou modifiées dans leur architecture, même si certains contenus sont éditables via ACF.

Plan du site
Slug : `plan-du-site`
Template : `template-plan.php`
Génération automatique via `inc/sitemap.php`.
Page masquée dans la liste des pages (filtre `pre_get_posts`) pour éviter les modifications ou suppressions accidentelles.

Page contact
Slug : `nous-ecrire`
Template : `template-contact.php`
Contenu géré via `contact-fields.php`.
Formulaire Contact Form 7, avec ajustements spécifiques décrits en 2.9 (dépollution CSS/JS, RGPD, accessibilité).

Page agenda / prise de rendez-vous
Slug : `prendre-rdv`
Template : `template-agenda.php`
Page prévue pour afficher un agenda ou un système de prise de rendez-vous (interne ou externe).

Pied de page
Slug : `pied-de-page`
Template : `template-footer.php`
Contenu éditorial alimenté par `footer-fields.php`.
La structure HTML du footer reste contrôlée par le template.

Recherche
Formulaire de recherche : `searchform.php`
Résultats de recherche : `search.php`
Le périmètre et l'extension de la recherche aux champs ACF sont détaillés en 2.8.

Pages d'erreur
404 : `404.php`
403 : `403.php`, déclenchée via `template_redirect` lorsque le serveur renvoie une 403, afin de servir un template 403 WordPress plutôt qu'une page brute Apache.

2.5 Page d'accueil

La page d'accueil utilise le template :
`template-home.php`

Ce template assemble plusieurs *template parts* :

- `section-cta.php`
- `section-actualites.php`
- `section-publications.php`
- `section-breves.php`
- `section-logos.php`

`D'autres sections sont disponibles mais peuvent rendre la page d'accueil un peu plus lourde.`

Section Actualités
`section-actualites.php` affiche les contenus du CPT `actualites`.
Aucun réglage dans l'édition de la page d'accueil : la gestion du contenu se fait uniquement via le CPT.

Activation et titres des sections
Un groupe de champs dédié, situé en bas de la page et défini dans :
`/inc/acf-fields/acf-accueil-fields.php`
permet d'activer/désactiver chaque section et d'en modifier le libellé.
Aucune autre logique conditionnelle n'est définie dans `template-home.php`.

2.6 Sprite SVG

Le thème utilise un sprite SVG unique pour centraliser toutes les icônes utilisées sur le site.
Fichier : `partials/sprite.php`
Chargé dans le front et dans l'administration.

Contenu du sprite
Il inclut notamment :

- Icônes du moteur de recherche (loupe)
- Icônes réseaux sociaux
- Icônes PDF / téléchargement
- Autres pictogrammes décoratifs ou fonctionnels

Objectif
Réduire le nombre de requêtes et limiter les ressources à charger, conformément aux choix d'éco-conception du thème.

Ajout d'une icône
Ajouter un SVG directement dans le sprite suffit pour qu'il soit disponible à l'appel dans les templates.

Emplacement des appels
Les templates et *template parts* référencent les icônes via `<use>` pointant vers les identifiants du sprite.

2.7 Attribution automatique des templates via URL

Certaines pages structurées peuvent être créées directement depuis l’interface « Gestion du site » sans passer par le sélecteur de modèles natif. Le template est attribué automatiquement lors de la création via un paramètre dans l’URL.

Exemples d’URL

- /wp-admin/post-new.php?post_type=page&page_template=template-cartes.php
- /wp-admin/post-new.php?post_type=page&page_template=template-accordeon.php

Templates concernés

- template-cartes.php
- template-accordeon.php

Fonctionnement technique

- Hook load-post-new.php : attribution du template dès la création de l’auto-draft.
- Hook save_post_page : réapplication si nécessaire.

Rôle dans l’administration

- Création de pages structurées via les boutons du tableau de bord « Gestion du site ».
- Le champ « Modèle » est masqué pour éviter les erreurs de configuration.

2.8 Recherche étendue ACF

La recherche interne est étendue pour prendre en compte des champs ACF spécifiques. Les résultats incluent les pages et les articles, mais pas le CPT `actualites`.

Types de contenus recherchés

- pages
- posts
- uniquement les contenus publiés

Champs ACF pris en compte
(la liste est définie dans le code dans le tableau `$acf_keys_exact`)

- page_text
- page_intro
- page_subtitle
- page_title

Fichiers concernés

- `searchform.php` : formulaire de recherche
- `search.php` : affichage des résultats

Comportement

- La recherche WordPress reste standard (formulaire et affichage natifs).
- L'extension porte uniquement sur l'ajout de correspondances dans les champs ACF listés.
- Aucun réglage n'est disponible sur la page d'accueil pour cette fonctionnalité.

Limites

- Le CPT `actualites` n'est pas inclus dans les résultats.
- Seuls les champs explicitement listés sont recherchés.

2.9 Intégration CONTACT FORM 7

Le thème prend en charge Contact Form 7, mais applique plusieurs ajustements afin de limiter l'impact sur la performance, d'éviter l'injection automatique de CSS, et de renforcer la conformité accessibilité.

CSS et scripts désactivés
Les feuilles de style Contact Form 7 (y compris variantes Astra) sont retirées pour éviter la surcharge front.
Le script global `wp-embed` et certains assets liés sont également supprimés lorsqu'ils ne sont pas nécessaires.

Injection sélective
Les styles CF7 ne sont chargés que sur la page Contact.
Aucun chargement automatique sur les autres pages.

Accessibilité RGPD : case d’acceptation
Sur la page Contact uniquement : ajout automatique d’un `id` unique sur la case à cocher RGPD afin d’assurer un lien explicite label/checkbox.

Limitation des pièces jointes
Validation serveur empêchant l’envoi lorsque la taille cumulée dépasse 10 Mo.
Message utilisateur explicite.

Comportement attendu
• Formulaire léger, sans styles externes imposés
• Compatibilité avec le design du thème
• Respect RGPD / accessibilité sans surcharge technique

2.10 Gestion du site (tableau de bord dédié)

Le thème inclut un tableau de bord dédié accessible via le menu « Gestion du site ».
Objectif : regrouper les actions essentielles et éviter l’accès à l’interface WordPress par défaut, jugée trop complexe pour des utilisateurs non techniques.

Localisation
template-parts/admin-dashboard.php
Chargé dans l’administration mais invisible côté front.

Accès
Ajouté via add_menu_page() avec la capacité minimale read, ce qui le rend accessible même aux comptes non administrateurs selon configuration.

Contenu principal
• Création guidée de pages avec pré-sélection de templates (page classique, cartes, accordéon).
• Accès direct à la gestion du CPT Actualités.
• Accès rapide à la médiathèque et aux pages existantes.
• Lien direct vers la page « Design PUBLICUM » (ACF options).
• Accès menus, utilisateurs, options générales.

Structure interne
Le tableau est structuré en blocs fonctionnels :
• Pages et contenus
• Design et configuration

• Entretien et vérifications
• Aide et support

Chaque bloc regroupe des actions sous forme de boutons rapides pour limiter les erreurs et réduire les temps d'apprentissage.

Comportement attendu
• Simplification radicale de l'administration.
• Réduction de l'errance utilisateur dans le back-office WordPress.
• Standardisation des entrées de contenu (évite erreurs de template et création non conforme).

Point important
Ce tableau de bord ne remplace pas les écrans WordPress natifs mais agit comme un hub central, cohérent avec la démarche d'éco-conception et de minimisation cognitive du projet.

2.11 SEO minimal intégré

Le thème intègre un système SEO léger basé sur ACF, permettant pour chaque page :

• Champ « seo_title » → injecté dans `<title>` via le filtre `pre_get_document_title`
• Champ « seo_description » → injecté dans `<meta name="description">` dans `<head>`

Objectif : fournir un contrôle éditorial minimal sans scripts externes ni surcharge fonctionnelle.

Caractéristiques :
• Aucune génération de données structurées (schema.org)
• Pas d'OpenGraph / Twitter Cards
• Pas de sitemap XML public (uniquement le sitemap HTML interne prévu par le thème)
• Pas de gestion des redirections ni canonicals personnalisés

Ce système ne remplace pas un plugin SEO complet.

3. Design

3.1 Palette Design PUBLICUM

La palette Design PUBLICUM est gérée via la page d’options ACF design-fields.php.
Elle permet de modifier la plupart des choix visuels sans toucher au code, tout en respectant les contraintes d’éco-conception et d’accessibilité du thème.

Localisation et accès

- Page d’options ACF : « Design PUBLICUM »
- Slug : design-publicum
- Accès : menu Apparence > Design PUBLICUM
- Déclarée via acf_add_options_page() dans functions.php

Éléments configurables

Couleurs
La page d’options permet de définir l’ensemble de la palette front :

- Verts (ex. color_green_dark, color_green_medium, color_green_forest, color_green_light)
- Neutres (white, gray_light, gray_medium, gray_icons, gray_dark)
- Beiges (beige_light, beige_dark)
- Oranges (orange_light, orange_hover, orange, orange_dark, orange_vivid)
- Couleur d’accent (color_green_accent)

Ces valeurs sont stockées dans des champs ACF (ex. color_green_dark) et injectées en variables CSS (ex. --color-green-dark) dans le head.

Typographie
Plusieurs options typographiques sont exposées :

- Taille du texte (font_size_text : small / normal / large)
- Taille des titres (font_size_titles : small / normal / large)

contact@lapetitefabriquedigitale – lapetitefabriquedigitale.fr - 71 B avenue Gros Malhon 35000 Rennes - SIREN : 990 410 946

- Choix des polices pour le texte, les titres et le menu (font_text, font_titles, font_menu)

Les piles de polices sont générées dynamiquement via un style inline dans le head :

- --font-base pour le texte
- --font-serif pour les titres
- --font-menu pour le menu

Un préchargement (preload) est ajouté pour la police utilisée dans le menu principal, à partir du dossier assets/fonts/.

Bordures et composants

La page Design PUBLICUM permet également de contrôler les bordures :

- Activation / désactivation de la bordure des boutons (btn_border_enable, btn_border_width, btn_border_color)
- Activation / désactivation de la bordure des cartes (card_border_enable, card_border_width, card_border_color)

Ces valeurs sont exportées sous forme de variables CSS :

- --btn-border-width, --btn-border-color
- --card-border-width, --card-border-color

Injection technique

Les réglages Design PUBLICUM sont injectés dans le front via deux blocs principaux dans functions.php :

- Un bloc qui génère un style inline `#design-dynamic-vars` avec toutes les variables CSS (couleurs, tailles, bordures).
- Un bloc qui génère un style inline `#publicum-dynamic-fonts` pour les piles de polices.

Ce mécanisme permet :

- D’éviter des feuilles de style supplémentaires.
- De limiter le volume de CSS généré.
- De conserver un lien direct entre ACF et les variables CSS utilisées dans style.css.

3.2 Gestion des images

Les images sont gérées via la médiathèque WordPress. Le thème n’intègre aucun système automatique de compression ou d’optimisation : les fichiers doivent être préparés avant import.

Formats recommandés

• Format principal : WebP
• Formats lourds à éviter : JPEG haute qualité, PNG non optimisés
• SVG réservé aux icônes ou éléments de branding (poids faible, contrôle manuel, vérification accessibilité)

Tailles conseillées à l’upload

• Sections pleine largeur : environ 1920 px
• Visuels internes et cartes : environ 600 px
• Visuels mobiles simples : environ 400 px
• Icônes : SVG ou raster fortement optimisé (≈ 20 Ko max)

Politique d’optimisation

• Redimensionnement et compression à effectuer avant import
• Aucune optimisation automatique fournie par le thème
• Lazy-load natif WordPress utilisable si activé au niveau du projet
• Certaines tailles automatiques de WordPress sont désactivées pour limiter la génération de fichiers (détails en 6.4)

Recommandations pour l’enregistrement des images

• Remplir systématiquement l’attribut ALT (texte uniquement si l’image porte du sens, vide si décorative)
• Format recommandé : WebP (PNG/JPEG uniquement si contrainte spécifique)
• Compression modérée (~80–90, éviter les artefacts sur texte/schéma)
• Taille adaptée au contexte d’affichage (ex. 1920px plein écran, 600px interne, 400/200px mobile)
• Nom de fichier explicite, sans espaces ni accents
• Poids modéré (quelques dizaines de Ko ; éviter >300Ko)
• Une seule version uploadée : le thème génère les tailles nécessaire

Responsabilité

L’optimisation finale (dimensions, format, compression) relève du projet.
Le thème fournit un cadre minimaliste et performant mais ne prend pas en charge la réduction du poids des médias.

4. Comportements désactivés

Le thème désactive plusieurs fonctionnalités natives de WordPress et d’Astra afin de réduire les scripts chargés, éviter les doublons CSS/JS, limiter la surface d’attaque et supprimer les effets visuels non souhaités. Ces choix répondent à une logique d’éco-conception et de structure “minimal front / administration guidée”.

Désactivations WordPress (cœur)

Les comportements suivants sont neutralisés par défaut :

- Éditeur de blocs (Gutenberg)
- wp-emoji-release
- wp-embed + oEmbed complet
- flux RSS / Atom (liens et endpoints)
- global-styles-inline-css
- speculative rules
- commentaires natifs
- JQuery-migrate
- styles et scripts liés au block editor

Ces désactivations réduisent la charge frontale et évitent l’injection automatique de CSS et JS non utilisés.

Désactivations Astra

Les éléments suivants sont explicitement coupés pour éviter la sortie automatique de composants de thème parent :

- JS principal Astra
- Icônes toggle menu (mobile)
- Styles Contact Form 7 ajoutés par Astra
- Meta-boxes Astra dans l’édition de pages
- Navigation mobile injectée par défaut

Ces désactivations permettent d’assurer un rendu visuel et un DOM strictement maîtrisé.

Désactivation du type de contenu "post"

Le type de contenu natif `post` est désactivé car Publicum fournit un CPT dédié (`actualites`) spécifiquement configuré pour les usages du thème.

Effets :

- menu "Articles" supprimé dans l'administration
- accès directe aux contenus d'actualité via le CPT dédié
- pas de confusion avec le blog WordPress natif (le blog peut être ajouté ultérieurement si nécessaire)

Flux, embed, XML-RPC, global styles et assets automatiques

Sont neutralisés :

- XML-RPC
- flux RSS / feeds
- embed REST
- ajout automatique d'assets front (fonts, styles global styles, inline styles)
- transformations automatiques des liens en embeds

Objectif : éviter les endpoints inutiles, supprimer les injections CSS/JS invisibles et limiter l'exposition à des services non utilisés dans le cadre de Publicum.

5. Personnalisation développeur

5.1 Fichiers modifiables

Les fichiers suivants peuvent être adaptés en fonction du projet :

– style.css
Ajustements de mise en forme spécifiques au client (compléments ou adaptations locales).

– js/scripts.js
Ajout de comportements front supplémentaires ou adaptations mineures.

– templates supplémentaires
Ajout de nouveaux modèles de pages à la racine du thème, en suivant la logique existante (exemple : templates de pages structurées).

– inc/acf-fields/ (ajouts uniquement)
Ajout de nouveaux groupes ACF pour répondre à des besoins projet, sans modifier les groupes existants.

– palette et design
Les couleurs, polices et bordures peuvent être modifiées soit dans functions.php via la section Theme Options (injection CSS dynamique), soit en ajustant directement les valeurs correspondantes dans design-fields.php. Les deux approches sont valides selon le niveau de contrôle souhaité.

5.2 Fichiers à ne pas modifier

Les éléments suivants constituent la base du thème et ne doivent pas être modifiés, sauf développement maîtrisé :

– functions.php
Chargement CSS unique, désactivations WP/Astra, SEO, search étendue, design dynamique.

– inc/cpt/actualites.php
Définition du CPT Actualités qui remplace l’usage du type post.

– inc/sitemap.php
Génération automatique du plan du site.

– inc/acf-fields/design-fields.php
Définition des options de design global.

– partials/sprite.php
Structure du sprite (on peut ajouter des SVG, mais ne pas altérer la structure globale).

– templates système
template-home.php, template-plan.php, template-contact.php, template-agenda.php, template-footer.php, 403.php, 404.php. Ils assurent la cohérence fonctionnelle du thème.

Règle ACF centrale

Les groupes ACF livrés avec le thème ne doivent pas être renommés ni supprimés. Toute extension du schéma se fait par ajout de nouveaux groupes/champs, pas par modification des clés existantes.

5.3 Points d’entrée (hooks et filtres)

Le thème peut être étendu en utilisant les hooks WordPress standards suivants :

– template_include
Permet d’ajouter ou de remplacer des templates sans toucher à la logique centrale.

– filtres SEO (pre_get_document_title, wp_head)
Adaptations locales des règles de titre et description.

– filtres de navigation (walker_nav_menu_start_el, nav_menu_link_attributes)
Modification structurelle du balisage HTML ou des attributs ARIA.

– recherche (pre_get_posts, posts_search)
Possibilité d'étendre ou restreindre les contenus pris en compte par la recherche interne en conservant la logique existante.

– Contact Form 7 (filtres de validation et hooks d’intégration)
Adaptations des comportements spécifiques au projet.

5.4 Arborescence principale

L’arborescence détaillée du thème et la fonction de chaque répertoire sont documentées au chapitre 1.3.
Ce chapitre ne duplique pas ces informations ; il rappelle simplement que la structure doit être conservée telle quelle pour garantir la stabilité du thème.

6. Performance

Le thème applique une stratégie de performance centrée sur la réduction du poids total, du nombre de requêtes, et des opérations de rendu. Les optimisations concernent principalement la structure CSS, la gestion du JS, la limitation des traitements WordPress automatiques, et la gestion des images.

6.1 CSS unique

Le thème utilise une feuille de style unique.
Elle regroupe l'ensemble des styles du site, afin de limiter les requêtes HTTP, éviter les conflits entre feuilles multiples et réduire le parsing CSS.

Organisation interne de la feuille :

- styles génériques du site (HTML, body, typographies, couleurs) ;
- composants récurrents (boutons, cartes, blocs génériques) ;
- styles par page ou par section (groupés par ordre logique) ;
- ajustements d'accessibilité (focus, modes alternatifs) ;
- media queries globales (mobile → desktop) ;
- section `@media print`.

Justification : limiter le nombre de fichiers à charger, réduire les recalculs de style et stabiliser les performances (TTFB, LCP).

6.2 Scripts

Un script principal unique est chargé sur le front (`/js/scripts.js`).
Le chargement est différé, afin de ne pas bloquer le rendu initial.
Pas de jQuery utilisé par le thème. WordPress ne charge pas `jquery-migrate` ni les scripts d'éditeur de blocs.

Objectif : réduire le JS dans le chemin critique, diminuer le TBT et stabiliser l'affichage.

contact@lapetitefabriquedigitale – lapetitefabriquedigitale.fr - 71 B avenue Gros Malhon 35000 Rennes - SIREN : 990 410 946

6.3 Editeur de blocs

L'éditeur de blocs WordPress est désactivé et ses feuilles de style ne sont pas chargées sur le front.
Justification : réduire les CSS génériques, conserver un markup propre et maîtriser l'empreinte front.

6.4 Fonctionnalités WordPress automatiques

Fonctionnalités supprimées ou neutralisées pour réduire les sorties HTML, scripts injectés et endpoints réseau :

- flux RSS,
- oEmbed,
- XML-RPC,
- `wp-embed`,
- speculation rules WordPress,
- styles globaux WordPress.

Objectif : réduire les scripts automatiques inutiles et la surface de fonctionnalités inactive.

6.5 Gestion des images

Les tailles lourdes générées nativement par WordPress sont désactivées.
Le thème déclare uniquement des tailles réellement nécessaires au design.

Tailles définies par le thème :

- `publicum-full` : 1920 px, largeur pleine ;
- `publicum-xtralarge` : 900 px, usage CTA Mobile ;
- `publicum-large` : 600 px, usage desktop standard ;
- `publicum-medium` : 400 px, usage mobile ;
- `publicum-compact` : 200 px ;

- `publicum-small` : 120 px.

Justification : réduire le nombre de fichiers générés par upload, limiter le poids stocké, servir des visuels adaptés.

Ce chapitre ne traite pas de la préparation des images (format, compression) : ces recommandations sont décrites au chapitre « Gestion des images ».

6.6 Plugins

Le thème fonctionne sans plugin de performance lourd.
La configuration de référence utilise des outils légers et courants :

- ACF Pro (structuration du contenu),
- Contact Form 7 (formulaire),
- WP Mail SMTP,
- WPvivid (sauvegardes, remplaçable par tout plugin équivalent),
- WP Fastest Cache (cache léger),
- Plugin de sécurité basique (par ex. Solid Security Basic),
- Smush (optimisation d'images côté client si besoin).
- Yoast SEO (ou un autre plugin de SEO)

Rien n'impose l'usage d'un plugin de minification ou d'agrégation des ressources **si les images sont correctement redimensionnées en amont et enregistrées en WebP**.

En cas d'ajout de plugins supplémentaires, privilégier **les solutions les plus légères** afin de préserver la démarche d'éco-conception.

contact@lapetitefabriquedigitale – lapetitefabriquedigitale.fr - 71 B avenue Gros Malhon 35000 Rennes - SIREN : 990 410 946

7. Accessibilité

Le thème applique une base technique conforme aux pratiques d'accessibilité.
Certaines exigences supplémentaires dépendent du contenu ajouté par l'utilisateur.

7.1 Actions automatiques du thème

Le thème applique nativement :

- navigation clavier complète (focus logique, retour focus, fermeture ESC)
- indicateur de focus visible (`body.user-is-tabbing`)
- menu accessible (balisage ARIA, `aria-haspopup`, `aria-expanded`, verrouillage du focus, suppression des décorations non pertinentes)
- SVG décoratifs masqués (`aria-hidden="true"`, `focusable="false"`)
- police Luciole activable (bascule persistée, `aria-pressed`)
- structure ARIA pour sous-menus
- affichage lisible jusqu'à 200 % sans perte de lisibilité ni superposition critique
- fonctionnement sans JavaScript actif (menu et contenus restent accessibles, navigation préservée)

Ces mécanismes sont intégrés au noyau du thème.

7.2 Points à contrôler selon les personnalisations

Certains choix éditoriaux ou graphiques peuvent dégrader l'accessibilité.
Vérifications recommandées :

- contraste couleur après personnalisation (notamment palette modifiée via ACF)
- alternatives textuelles aux images ajoutées par l'utilisateur
- cohérence des libellés de liens, CTA, titres ACF
- hiérarchie Hn en fonction des pages créées
- absence d'éléments porteurs d'information déposés uniquement comme images décoratives

- **ajout de médias lourds (audio/vidéo)** : le thème reste fonctionnel, mais ces contenus dégradent l’éco-conception et nécessitent sous-titres, transcription, contrôle clavier et alternatives.
  Ils sortent du périmètre du thème, qui est conçu pour rester léger.

7.3 Ajustements spécifiques au menu

Le menu implémente :

- ouverture/fermeture au clavier et à la souris
- retour du focus au déclencheur à la fermeture
- verrouillage du focus dans le contenu ouvert
- prise en charge `Escape`
- masquage du fallback input/checkbox quand JS actif
- balisage cohérent pour items avec sous-menus (`menu-item-has-children`)

L'ajout d'éléments dans le menu doit maintenir cette structure.

8. Limites et périmètre

8.1 Ce que le thème ne fournit pas

• Pas de builder visuel (Elementor, Divi, Bricks, Gutenberg est désactivé)
• Pas de fonctionnalités e-commerce natives (WooCommerce ou équivalent à installer et intégrer manuellement)
• Pas de mise en cache avancée, minification ou optimisation automatique au-delà de Fastest Cache
• Pas de plugin SEO tiers (Yoast, SEOPress, etc. non inclus)
• Pas de gestion de newsletter, CRM ou automatisations externes
• Pas d’outils multimédia lourds (lecteurs audio/vidéo, carrousels complexes, lightbox avancée)

8.2 À réaliser côté projet / intégration

• Création et hiérarchisation des contenus éditoriaux
• Production des médias optimisés (images compressées, formats adaptés)
• Paramétrage SEO avancé si besoin (plugin au choix)
• Ajout éventuel d’un système d’agenda, d’un module boutique, d’un blog natif, ou autres fonctionnalités métier
• Mise en place des sauvegardes automatiques (plugin déjà installé mais à configurer selon l’hébergeur)

8.3 Comportements non supportés

• Ajout de médias lourds non optimisés (vidéo en fond, PDF volumineux, images haute définition non compressées)
• Ajout d’un second système de page builder ou d’un thème parent alternatif
• Réactivation de Gutenberg ou injection massive de blocs natifs WordPress
• Réintroduction de flux RSS, oEmbed et XML-RPC (désactivés pour sécurité et sobriété)
• Modifications de structure via plugins modifiant la sortie HTML (risques de rupture accessibilité et cohérence CSS)

8.4 Maintenance et mises à jour

• Les mises à jour de Publicum doivent préserver la structure existante :
– Pas de renommage des groupes ACF fournis
– Pas de suppression ou modification des fichiers système (`functions.php`, templates racine)
– Les ajouts doivent se faire via hooks, filtres, fichiers séparés ou plugins
• Aucun système de migration automatique entre versions : intégration à tester avant déploiement

contact@lapetitefabriquedigitale – lapetitefabriquedigitale.fr - 71 B avenue Gros Malhon 35000 Rennes - SIREN : 990 410 946

9. Licence & redistribution

9.1 Utilisation en marque blanche

Le thème peut être utilisé pour des projets clients, y compris en marque blanche, sans mention publique de Publicum. Chaque installation doit correspondre au nombre de licences effectivement acquises. Les agences peuvent adapter le thème (design, templates, ajustements CSS, création de modèles supplémentaires) pour l’intégrer dans leurs propres projets. Cette utilisation s’applique uniquement à des sites finaux livrés à des clients, et non à la revente du thème en tant que produit autonome.

9.2 Propriété et périmètre de diffusion

Le code source du thème est sous licence GPL, mais la structure, l’organisation interne, les groupes ACF, et l’ensemble de la documentation associée restent protégés et ne peuvent pas être revendues comme produit distinct ni commercialisées sous un autre nom. La diffusion publique d’un package dérivé ou d’une variante clé-en-main du thème nécessite un accord préalable. Dans le cadre d’un projet client, les modifications locales sont autorisées dès lors que la nomenclature générale et l’architecture ne sont pas republiées comme produit redistribuable.

9.3 Éco-conception et accessibilité dans les usages dérivés

Le thème fournit un socle performant, sobre et aligné avec les recommandations d’accessibilité. Ces propriétés sont conservées uniquement si les intégrations respectent les pratiques prévues : utilisation des tailles d’images fournies, absence d’overlays ou de scripts tiers intrusifs, maintien du balisage et des contrôles ARIA, typographies modérées, plugins légers, absence de vidéos lourdes en lecture automatique, etc. Les choix d’intégration ou d’ajout de fonctionnalités externes peuvent modifier le niveau d’éco-conception ou de conformité, et ces évolutions relèvent de la responsabilité du projet final.

9.4 Support professionnel

Une assistance peut être fournie pour l’installation, l’intégration ou l’ajout de modules complémentaires, sur devis ponctuel. Le fonctionnement est garanti dans la configuration fournie lors de la livraison. Lorsque la structure interne est modifiée en profondeur, que des groupes ACF sont renommés, ou que des plugins modifient le comportement de rendu, la stabilité et la conformité dépendent alors du projet intégrateur. Le thème reste néanmoins conçu pour permettre des évolutions contrôlées via CSS, ACF, et fichiers de template identifiés sans dépendance à des chaînes d’optimisation lourdes.

contact@lapetitefabriquedigitale – lapetitefabriquedigitale.fr - 71 B avenue Gros Malhon 35000 Rennes - SIREN : 990 410 946