

L i a m T A R D I E U

www.evogue.fr



➤ Sommaire

➤ Sommaire.....	2	➤ Questions stratégiques	17
➤ Présentation	3	<i>Les limites des CMS</i>	<i>17</i>
<i>Définition.....</i>	<i>3</i>	<i>Comment faire son choix ?</i>	<i>17</i>
<i>Fonctionnement.....</i>	<i>3</i>	<i>Quels sont les risques ?</i>	<i>18</i>
<i>Développement d'un site Internet.....</i>	<i>4</i>	➤ Open Source.....	19
➤ Les CMS, une évolution logique	5	<i>Qu'est-ce que c'est ?.....</i>	<i>19</i>
<i>Au XXe siècle : Un monde Sans CMS</i>	<i>5</i>	<i>Avantages / Inconvénients de l'open source</i>	<i>19</i>
<i>Internet et l'esprit de partage : Naissance des CMS.....</i>	<i>5</i>	➤ Licence propriétaire.....	20
<i>Pourquoi en est-on arrivé là ?.....</i>	<i>6</i>	<i>Un bon choix ?.....</i>	<i>20</i>
➤ L'arrivée des CMS et ses conséquences	7	<i>Avantages / Inconvénients des licences propriétaires</i>	<i>20</i>
<i>Les enjeux des CMS et l'impact sur le marché de l'emploi.....</i>	<i>7</i>	➤ Wordpress	22
<i>Les CMS ne remplacent pas la programmation.....</i>	<i>8</i>	<i>Description</i>	<i>22</i>
<i>Autonomie du client</i>	<i>8</i>	<i>Histoire</i>	<i>22</i>
➤ Notion : FrontOffice / BackOffice	10	<i>Récompenses</i>	<i>22</i>
<i>A quoi sert la séparation Back / Front ?</i>	<i>10</i>	<i>Installation</i>	<i>22</i>
<i>Accès.....</i>	<i>10</i>	<i>Personnaliser son thème</i>	<i>24</i>
<i>Techniquement, Comment cela est-ce possible ?</i>	<i>11</i>	<i>Les publications</i>	<i>25</i>
<i>Formation sur l'outil</i>	<i>11</i>	<i>Les Plugins.....</i>	<i>25</i>
➤ Principales caractéristiques des CMS	12	<i>Les widgets.....</i>	<i>25</i>
<i>Séparation entre contenu et présentation</i>	<i>12</i>	➤ Drupal.....	26
<i>Dissociation du design du site.....</i>	<i>12</i>	<i>Description</i>	<i>26</i>
<i>Édition de page simplifiée.....</i>	<i>12</i>	<i>Histoire</i>	<i>26</i>
<i>Séparation BackOffice / FrontOffice</i>	<i>12</i>	<i>Recompenses</i>	<i>26</i>
<i>Définition de droits d'accès des utilisateurs</i>	<i>13</i>	<i>Références.....</i>	<i>27</i>
<i>Avantages liés à l'utilisation d'un cms</i>	<i>13</i>	<i>Le module node</i>	<i>28</i>
➤ Faut-il systématiquement utiliser un CMS ?	14	<i>Les templates.....</i>	<i>28</i>
<i>Les points interessant pour le développeur</i>	<i>14</i>	<i>Le module PathAuto.....</i>	<i>29</i>
<i>Les points pouvant bloquer le développeur.....</i>	<i>14</i>	<i>Schéma</i>	<i>29</i>
<i>L'attrance des clients vers les CMS</i>	<i>15</i>	<i>Le module block.....</i>	<i>30</i>
<i>Quelle est la démarche qui conviendra le plus au client ?</i>	<i>15</i>	<i>Le module views</i>	<i>30</i>
<i>De quels clients parlons-nous ?.....</i>	<i>16</i>	<i>Le module cck.....</i>	<i>30</i>
<i>A quels besoins répondent les CMS ?</i>	<i>16</i>	<i>Les rôles</i>	<i>30</i>
		<i>Pourquoi créer son module.....</i>	<i>31</i>

➤ Présentation

DEFINITION

Un système de gestion de contenu (SGC) ou plus communément Content Management System pour CMS est une famille de logiciels destinés à la conception et à la mise à jour dynamique de sites Web ou d'application multimédia.

Plus précisément, il s'agit d'une application Web accessible, en général par un navigateur, qui permet de gérer le contenu d'un site Web sans avoir de connaissances informatiques poussées dans la conception de sites Web.

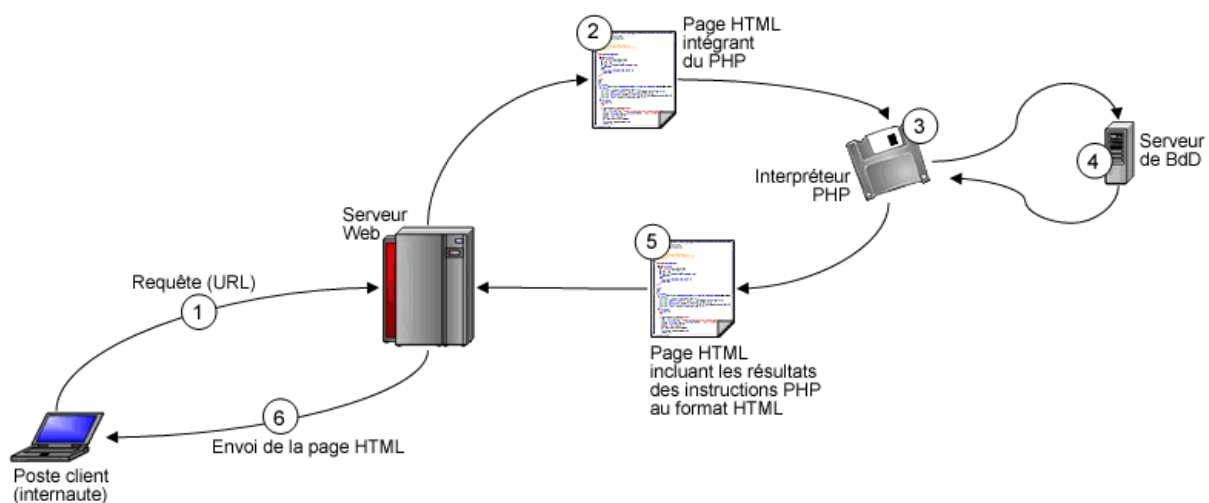
FONCTIONNEMENT

PHP

La plupart des CMS sont écrits à l'aide du langage PHP.

PHP (PHP : Hypertext Preprocessor) est un langage de scripts généralistes et Open Source, spécialement conçu pour le développement d'applications web.

PHP est un langage s'exécutant « coté serveur » et requiert l'architecture suivante pour fonctionner :



- L'internaute effectue une requête (depuis un lien hypertexte, ou en saisissant une URL) (1)
- La requête arrive sur le serveur web, qui transmet la page PHP à l'interpréteur PHP (2)
- L'interpréteur décode les instructions (3)
- Si nécessaire, l'interpréteur lance des requêtes sur le serveur de Base de Données (4)
- L'interpréteur retourne la page au serveur web. Cette page ne contient plus que du code HTML / CSS (5)
- Le serveur retourne la page à l'internaute (6)
- La page est interprétée par le navigateur de l'internaute

Wamp (www.wampserver.com)

Nous venons de le voir, PHP nécessite un serveur PHP distant.

Comme il n'est pas toujours très aisé de disposer en permanence d'un serveur de test distant, nous allons en installer un de façon locale.

WAMP (Windows, Apache, PHP, MySQL), intègre la version Windows de Apache (serveur Web), PHP, et MySQL (serveur de Bdd).

Il contient en plus le module PHPMyAdmin, qui permet la gestion des Bdd.

Son installation est indispensable pour la suite du cours.

DEVELOPPEMENT D'UN SITE INTERNET

Un site Internet est développé à l'aide de différents langages, mis en œuvre par des lignes de code formant un code source.

Ce code source est lu par le navigateur qui y accède et se charge par divers moyen de renvoyer le rendu de la page web à l'internaute.

Chaque langage à un rôle bien distinct au sein d'un site Internet.

Langage de conception :

Le HTML présente les informations.

Le CSS met en forme les informations déclarées via le html et s'occupe de la mise en page.

Langage de programmation :

Le PHP permet de développer des fonctionnalités sur le site.

Le SQL permet de formuler des requêtes vers une base de données.

CMS

Les CMS sont souvent désignés par le terme « logiciel », ceci dit, il est préférable de les désigner par le terme « solutions » car un CMS n'est autre qu'une série de plusieurs dossiers et de fichiers comportant au minimum les langages cités ci-dessus.



Les CMS, une évolution logique

AU XXE SIECLE : UN MONDE SANS CMS

Que serait le monde de l'informatique sans les CMS ?

- Le **client1** demande au **développeur 1** la création d'un site Internet e-commerce. il le met en œuvre en 5mois et le livre au **client1**.
- Le **client2** demande au **développeur 1** la création d'un site Internet institutionnel. il le met en œuvre en 3 semaines et le livre au **client2**.
- Le **client3** demande au **développeur 1** la création d'un site Internet « blog ».
il le met en œuvre en 2 mois et le livre au **client3**.
- ...
- Un jour le **développeur 1** reçoit le **client4** qui lui demande la création d'un site Internet e-commerce, et là, deux choix s'offrent à lui :
 - Reprendre le code qu'il avait mis en œuvre pour le **client1** et le modifier en fonction des besoins de son nouveau client (**client4**).
 - Repartir de zéro.

Il est évident que le **développeur 1** va repartir de l'existant et le modifier pour satisfaire le **client4**.

Il avait mis 5 mois pour faire le site du **client1**, s'il le reprend et qu'il le modifie, il prendra peut-être 1 mois pour effectuer les modifications et le livrer au **client4** et gagnera ainsi 4 mois (Dans le cas où le site e-commerce du **client1** est de la même envergure que le site du **client4**).

En développement, comme dans d'autres métiers, il est normal de réutiliser ce qui fonctionne et ne pas tout recommencer de zéro.

INTERNET ET L'ESPRIT DE PARTAGE : NAISSANCE DES CMS

Internet favorise l'échange de savoirs de façon exponentielle. Tout internaute un peu curieux peu apprendre rapidement sur le sujet de son choix, il trouvera de l'information sous différentes formes (explications, tutoriaux, discussion, vidéos, etc.) et, cela sera à lui de savoir l'exploiter.

Dès les débuts d'Internet, les développeurs se partageaient des morceaux de code source sur des forums et autres site Internet.

C'est dans cet esprit de partage que les CMS sont nés en reprenant les meilleurs codes pour chaque module (le terme module désigne une fonctionnalité d'un site Internet).

Techniquement, la création de CMS a été rendue possible dès l'apparition de certains langages (JavaScript, PHP, Java...) et le développement des bases de données relationnelles permettant à des rédacteurs non techniciens d'administrer un site Web et de produire son contenu.

Ces CMS rassemblent des modules que les développeurs mettent à jour de manière collaborative en le partageant aux autres en opposition avec l'esprit « chacun pour soi ».

Ainsi sur un CMS, vous activerez le module « boutique » pour obtenir une boutique « type » que vous pourrez modifier.

Vous pourrez aussi activer le module « forum » pour obtenir un forum « type » que vous pourrez modifier selon votre besoin.

Il est préférable d'utiliser le module forum d'un CMS plutôt que d'utiliser des bouts de code récupérés sur Internet pour créer le vôtre car le code du module sera certainement plus professionnel étant donné qu'il aura été mis à jour par une communauté de développeurs participant ainsi à l'évolution du CMS.

Sans les CMS, les développeurs fabriqueraient des sites Internet de leur côté pour leurs clients. Avec les CMS, les développeurs peuvent ainsi continuer à développer autour de la plateforme qu'ils auront choisie (le choix d'un CMS ne manque pas). Ils partagent leurs codes au reste de la communauté et font ainsi vivre le CMS.

Le développeur peut aussi reprendre des modules existant pour effectuer « un espace membre » et gagner un temps non négligeable, en retour le temps gagné par le développeur lui permettra peut-être de publier son propre code (sur un autre module) et d'en faire gagner à quelqu'un d'autre.

POURQUOI EN EST-ON ARRIVE LA ?

Sur un site web, si nous devons créer un espace membre, un blog, un forum ou autres.

Il ne serait pas forcément judicieux de tout créer de A à Z, pour au moins deux raisons :

- Il faut du temps pour créer ces fonctionnalités, les tester et les rendre fiables.
- Cela a déjà été fait par d'autres développeurs à plusieurs reprises.

Si les ingénieurs en matériel électronique devaient partir d'un tas de sable à chaque fois qu'ils conçoivent un nouveau dispositif, si leur première étape devait toujours consister à extraire le silicium pour fabriquer des circuits intégrés, ils ne progresseraient pas bien vite. Or, un concepteur de matériel construit toujours un système à partir de composants préparés, chacun chargé d'une fonction particulière et fournissant un ensemble de services à travers des interfaces définies. La tâche des concepteurs de matériel est considérablement simplifiée par le travail de leurs prédécesseurs.

La réutilisation est aussi une voie vers la création de meilleurs logiciels.

➤ L'arrivée des CMS et ses conséquences

LES ENJEUX DES CMS ET L'IMPACT SUR LE MARCHÉ DE L'EMPLOI

L'arrivée des CMS bouscule le monde des PME-PMI. Il convient de préciser que ces nouveaux produits ont déjà changé les méthodes de travail des entreprises et des clients.

D'autres conséquences sont-elles prévisibles ?

Depuis quelques années, l'arrivée de produits performants (le plus souvent en PHP) a modifié la façon de réaliser le premier système d'informations des petites entreprises. Ces produits se sont largement imposés, seront-ils pour autant présents dans l'avenir ?

Les entreprises attendent d'avantage de connaissances de la part des candidats lors des recrutements, cela a donc une répercussion inévitable sur le marché de l'emploi. La demande du marché peut-elle changer brusquement autour d'un seul CMS ?

Les écoles doivent de plus en plus adapter leurs formations en coordination avec les demandes des entreprises du moment. Est-ce un pari risqué ?

L'apprentissage des CMS ne remplace pas l'apprentissage des langages de programmation, il vient s'ajouter aux connaissances des étudiants, de manière à ce qu'ils soient formés aux derniers outils en vogue sur le marché.

On peut alors parler de "sous spécialité", un développeur peut être spécialisé PHP et sous spécialisé Joomla, un autre peut être spécialisé PHP et sous spécialisé TYPO3.

Auparavant, il était courant qu'un recruteur d'une webagency, cherche des développeurs maîtrisant le PHP. Aujourd'hui, il est de plus en plus fréquent de voir apparaître dans les annonces d'emploi, l'exigence que le développeur maîtrise un CMS en plus du PHP et des autres langages web. Avec le temps, il est possible que l'intitulé des offres d'emploi ne soit plus « stage PHP », mais « stage PHP – Magento » ou « stage PHP - Drupal ». Les écoles et les étudiants doivent s'adapter en vue de ces demandes.

Si le fait de maîtriser un CMS constitue un atout majeur pour un étudiant sur le marché de l'emploi, il ne doit pas néanmoins ne maîtriser que cela. Inutile de connaître 10 CMS. Il est préférable d'en connaître qu'un, voire deux, en profondeur et de garder l'autre part de ses connaissances pour les langages de programmation.

LES CMS NE REMPLACENT PAS LA PROGRAMMATION

Pour un étudiant, le risque est d'apprendre les solutions CMS au profit des langages eux-mêmes. Evidemment, il est plus facile et plus rapide d'accomplir des tâches grâce à un CMS, exemple : effectuer un espace membre, un forum, une boutique en ligne, etc. tout cela est possible juste en cochant une case mais il est indispensable de savoir le faire auparavant en PHP, à la main et en partant d'une page blanche.

La première raison, c'est que, à un moment donné, l'étudiant aura certainement besoin de modifier quelque chose dans le code pour arriver à ce qui lui est demandé, le CMS apporte beaucoup mais même s'il apporte 90% il y a toujours du spécifique demandé par le client. Sans connaissance en langage de programmation, cela paraît difficile à accomplir.

La seconde raison, c'est que le CMS sur lequel l'étudiant a choisi de se former n'est peut-être pas éternel. Il faut donc penser à surveiller ses arrières et ne pas « couler » en même temps que le CMS.

AUTONOMIE DU CLIENT

Au siècle dernier, les sites Web se présentaient sous une forme homogène et statique : le fond (le contenu) et la forme (le contenant) étaient indissociables.

De la création graphique, à la rédaction et à la publication des pages, tout passait par une équipe qui gérant et contrôlait le site Web.

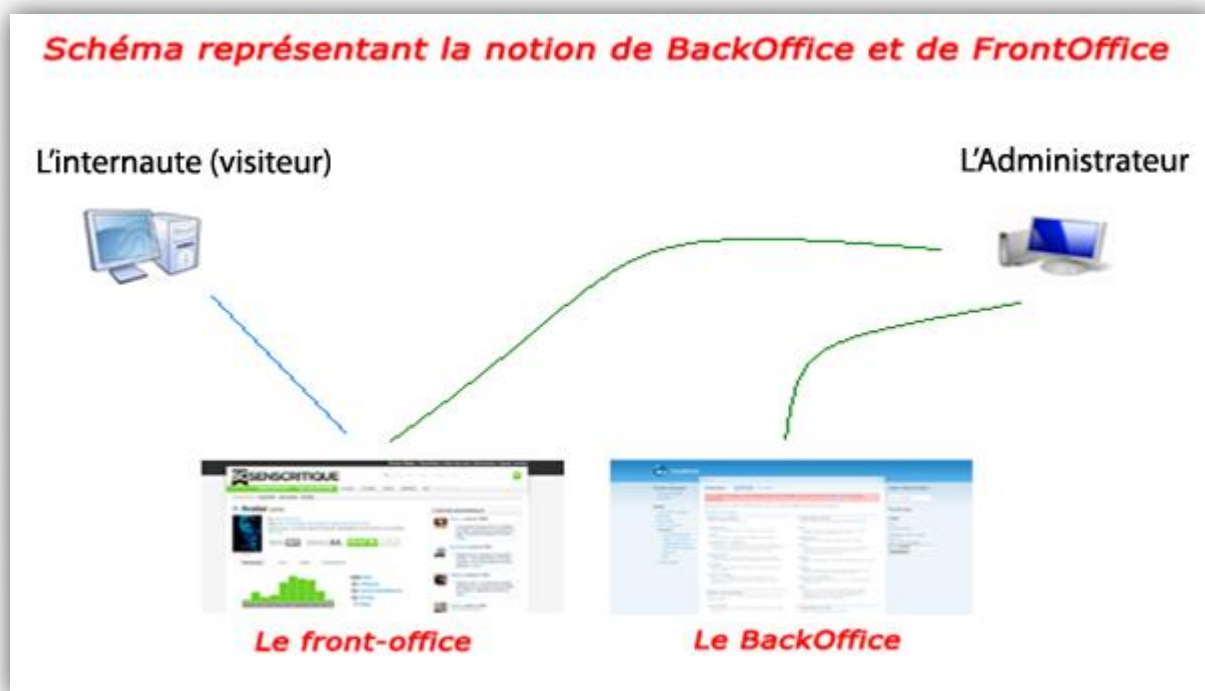
Ce contrôle nécessitait de posséder de fortes connaissances des langages de programmation.

Ainsi les clients rappelaient les développeurs pour effectuer une multitude de modifications mineures (exemple : ajouter un nouveau lien dans le menu, changer légèrement le texte de la page d'accueil, etc.) ceci donnait lieu à des contrats de maintenance.

Le CMS est l'outil idéal pour les sites communautaires, associatifs, ou liés à l'actualité. Un CMS permet de créer des sites Web dynamiques (c'est-à-dire le contraire des sites "statiques", où le contenu est figé, et ne varie pas avec le temps).

Grâce au temps de réalisation moins long, les cms changent la vie des entreprises et, par conséquent, celle des clients également. Cela leur permet d'être autonomes, c'est le compromis idéal.

Le prestataire n'est pas perdant dans sa facturation, certes il ne fait plus de contrat de maintenance car le client devient autonome avec le cms mais il ne vend plus un seul site mais deux : Le front et le back. Il s'agit d'un site « deux en un » !





Notion : FrontOffice / BackOffice

A QUOI SERT LA SEPARATION BACK / FRONT ?

En informatique, le back office est une partie d'un site web ou d'un système informatique.

Elle concerne la partie qui permet à l'organisation en charge d'un système informatique de l'administrer et de le gérer. On pourra retrouver comme service par exemple :

- Ajout des produits et des services
- Modification des services
- Modifications de paramètres
- Administration d'un forum
- Gestion des utilisateurs
- Création et modification des pages
- Gestion du référencement naturel – SEO

Le Back office (BO) est opposé au Front Office qui est la partie servant à l'interaction avec l'utilisateur. Avec peu de connaissances en développement, un Webmaster peut mettre à jour un site web via le back office en toute simplicité et ainsi publier de nouveaux contenus instantanément.

ACCES

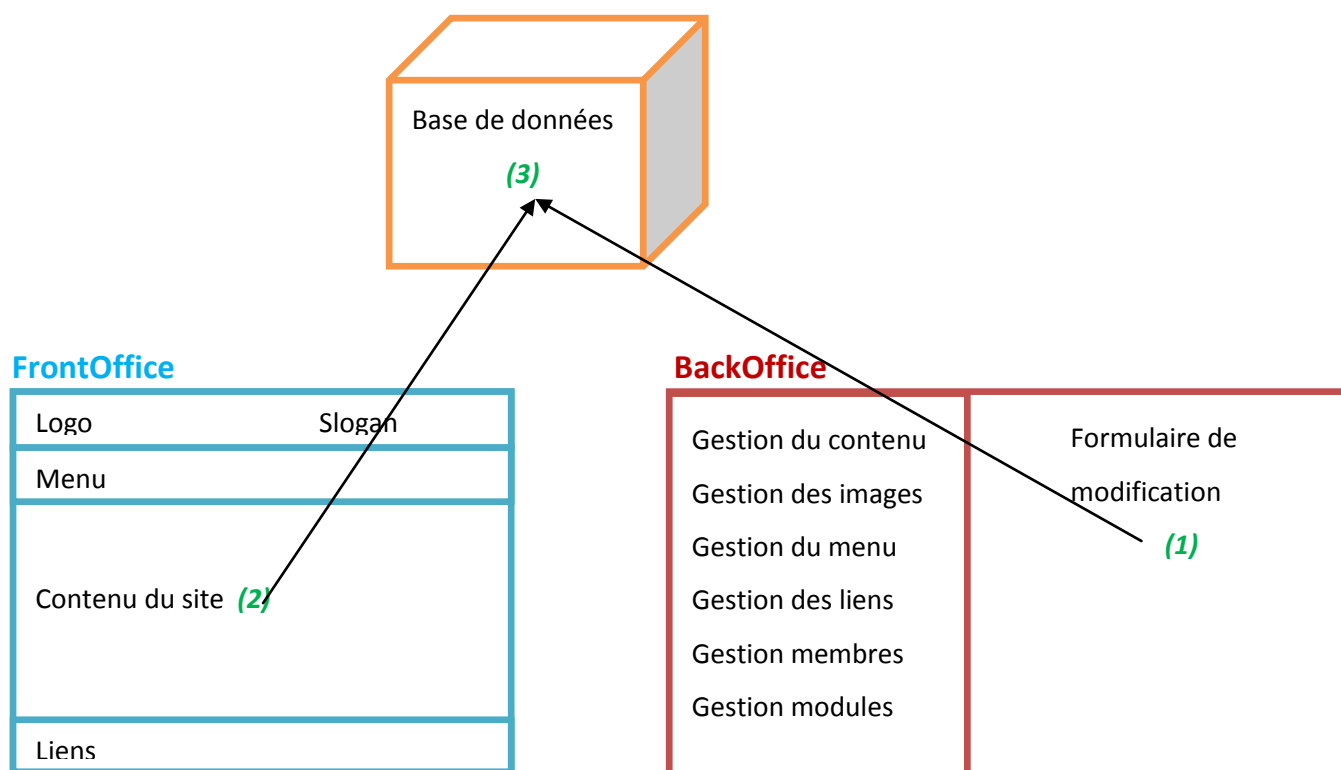
La partie du BackOffice étant réservée à l'administrateur, l'accès se fait via une adresse « secrète » et avec authentification requise.

Bien qu'il soit tout à fait possible de mettre en place un FrontOffice et un BackOffice sans CMS, cette séparation est généralement présente dès l'installation du CMS.

Elle est réservée au propriétaire du site ainsi qu'à son équipe d'administration.

TECHNIQUEMENT, COMMENT CELA EST-CE POSSIBLE ?

Nous ne pouvons pas faire une modification sur la partie BackOffice qui impacte directement le FrontOffice car il s'agit d'affichages différents, c'est la raison pour laquelle nous avons besoin d'une base de données.



- (1) Une modification se fait dans la partie BackOffice et est enregistrée en base de données.
- (2) Le Front génère son affichage en fonction des informations qu'il trouve en base de données.
- (3) Le Back et le Front ne sont pas liés directement et passent naturellement par la base de données.

FORMATION SUR L'OUTIL

Souvent, c'est au développeur du site d'effectuer une formation à son client sur la partie BackOffice afin de lui faire une démonstration des différentes possibilités qui s'offrent à lui.

De cette manière, le client peut être autonome dans la mise à jour de son site Internet.

Le client peut effectuer des mises à jour lorsqu'il en a besoin et ceci ne requiert aucune connaissance en programmation.

Principales caractéristiques des CMS

SEPARATION ENTRE CONTENU ET PRESENTATION

C'est un principe fondateur de la gestion de contenu :

Le contenu est stocké le plus souvent dans une base de données, structurée en tables et en champs.

C'est le contenu des champs de la base qui est créé/modifié par le rédacteur, et non pas la page elle-même. On parle de site « dynamique ».

Le stockage central du contenu permet de réutiliser ces informations à différents endroits du site et de les formater différemment.

DISSOCIATION DU DESIGN DU SITE

Dissociation du design du site (le graphisme, la mise en page) et de son contenu. On peut ainsi modifier le design, à tout moment, sans avoir à extraire le contenu existant et à le replacer dans le nouveau design. Cela permet également de fournir une cohérence graphique au site entier. Il est également possible à chaque utilisateur d'avoir son environnement personnalisé.

La présentation est définie dans un gabarit. Le gabarit définit deux choses : la mise en page proprement dite - via les feuilles de style (parmi lesquelles les CSS, ou les règles de transformation XSLT), et la structuration des données, au moyen de standards tels que XML, ainsi que les informations extraites de la base de données (de même que l'endroit où celles-ci doivent être affichées et sous quelles conditions).

ÉDITION DE PAGE SIMPLIFIÉE

Du fait du principe de séparation contenu/présentation, les rédacteurs peuvent se concentrer sur le contenu.

L'édition des pages est considérablement simplifiée grâce à une barre d'outil développée WYSIWYG qui propose une interface graphique de mise en forme similaire à celle qui existe dans un traitement de texte.

SEPARATION BACKOFFICE / FRONTOFFICE

Un CMS comporte une interface "graphique" d'administration, intuitive, qui permet une gestion aisée et rapide du contenu. Le FrontOffice devient la partie présentée au client, il faut bien dissocier ces deux interfaces et bien comprendre l'intérêt de cette notion avant d'aller plus loin.

DEFINITION DE DROITS D'ACCES DES UTILISATEURS

Des utilisateurs peuvent être définis et on peut leur octroyer des rôles et différentes permissions d'accès au contenu ou à l'administration du site.

Plusieurs personnes peuvent être autorisées à gérer le site, ou une partie de celui-ci.

AVANTAGES LIES A L'UTILISATION D'UN CMS

Voici quelques avantages liés à l'utilisation d'un CMS :

- Ces logiciels présentent de nombreux avantages dont le premier est qu'il n'est pas nécessaire de connaître les différents langages de programmation pour les mettre en œuvre.
- Ces logiciels sont gratuits et open source pour la plupart, c'est-à-dire que l'accès au code source du logiciel est autorisé par ses auteurs, donc qu'ils sont libres d'utilisation et de modifications.
- Ensuite, il est possible de réaliser des sites Internet dynamiques, c'est-à-dire dont le contenu est entièrement modifiable à travers une interface de l'administration (par opposition aux sites Internet statiques dont le contenu est figé).
- Possibilité de travail simultané à plusieurs sur la même plateforme et participation valorisante des collaborateurs au contenu éditorial lié aux activités de l'entreprise



➤ Faut-il systématiquement utiliser un CMS ?

LES POINTS INTERESSANT POUR LE DEVELOPPEUR

Un code optimisé et fiable, c'est ce que tout bon développeur recherche, parfois 90%, voire 100% de ce qu'un client vous demande est déjà disponible sous forme de « module / extensions » via un CMS. Alors si on vous le propose tout prêt, pourquoi hésiter ?

Gain de temps

Pour le développeur, l'évaluation du temps gagné par l'utilisation d'un CMS sur un projet est souvent déterminante dans son choix. Ce serait donc une perte de temps évidente de chercher à réinventer ce que d'autres ont développé. Pourquoi réinventer la roue ? D'autant que bien souvent les projets communautaires sont d'une grande qualité et la somme des compétences techniques de leurs auteurs est bien plus grande que la vôtre ou celle de votre équipe. En utilisant une solution suivie par plusieurs développeurs dans le monde vous gagnerez en temps et même sur la maintenance, les évolutions et corrections de bugs de la plateforme ne seront pas de votre ressort.

Il arrive parfois que le client ait réalisé une étude ou ait déjà travaillé avec un CMS en particulier et veuille absolument une plateforme pour son projet, alors il demande conseil ou parfois impose son choix au développeur.

LES POINTS POUVANT BLOQUER LE DEVELOPPEUR

Quand un CMS apporte 80% du besoin du client, c'est un gain de temps considérable et il est intéressant de l'utiliser, en revanche il est capital de déterminer combien de temps prendront les 20% restant ? Il convient de se poser les bonnes questions :

- Le temps pour développer les 20% restants est-il supérieur au temps pour développer les 100% à la main ?
- Il est plus facile de modifier un code écrit par soi-même que le code de quelqu'un d'autre alors quel est le niveau de difficulté du code présent dans le module ? est-il facilement modifiable ?

Le développement doit se faire en interaction avec le CMS utilisé, s'il ne l'est pas, le développeur obtiendra très vite les 80% du site mais risque de ne jamais terminer le site à 100% et ne pourra pas le livrer à son client.

Parfois le CMS fait gagner du temps au développeur, parfois il lui en fait perdre.

D'autres points bloquants peuvent intervenir dans la partie prévue par le CMS, par exemple :

Lorsqu'un client souhaite obtenir un espace membre, le développeur active le module « membres » du CMS, si le client souhaite obtenir un espace membre se faisant en deux étapes (deux pages) et que le module mis en place par le développeur ne prévoit pas cette configuration, le développeur devra modifier le code du module (qui n'est pas prévu à cet effet) tout en veillant à ne pas faire

dysfonctionner le reste. Ce qui est différent que de concevoir l'espace membre de A à Z en prenant directement en compte ce souhait.

Même si les CMS permettent de mettre rapidement des fonctionnalités en place, il faut des connaissances avancées en programmation au risque de ne jamais pouvoir terminer un site.

L'ATTIRANCE DES CLIENTS VERS LES CMS

Pour le client, avoir une plateforme connue et reconnue sur le marché lui donne confiance en l'outil. Avoir une plateforme ayant un BackOffice offrant de riches possibilités fait souvent la réputation du CMS et le choix du client pour les raisons suivantes :

- Diminuer les coûts liés à la mise à jour du site Internet.
- Ne pas être obligé de recontacter l'entreprise de développement pour publier de nouveaux contenus ou pour déplacer certains éléments sur son site Web.
- Parce qu'il permet de décentraliser les sources d'informations en permettant à plusieurs personnes de faire la mise à jour du site Internet. Les personnes autorisées pourront effectuer des mises à jour et ainsi travailler simultanément sur des sections différentes du site Internet.
- Le CMS permet de gérer dynamiquement la publication du contenu d'un site Internet comme le texte, fiches produits, visuels, photographies, vidéo ou diaporama afin d'avoir un site évolutif au niveau de l'information et améliorer la communication avec sa clientèle cible.
- Dans un CMS, les contenus (publication) sont séparés du thème graphique, cela constitue un énorme avantage pour le référencement et l'indexation du site Web par les moteurs de recherche comme Google.
- De plus, étant donné que la programmation du futur site Web est déjà réalisée, le temps de développement est réduit et le site revient moins cher au client sur le long terme.

En conclusion, les CMS apportent un gain de temps non négligeable, des interfaces d'administration soignées et des budgets de développement raisonnables.

QUELLE EST LA DEMARCHE QUI CONVIENDRA LE PLUS AU CLIENT ?

Si un client demande une plate-forme e-commerce avec beaucoup de fonctions présentes dans le CMS Magento, le développeur a le choix entre deux solutions :

- tout réécrire, par exemple à la main ou avec un framework.
- Prendre un CMS tel Magento, et y rajouter les modules dont votre client a besoin.

Pour répondre à cette question il est important de savoir de quel type de clients nous parlons (*voir chapitre suivant*).

Dans le cas où le code est écrit à la main sans CMS, il est important que le développeur documente son code de manière à ce qu'un autre développeur puisse reprendre le projet plus tard et ne pas

contraindre le client à être obligatoirement dépendant du développeur d'origine pour l'évolution du projet.

DE QUELS CLIENTS PARLONS-NOUS ?

S'il s'agit d'une PME qui démarre et dont la force sont ses produits... elle a besoin d'un outil et le package open-source, complété par le développeur, est sans doute la meilleure solution pour le client.

Si au contraire, il s'agit d'une entreprise plus grande qui souhaite faire évoluer son site sans cesse et qu'une partie très importante de son succès est due à sa créativité e-commerce, pas question d'utiliser l'open-source dont, juridiquement, les développements risquent d'être eux aussi open-source, il ne reste donc que la solution du sur-mesure.

A QUELS BESOINS REPONDENT LES CMS ?

Un client venant de lancer son activité, une TPE ayant besoin d'une présence sur Internet et par conséquent avec une certaine flexibilité sur la précision des fonctionnalités et avec des besoins basics, sera ravi d'avoir un site basé sur un CMS parce qu'il permet de fournir un site rapidement administrable et possédant de la modularité fonctionnelle (en particulier concernant les gabarits) ce qui rend le site différenciable de celui de la concurrence.

Peu de développements auront été réalisés par le développeur et cela n'empêchera pas le site d'être esthétique avec une partie d'administration à la clé.

Certains clients sont plus exigeants et ont besoin d'un site sur mesure dont les fonctionnements attendus ne sont pas forcément gérés par la communauté d'un CMS, il faut voir si le CMS permet de créer les fonctionnements attendus par le client mais il est souvent bien plus rentable d'utiliser un Framework pour sa "modularité technique".

Les solutions sur mesure développées à l'aide d'un framework sont souvent plus chères que des solutions basées sur des CMS.

En conclusion, le choix d'une solution CMS ou autres est stratégique pour le développeur comme pour le client, un certain nombre de paramètres doit être pris en compte et cette décision doit être mûrement réfléchie.



Questions stratégiques

LES LIMITES DES CMS

Aux programmes des CMS eux-mêmes, viennent s'ajouter des modules, (ou des extensions), qui permettent d'ajouter des fonctionnalités au site.

Certaines communautés de CMS proposent des modules gratuits et d'autres payants... Tout le monde peut créer son module et ainsi apporter sa contribution à la communauté.

Ces solutions sont-elles toujours souhaitables et durables ? Comportent-elles des limites ou des risques ?

Les limites d'un CMS sont souvent liées au niveau de compétence en programmation d'un développeur ainsi qu'à sa capacité à se servir de ses connaissances en homogénéité avec le noyau du système qu'il utilise.

COMMENT FAIRE SON CHOIX ?

Beaucoup de CMS sont sur le marché (peut être trop ?), et avec les variations de popularité il n'est pas toujours évident de choisir LE CMS de la situation qui restera sûr dans l'avenir.

Il est évident qu'une société ne peut pas vraiment se permettre de changer de CMS tous les 3 mois, il est donc important qu'elle choisisse bien le CMS qui lui convient en fonction des missions qu'elle doit accomplir.

En entreprise, il est important d'être force de proposition et d'apporter des réponses argumentées. Alors voici quelques points sur lesquelles le développeur et le client doivent être vigilant.

1. Un CMS populaire faisant parler de lui n'étant pas en perte de vitesse, apprécié par des professionnels, récompensé aux concours, ayant des références et utilisé par beaucoup de monde.
2. Une gestion complète via un panel d'administration simple et intuitif.
3. Le CMS doit être modulaire, il doit posséder un grand choix d'extensions/modules simples et rapides à mettre en place. Les modules d'un CMS font souvent sa richesse car cela permet d'effectuer moins de développement mais il ne faut pas oublier de regarder si les modules sont gratuits ou payants.
4. Nombre de ressources, l'accès à l'information se fait-elle rapidement ? (Recherche à effectuer dans plusieurs langues)
5. Une communauté réactive et avenante pour aider et répondre en cas de problème.

QUELS SONT LES RISQUES ?

Les risques de voir « couler » un CMS leader sur le marché sont faibles.

Ce qui pourrait en être les causes serait :

- des problèmes de sécurité fréquents, sans mise à jour proposée par les propriétaires du CMS
- la communauté qui s'affaiblit et par conséquent moins d'informations pour les apprentis qui souhaiteraient se former.





Open Source

QU'EST-CE QUE C'EST ?

On dit d'un logiciel qu'il est "Open Source" dès lors qu'il est fourni avec ses sources complètes et que la licence permet à tous, sans restrictions et sans participer aux frais de développement, d'ajouter et de modifier le contenu du logiciel (il existe une multitude de licences la plus connue est GNU).

La nétiquette voudrait que chaque utilisateur de produit Open Source participe ainsi à l'évolution du produit, et ce, gratuitement.

Nous constatons que malheureusement les produits Open Source sont beaucoup plus souvent utilisés à sens unique.

C'est-à-dire que l'utilisateur final ne signale pas toujours les efforts fournis par l'équipe qui a créé ce projet en plaçant un lien retour, vers le site de téléchargement du logiciel Open Source par exemple.

L'Internet tel que nous le connaissons s'est constitué autour de logiciels Open Source et libre.

À titre d'exemples : Linux - Apache - PHP - MySQL rendent possible l'existence de la grande partie des sites présents sur le Web.

Ces quatre produits sont et n'existent que par l'Open Source (d'importantes entreprises se cachent derrière ces quatre produits Open Source).

Contraire d'Open Source

Logiciel propriétaire, logiciel fermé ...

AVANTAGES / INCONVENIENTS DE L'OPEN SOURCE

Avantages

- Une communauté conséquente de développeur.
- Les forums proposant un support gratuit.
- Le temps de développement grandement réduit (il s'agit plus souvent d'adaptation que de création).
- Une image positive que vous pourrez véhiculer.
- Des produits qui peuvent être testés gratuitement.
- Sécurité pour le client d'avoir un outil connu par le monde des professionnels (peu de crainte que le CMS devienne obsolète).
- Une conception du développement qui mêle des passionnés et des fonctionnalités utiles.
- Et surtout la facilité de reprendre par vous-même la solution que l'on vous aura fournie.

Inconvénients

- Induite par l'accès au code source, la transparence est considérée comme une faiblesse en matière de sécurité.



Licence propriétaire

UN BON CHOIX ?

Il n'est pas judicieux pour le client d'effectuer un site basé sur un CMS sous licence d'une entreprise de développement car bien souvent ce ne sont que quelques-uns des employés qui sont à l'origine de la création de celui-ci et détiennent, seuls, les connaissances avancées sur le système qu'ils ont créé. Pour toutes les évolutions futures cela condamne le client à travailler exclusivement avec l'entreprise qu'il aura choisie lors de la création du site car ils sont à l'origine de la création du CMS et qui mieux que les concepteurs peuvent le maintenir ? Ce n'est pas une fin en soit mais cela peut profiter à l'entreprise de développement qui peut imposer des tarifs plus élevés sans marge de négociation pour le client car ils seront obligatoirement en position de force en étant les seuls à connaître leur solution.

Si pour une raison quelconque, les employés à l'origine de la création du CMS venaient à quitter l'entreprise, cela en menacerait son utilisation car une partie des connaissances partirait avec eux.

Si l'on suit cette première menace, cela constitue un risque pour les clients ayant décidé de faire un site appuyé sur cette solution. Imaginons qu'un client veuille mettre à jour son site Web et développer de nouvelles fonctionnalités, si la société par laquelle il est passé n'est plus capable d'assurer la maintenance et que par conséquent leur système de gestion de contenu devient obsolète, le client devra contacter une autre entreprise, qui elle, ne se plongera pas dans la doc (si elle existe) de ce CMS mais proposera certainement de recréer complètement le site avec un CMS open source. Si le site était de grande envergure, la facture sera élevée !

La conception d'un système de gestion de contenu ne se fait pas en quelques jours, Il y en a suffisamment sur le marché (peut-être trop ?) pour en choisir un qui répond aux besoins d'un client et l'adopter !

AVANTAGES / INCONVENIENTS DES LICENCES PROPRIETAIRES

Avantages

Une solution maîtrisée sur le bout des doigts et peu de risques de contrainte technique car les concepteurs sont les développeurs eux-mêmes.

Inconvénients

L'entreprise de développement connaît sa solution et peut faire des mises à jour pour le client mais cela oblige le client à rester dépendant de l'entreprise prestataire sans pouvoir se tourner vers d'autres développeurs en cas de besoin.

La durée de vie de la solution sous licence propriétaire peut être menacée si certains employés choisissent de quitter l'entreprise de développement.

Pour l'employé qui l'utilise que ce soit en terme de développement, d'intégration ou d'administration cela ne lui permettra pas de mettre une référence connue sur son CV, c'est pourtant un atout de taille lors d'entretiens d'embauche sans oublier qu'il ne lui sera pas possible de mettre en ligne des sites Internet « personnel » sur la solution qu'il connaît.

WordPress

DESCRIPTION

WordPress est un système de gestion de contenu libre écrit en PHP et reposant sur une base de données MySQL. WordPress est surtout utilisé comme moteur de blog, mais ses fonctionnalités lui permettent également de gérer n'importe quel site web. Il est distribué selon les termes de la GNU GPL. Le logiciel est aussi à l'origine du service WordPress.com.

HISTOIRE

WordPress a vu le jour en 2003 et est reconnu pour la simplicité de sa prise en main. Développé principalement par Matthew Mullenweg, dont une amie eut l'inspiration du nom, WordPress a connu un succès grandissant auprès des utilisateurs par ses nombreuses fonctionnalités, le dynamisme de sa communauté et la possibilité de développer des plugins.

RECOMPENSES

Dans le concours du meilleur CMS Open source 2009 organisé par la société britannique Packt, WordPress est arrivé 1er dans la catégorie Overall Winner et 2e dans la catégorie Best Open Source PHP Content Management System, précédé par Drupal.

INSTALLATION

Pour une installation en local sous WAMP, il vous faut télécharger puis décompresser l'archive WordPress.

Pour une installation à la racine du répertoire /www/, il vous faudra ouvrir le navigateur à l'adresse suivante :

<http://localhost/wordpress/>

Je ne trouve pas votre fichier `wp-config.php`. J'en ai besoin avant de lancer l'installation.

Besoin d'aide ? [En voici.](#)

Vous pouvez créer un fichier `wp-config.php` à l'aide de notre interface Web, mais ça ne marche pas pour toutes les configurations de serveur. La méthode la plus sûre reste de créer le fichier à la main.

Créer le fichier de configuration

Wordpress demande de créer le fichier de configuration

Si cela n'est pas possible, il existe un fichier répondant au nom de « wp-config-sample.php ».

Renommez-le en « wp-config.php », puis éditez-le en l'ouvrant avec le bloc-notes.

```
define('DB_NAME', 'wordpress') ; // le nom de la base de données
```

```
define('DB_USER', 'root') ; // votre identifiant MySQL
```

```
define('DB_PASSWORD', '') ; // ...et votre mot de passe
```

Ensuite il suffit de suivre les étapes :



Bienvenue dans WordPress. Avant de nous lancer, nous avons besoin de quelques informations à propos de la base de données. Il vous faudra connaître les points suivants pour aller plus loin.

1. Le nom de la base de données ;
2. Votre identifiant de base de données ;
3. Votre mot de passe de base de données ;
4. L'hôte de la base de données ;
5. Le préfixe de table (si vous voulez installer plus d'un blog WordPress sur la même base de données).

Si pour une raison quelconque la création automatique du fichier ne fonctionnait pas, ne paniquez pas : elle ne fait que remplir les informations de la base de données dans un fichier de configuration. Vous pouvez donc simplement ouvrir `wp-config-sample.php` dans un éditeur de texte, y entrer vos informations, et enregistrer le fichier sous le nouveau nom `wp-config.php`.

Normalement, ces informations vous ont été fournies par votre hébergeur. Si vous ne les avez pas, il vous faudra le contacter pour continuer. Si vous êtes prêt...

Allons-y !

A ce moment précis, il vous faut renseigner le nom de votre base de données (créé au préalable via PhpMyAdmin par exemple) ainsi que vos identifiants (si vous n'avez rien changé lors de l'installation de WAMP, les identifiants sont « root » pour le pseudo et un mot de passe vide).



The screenshot shows the WordPress installation database configuration screen. At the top is the WordPress logo. Below it, a text box instructs the user to enter database connection details. There are five input fields, each with a label on the left and a description on the right:

- Nom de la base de données**: Input field contains 'wordpress'. Description: 'Le nom de la base dans laquelle vous voulez installer WP.'
- Identifiant**: Input field contains 'username'. Description: 'Votre identifiant MySQL.'
- Mot de passe**: Input field contains 'password'. Description: '...et votre mot de passe MySQL.'
- Hôte de la base de données**: Input field contains 'localhost'. Description: 'Si localhost ne marche pas, vous devrez demander cette information à votre hébergeur.'
- Préfixe de table**: Input field contains 'wp_'. Description: 'Si vous voulez installer plusieurs blogs WordPress dans une même base de données, modifiez ce champ.'

At the bottom left, there is a 'Valider' button.

La dernière étape vous permet de valider complètement l'installation et vous pourrez accéder à votre BackOffice.

L'url d'accès pour votre BackOffice est la suivante :

<http://localhost/wordpress/wp-admin/>

PERSONNALISER SON THEME

Il s'agit de l'habillage, de la mise en forme.

Il est possible de créer un thème facilement afin de rendre son blog unique :

Une fois positionné sur votre serveur, déplacez-vous dans le dossier `/wp-content/themes/`,

Une manière très simple consiste à copier/coller le thème par défaut et à éditer les commentaires ainsi que le code de la feuille de style css pour avoir l'affichage correspondant.

Il ne faut pas oublier de se rendre dans l'interface d'administration et d'activer votre nouveau thème.

Il est également possible d'en télécharger via le BackOffice ou à l'adresse suivante :

<http://wordpress.org/extend/themes/>

Il est préférable de cocher « Widget ready ». Cela vous permettra d'avoir un thème moderne, compatible avec les widgets.

LES PUBLICATIONS

Les publications créées via le BackOffice permettent de publier rapidement du contenu sur le coté FrontOffice du site Internet.

L'Article n'est pas statique...

Contrairement à une Page. Il s'affiche dans différents endroits, en mode réduit ou développé. On les fait défiler sur l'écran du haut vers le bas. Si on clique sur le titre d'un article, on l'affiche en pleine page.

Les articles (appelés aussi billets ou posts en anglais) sont la base d'un blog. Avec leurs date & heure, leurs commentaires, leurs tags... Ils permettent de publier des actualités, des humeurs, des informations... Ils sont donc classés par date. En général, l'article le plus récent est en haut de l'écran.

...alors que la Page est statique.

Il remplit en général tout l'écran. Cela peut servir pour une présentation, un CV, une page contact, un album photo, ou l'accueil du blog.

Tout cela peut être naturellement organisé par des catégories ainsi que des rubriques dans le menu.

LES PLUGINS

Les plugins vous permettent d'ajouter des fonctionnalités sur votre site, il vous est possible d'en créer un ou d'en télécharger directement via le BackOffice ou à l'adresse suivante :

<http://wordpress.org/extend/plugins/>

Les plugins se mettent sur le serveur dans le dossier `/wp-content/plugins/`. Une fois téléchargés, rendez-vous sur votre interface d'administration sur l'onglet « Plugins ».

LES WIDGETS

Les widgets de Wordpress sont des petites sections d'informations constituées d'un titre et d'un "texte" à placer dans des zones du site.

Ils vous est possible d'en créer, généralement ils sont fournis grâce aux plugins.



DESCRIPTION

Drupal est un système de gestion de contenu (CMS) libre et open-source publié sous la licence publique générale GNU, et écrit en PHP. Dries Buytaert, développeur initial du projet à partir de 2000 à l'université d'Anvers, le définit comme « assembleur rapide de site web » (Rapid website assembler). Il est utilisable tel que fourni - sur toute base LAMP, WAMP ou MAMP - mais largement personnalisable et programmable ensuite. D'après son créateur, environ 500 000 sites l'utilisent en septembre 2009.

Drupal est un gestionnaire de contenus (CMS) qui permet de créer une grande variété de sites Internet : blogs, portails, corporatifs, e-commerces, communautaires, réseaux sociaux, wiki, forums... Il est distribué sous licence open-source GPL. Il est donc gratuit et est développé par des milliers de développeurs expérimentés.




HISTOIRE

La première version de Drupal à vu le jour en janvier 2001.

RECOMPENSES

En 2009, il est classé 1er du concours organisé par la société britannique Packt dans la catégorie Best Open Source PHP CMS suivi par WordPress et Joomla!. Il fait également son entrée en première position dans le Hall of Fame Award nouvellement créé (suivi par Joomla!).

En 2010, Angela Byron annonce que Drupal équipe à peu près 1% des sites Internet mondiaux.

 <p>Premier au concours du meilleur CMS (Récompense du meilleur système de gestion de contenu en accès libre).</p>	 <p>Second dans la catégorie (Meilleur système de gestion de contenu en accès libre et en PHP).</p>	 <p>Second au (Meilleur système de gestion de contenu en accès libre orienté réseaux sociaux, derrière WordPress).</p>
--	---	--

REFERENCES

Sites institutionnels

- Le portail du gouvernement français : gouvernement.fr
- Le site de la Maison Blanche : whitehouse.gov

Presse

- Madiapart : mediapart.fr
- Rue 89 : rue89.com
- France 24 (France Télévisions) : france24.com
- France Télévisions SPORTS : sport.francetv.fr
- Courrier Cadres : courriercadres.com

Divers

- Centre de formation et de perfectionnement des journalistes (CFPJ) : cfpj.com
- Amnesty International : amnesty.org
- International Justice Mission (IJM) : ijmuk.org
- Unicef : unicef.fr

LE MODULE NODE

(1 nœud = 1 contenu)

Ce module essentiel permet de créer de nouvelles pages (articles ou autres...) dans le site et de les afficher. Ceci est bien l'essentiel de ce que nous voulons faire quand nous créons un site.

Drupal, comme tout contenu qu'il gère, est un "nœud". Ce nœud possède un type : forum, article de fond, information, brève, tutoriel, blog, commentaire, formulaire de saisie, livre collaboratif, etc. La forme n'est plus assujettie à une architecture prédéterminée, ce qui rend le contenu aisément

« reconfigurable ». Contrepartie de cette liberté : on doit se familiariser avec sa logique particulière.

Chaque nœud peut être attaché simultanément à plusieurs taxinomies distinctes si on le désire (ainsi une brève sur une médaille d'or française aux jeux Olympiques peut être rattachée à la fois à "Sport" et à "France"). Le concepteur ou l'administrateur ne sont donc pas obligés d'insérer leurs contenus dans une hiérarchie initiale unique.

À sa création, un nœud se voit attribuer un node ID (NID) qui le caractérise. Au fur et à mesure de ses révisions, s'il y en a, le système incrémentera un compteur de révision ID (RID). Chaque nœud possède aussi un titre, ce qui permet aux administrateurs de les gérer de façon plus commode.

Chaque contenu sur le site est donc un nœud (node en anglais). La page d'accueil l'est également, dans la création d'un nœud, nous pouvons cocher « promu sur la page d'accueil » pour définir celle-ci.

LES TEMPLATES

Par défaut, la page d'accueil s'intègre dans le thème défini pour le BackOffice de Drupal. Pour personnaliser son affichage, il suffit de créer un fichier de template dans le dossier qui contient le thème du site. Pour la page d'accueil, ce sera page-front.tpl.php, pour les autres cela sera page-node-1.tpl.php, page-node-2.tpl.php, etc. Il est également possible de créer des fichiers de template pour un groupe de nœuds ou pour des types de contenus différents et bien spécifiques.

Le fichier template page.tpl.php permet de coder l'apparence du site (le design). Le fichier template node.tpl.php permet d'afficher et de personnaliser l'affichage du nœud lui-même.

Ni les nœuds ni les modules ne s'occupent de la présentation. Ce sont les styles qui en sont chargés, à la manière des feuilles de style en (X)HTML. Un administrateur de site Drupal peut en changer profondément le style juste en quelques clics de souris.

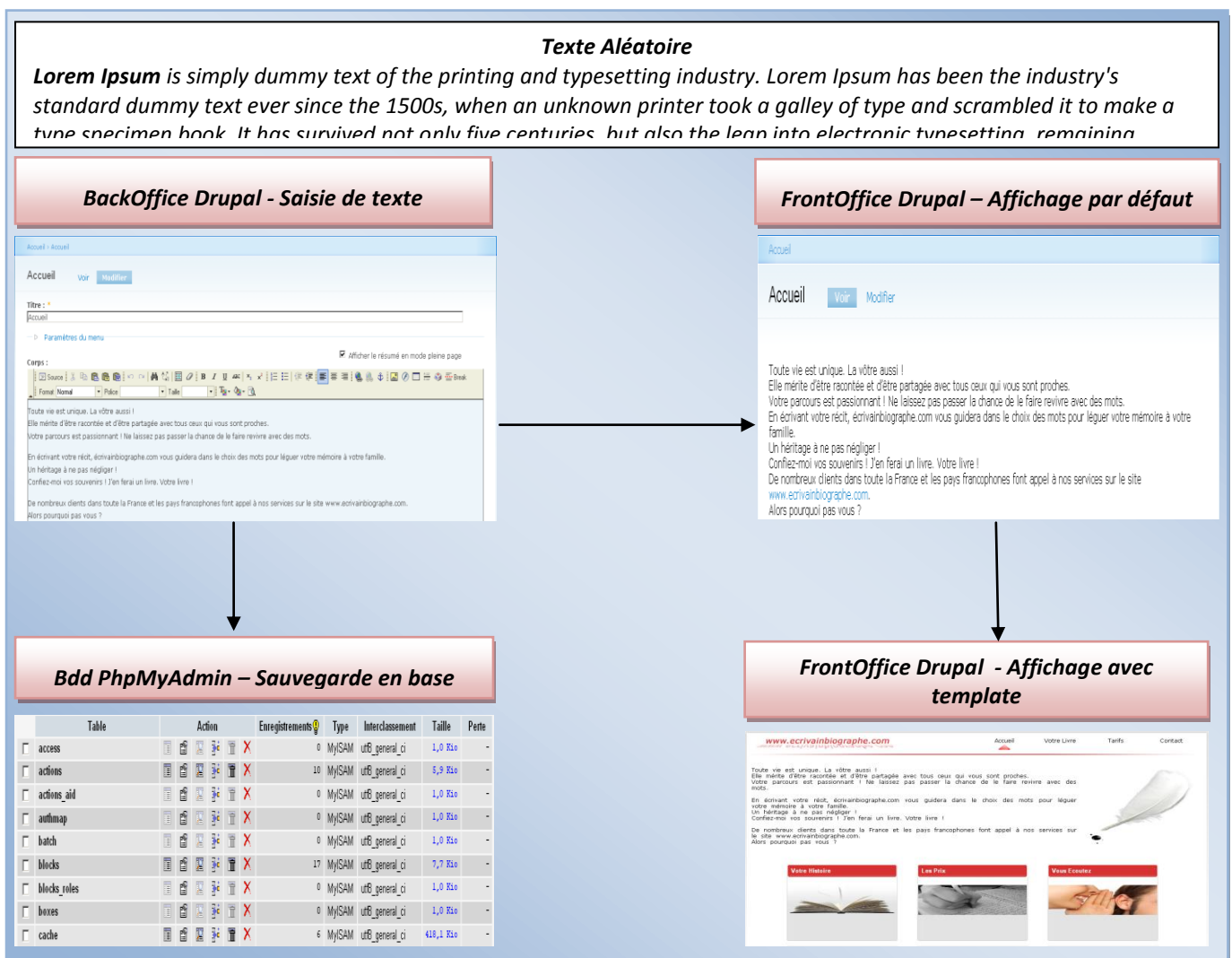
LE MODULE PATHAUTO

Il permet de renommer des url tels que www.monsite.com/node/6 en www.monsite.com/article/le-vote-des-regionnales

Après avoir donné un alias d'url, nous perdons le nid (identifiant du nœud) dans l'url, mais nous pouvons le connaître en 1 ligne de code : `<?php print $node->nid ; ?>`. Il est plus judicieux de se servir de pathauto pour renommer les urls si l'on souhaite référencer notre site et l'indexer sur les moteurs de recherches.

SCHEMA

Schéma représentant la création d'un contenu (un nœud, notion abordée juste au-dessus), avec l'affichage par défaut, l'enregistrement en base et l'affichage de ce même nœud via une page de template.



LE MODULE BLOCK

Les blocs Drupal sont des petites sections d'informations constituées d'un titre et d'un "texte". Ils peuvent être placés dans des zones dites « régions » sur le site Internet. Nous pouvons ajouter des régions à volonté dans le fichier .info du dossier conteneur du thème. Dans ce fichier .info, nous pouvons également déclarer d'autres feuilles de style ou des scripts JS à prendre en compte au chargement des pages. Ce qui nous permettra plus tard de styliser à notre guise les blocs, les nœuds et autres...

LE MODULE VIEWS

Il permet d'afficher le contenu d'une page (en liste, en tableau, en résumé). Il permet de définir dans quel ordre afficher le contenu d'une page (alphabétique, chronologique, selon le choix de l'utilisateur...). Et aussi quels types de contenus (nodes) sont à afficher sur une même page. L'interface des vues permet de faire ses propres choix, on peut y ajouter des arguments et des filtres ; Views génère les requêtes sql en conséquence.

LE MODULE CCK

Le Content Construction Kit sert à créer des champs personnalisés à l'intérieur de la création de nœud. Un nœud sera composé par défaut d'un titre et d'un corps pour du texte, à l'aide de cck, nous pourrions ajouter par exemple un champ pour uploader une image dans le nœud ou ajouter un champ « node référence » de manière à faire communiquer les nœuds entre eux.

LES ROLES

L'administrateur peut affecter à chaque utilisateur un ou plusieurs rôles, qui regroupent un ensemble de permissions. Il est alors possible de définir finement autant de permissions que nécessaire entre l'administrateur - qui peut tout faire - et l'utilisateur non enregistré, qui peut par exemple n'avoir le droit que de regarder le site sans le modifier.

Un rôle peut être défini pour le service commercial, un autre pour le service communication. Ils n'auront pas le droit d'interagir aux mêmes endroits.

POURQUOI CREER SON MODULE

Une des grandes forces de Drupal est qu'il est très modulaire.

Cela veut dire que, d'une part, il est relativement facile pour un développeur de créer un nouveau module et de le maintenir, car il n'a pas à modifier le code de base du tout.

D'autre part, l'administrateur n'a besoin que d'installer les modules dont il a besoin, sans alourdir son site de modules inutiles. Le système est ainsi très flexible. Cependant, quand nous créons des sites de grande envergure, il est probable que l'on soit obligés de faire quelque chose de spécifique à la demande du client, à tel point qu'il est possible que cela n'est jamais été réalisé par la communauté Drupal. Dans ce cas, il est utile de créer son propre module.

