

CHAPITRE 3 — Le principe de séparation entre contenu et design

Une analogie pour comprendre

Le rapport entre développeur et éditeur peut être comparé à celui qui unit un architecte et l'occupant d'un appartement.

Rôle	Responsabilité
L'architecte	Conçoit les murs, les ouvertures, les volumes, les circulations. Une fois le chantier livré, sa mission est terminée.
L'occupant	Choisit la peinture, le mobilier, l'agencement. Vit dans le logement et l'adapte à ses besoins, dans le respect des volumes existants.

InfraStudio applique le même découpage à votre site. Le développeur conçoit la structure et la livre. L'éditeur en remplit les contenus, sans toucher aux volumes.

Cette discipline est la condition pour qu'un site reste maintenable et cohérent dans la durée.

L'approche fondamentale du module se résume à un principe simple :

Le design relève de la responsabilité du développeur. Le contenu relève de celle de l'éditeur. Les deux périmètres ne se superposent jamais.

Cette séparation détermine l'architecture du module, le fonctionnement de l'éditeur, le découpage des permissions et même les fonctionnalités qui ne sont volontairement pas implémentées.

Pourquoi cette séparation est essentielle

Dans la majorité des outils de gestion de contenu, les frontières entre contenu et structure finissent par s'estomper. Le rédacteur a accès au HTML, au CSS, à la mise en page. À l'usage, il modifie sans s'en rendre compte des éléments structurels — parce que les outils l'autorisent à toucher à ce qui ne devrait pas l'être.

InfraSStudio adopte la position inverse : la structure est verrouillée et seules les zones explicitement marquées comme éditables par le développeur sont accessibles à l'éditeur.

Bénéfices observés

- **L'éditeur ne peut pas altérer la mise en page**

Il modifie uniquement les zones identifiées comme slots. Les conteneurs, les classes CSS et l'ordre des sections lui sont totalement inaccessibles.

- **Le design peut évoluer indépendamment du contenu**

Une refonte ne casse pas les contenus existants. Les valeurs des slots sont conservées et viennent simplement remplir la nouvelle structure.

- **La maintenance reste compréhensible dans la durée**

Plusieurs mois après la livraison, un autre développeur peut reprendre le code et identifier sans ambiguïté ce qui est éditable et ce qui ne l'est pas.

Les mécanismes qui matérialisent cette séparation

La séparation n'est pas une simple convention, elle est inscrite à plusieurs niveaux dans le fonctionnement technique du module.

Le HTML reste sous le contrôle du développeur

Le HTML d'une page Dolibarr Website est stocké dans deux emplacements synchronisés : la base de données et un fichier `.tpl.php` sur le disque. Le module ne propose à aucun moment d'éditer ce HTML. Il n'existe pas de bouton « voir le code source », pas d'éditeur de texte brut, pas d'option dissimulée. L'éditeur n'a accès qu'aux zones marquées comme slots, le reste lui demeure transparent.

Les slots sont détectés automatiquement

Lorsque le développeur ajoute un slot dans un template, aucune autre opération n'est nécessaire. Le module dispose d'un scanner qui parcourt les fichiers `.tpl.php` du site, détecte les tokens `{{slot:...}}`, les enregistre dans la table dédiée et génère l'interface d'édition correspondante.

```
<h2>{{slot:section_title|type=text|default=Nos services|label=Titre de la section}}</h2>
```

Au prochain rescan, manuel ou automatique, le slot apparaît dans le panneau d'édition. Aucune configuration intermédiaire n'est requise.

Le rendu public est transparent

Lorsqu'un visiteur consulte une page publique, le navigateur ne voit jamais de slot. Le module intercepte le HTML avant son envoi, remplace chaque token `{{slot:...}}` par sa valeur courante et transmet au visiteur un document HTML standard.

Cette substitution intervient au moment du rendu, jamais au moment de l'édition. Le fichier `.tpl.php` conserve toujours les tokens originaux, ce qui présente plusieurs avantages :

- Le développeur peut consulter et modifier le template sans craindre une pollution par les modifications de l'éditeur.
- Si vous purgez la table des slots, le site affichera simplement les valeurs par défaut, sans cesser de fonctionner.
- L'export du code d'un site et l'export de son contenu sont deux opérations distinctes, ce qui simplifie les migrations.

Les permissions traduisent la séparation

Le module définit sept permissions Dolibarr distinctes, qui permettent d'adapter précisément l'accès aux différents rôles :

Permission	Destinataire type
<code>paramMenu</code>	Tout utilisateur autorisé à voir le menu du module
<code>readContent</code>	Lecteur seul, par exemple un commercial qui consulte le site avant un rendez-vous
<code>editContent</code>	Rédacteur autorisé à modifier les slots de texte
<code>editTranslations</code>	Traducteur intervenant sur les versions linguistiques
<code>editMedias</code>	Gestionnaire des médias, autorisé à téléverser et remplacer des images
<code>publish</code>	Responsable de la publication finale des contenus
<code>admin</code>	Administrateur du module : configuration, sites gérés, gabarits

En pratique, un rédacteur reçoit en général `readContent` et `editContent`. Un traducteur reçoit `editTranslations`. L'intégrateur conserve `admin`. Aucune permission n'est attribuée au-delà du strict nécessaire.

Ce qui n'est volontairement pas proposé

Pour rester fidèle à ce principe, certaines fonctionnalités ne sont pas implémentées. Ce choix est délibéré.

Pas d'éditeur visuel par blocs

Il n'est pas possible de déposer des blocs à la souris pour composer une page. La structure reste codée par le développeur. Si une nouvelle section est nécessaire, deux options se présentent : utiliser un slot de type `richtext`, qui autorise une mise en forme limitée à l'intérieur d'une zone existante, ou demander au développeur d'ajouter un nouvel emplacement dans le template.

Pourquoi — Les éditeurs visuels par blocs produisent fréquemment des pages incohérentes lorsque l'utilisateur n'est pas formé à la mise en page. La structure verrouillée garantit la cohérence visuelle dans la durée.

Pas d'édition HTML brute

Même un éditeur expérimenté ne dispose pas d'un mode permettant de modifier le HTML directement. Les slots de type `richtext` proposent un éditeur visuel avec mise en forme (gras, italique, listes, liens, sous-titres) mais aucun bouton « source ».

Pourquoi — L'édition de HTML brut est la première source d'erreurs de mise en forme. Si une présentation spécifique est nécessaire, elle relève du CSS, et donc du développeur, non du contenu.

Pas de gestion de menus côté éditeur

Les menus de navigation (en-tête, pied de page) sont des pages Dolibarr Website particulières. Elles peuvent comporter des slots pour les libellés, le logo ou les liens sociaux, mais l'ordre des éléments et la structure restent codés dans le HTML.

Pourquoi — La navigation est un élément structurel du site, non un contenu éditorial. Sa modification doit être un acte délibéré du développeur, non un effet de bord d'une manipulation par l'éditeur.

Pas de logique conditionnelle dans les slots

Les slots ne portent aucune logique métier. Il n'est pas possible de définir une règle du type « afficher ce texte uniquement si l'utilisateur est connecté ». Les slots sont strictement des conteneurs de contenu.

Pourquoi — La logique relève du PHP, donc du développeur. Mêler logique et contenu rend les sites difficiles à maintenir au-delà de quelques années.

Revision #6

Created 2026-05-07 10:00:51 UTC by Lucky Ranasolonirina

Updated 2026-05-07 15:29:08 UTC by Lucky Ranasolonirina