

Introduction aux concepts d'eZ Publish

Tutoriel rédigé par [Bergfrid Skaara](#).

Traduit de l'Anglais par [Benjamin Lemoine](#)

Mercredi 30 Janvier 2008

Sommaire

Concepts d’eZ Publish	3
Système de Gestion de Contenu	3
Séparation du contenu et du design	3
Contenu	4
Design	4
Les composants d’une page Web	4
Siteaccess	5
Interfaces.....	5
Siteaccess	5
L’Interface d’Edition	5
L’Interface d’Administration	6
Le système de Template	6
Un système axé sur les Composants	6
Les tâches des Templates.....	7
Un modèle de contenu orienté objet	7
Qu’est-ce que l’orientation objet ?.....	7
Content class	7
Content object.....	8
Versions et traductions.....	8
L’arbre des nœuds de contenu	9
Content nodes	9
L’arbre des nœuds (Content node tree)	9
Conclusion et ressources	11

Concepts d'eZ Publish

eZ Publish est un *Système de Gestion de Contenu* (CMS pour *Content Management System*) très flexible et personnalisable. Il peut être utilisé aussi bien pour des petites réalisations comme des pages personnelles que pour des solutions Web, pour les entreprises qui souhaiteraient mettre en place, par exemple, des accès restreints avec gestion des droits, des vitrines, des forums de discussion et bien d'autres fonctions plus avancées. Basé sur les technologies et principes Open Source, il permet aisément l'ajout d'extensions et sa personnalisation, afin d'interagir avec d'autres solutions.

Ce tutoriel est le premier d'une série se focalisant sur les concepts et les caractéristiques, détaillés dans le livre [eZ Publish Content Management Basics](#) paru récemment. La série cible les débutants qui n'ont pas ou peu d'expérience avec le logiciel. Dans ce tutoriel, nous fournissons une explication compréhensible, non technique, des concepts clés d'eZ Publish, incluant la séparation du contenu et du design, le modèle de contenu orienté Objet et l'arbre des nœuds de contenu.

Système de Gestion de Contenu

Le rôle d'un *Système de Gestion de Contenu* est d'organiser et de stocker du contenu, sans tenir compte de son type et de sa complexité. Les éléments d'information dans une installation d'eZ Publish sont définis à la fois par leur *Content Class* (c'est-à-dire le type de contenu, par exemple une image ou un article) et leur *Content Hierarchy* (arborescence des données). Le but principal d'un tel système est de fournir une solution flexible bien structurée, déjà automatisée, qui permette de distribuer librement de l'information et de la mettre rapidement à jour au travers différents moyens de communication (comme Internet, des Intranets et divers systèmes de backend/frontend). Pour résumer, un CMS collecte, organise et publie du contenu sur divers types de sites Internet.

Séparation du contenu et du design

L'un des points fondamentaux dans l'élaboration d'un site qui se veut facilement organisable et actualisable, est de s'assurer que le système sépare bien clairement le contenu de la présentation visuelle. La séparation du contenu et du design permet les avantages suivants :

- Les rédacteurs et les designers peuvent travailler séparément sans conflits.
- Le contenu peut facilement être publié dans divers formats.
- Le contenu peut facilement être transféré et republié.
- Les changements globaux au niveau du design peuvent être appliqués par de simples modifications.

Contenu

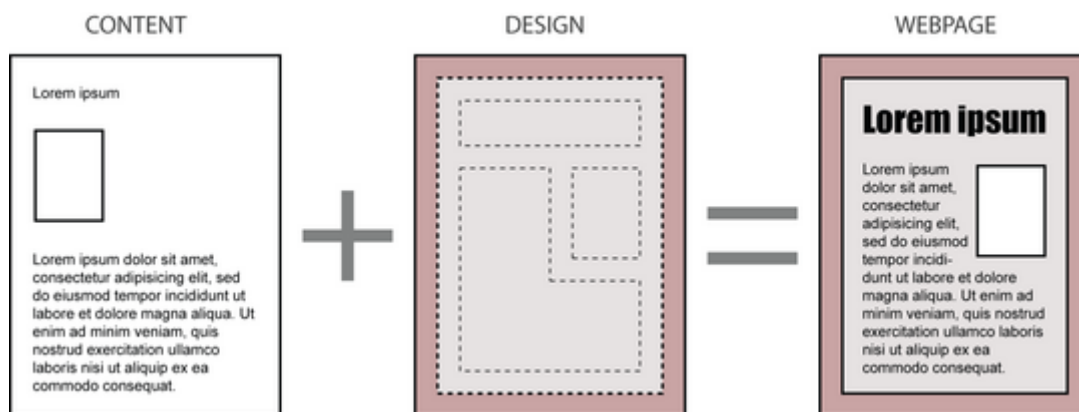
Le *Contenu* est l'information qui est organisée et stockée, de manière structurée, par eZ Publish. Par exemple, le contenu peut correspondre aux composants d'un article de News (titre, introduction, corps de texte, images) ou aux propriétés d'un produit (modèle, année, couleur). Toute information qui est stockée afin d'être réutilisée ultérieurement est référencée comme *Contenu*.

Design

La présentation du contenu est définie par le design d'un site. Tout comme on assimile le *Contenu* à des données structurées, le *Design* représente la manière dont les données seront présentées visuellement. Le *Design* correspond aux divers éléments qui embellissent une interface web : HTML, feuilles de style (CSS), images ne faisant pas partie du contenu, typographie, polices, etc.

Les composants d'une page Web

Alors que le *Contenu* correspond à l'enregistrement et la structuration des données, le *Design* implique la manière dont le contenu doit être présenté. Ces deux éléments s'assemblent pour former une interface complète - *une page Web* – comme le montre le schéma ci-dessous.



Contenu + Design = Page Web

Un *Site Web* fait référence à tout un ensemble, par exemple <http://www.ez.no>, alors qu'une *Page Web* fait référence à une partie spécifique d'un site, par exemple http://www.ez.no/ez_publish. Un site Web correspond à une série de pages Web ; il englobe et comprend tout ce qui appartient à un domaine particulier sur le Web : paramètres de configuration, une base de données contenant la structure et les éléments de contenu, les fichiers liés au contenu et ceux liés au design.

Siteaccess

Interfaces

Le contenu d'un site peut être affiché (et modifié) suivant différentes manières. Ceci est possible au travers d'*Interfaces*. Au minimum, chaque site possède deux interfaces : une interface dite d'*Administration* et une interface publique visible par les visiteurs. Cette dernière peut également inclure une interface d'*Edition* pour l'agencement du contenu.

Exemple : Un administrateur utilisera l'*Interface d'Administration* afin de construire et modifier un site, tandis qu'un éditeur (ou rédacteur) utilisera l'*Interface d'Edition* pour gérer le contenu.

Siteaccess

Afin de décider quelle *Interface* du site doit être affichée, eZ Publish utilise une solution générique intitulée *Siteaccess*. Les *Siteaccess* définissent deux choses :

- La manière dont eZ Publish reconnaît l'interface à laquelle on accède
- Divers paramètres de configuration qui surchargeront ceux par défaut

Les surcharges de configuration les plus importantes sont les paramètres concernant le design et la base de données utilisée.

Un moyen de reconnaître une interface est son adresse (URL) dans le navigateur Web. En d'autres termes, une *Interface* fait allusion à la présentation visuelle tandis qu'un *Siteaccess* intervient au niveau de la gestion du système. L'*Interface d'Administration* correspond au Siteaccess *admin*, et l'interface publique (pouvant contenir une *Interface d'Edition*) correspond au Siteaccess *public*. Les sites eZ Publish multilingues possèdent aussi, généralement, une interface différente pour chacune des langues disponibles. De ce fait, ils possèdent plusieurs Siteaccess publics, soit un par langue.

L'Interface d'Edition

La plupart des tâches de gestion de contenu basiques peuvent être effectuées via l'*Interface d'Edition*. Il s'agit d'une Interface Utilisateur Graphique (GUI) bien plus simple d'utilisation que l'*Interface d'Administration*. L'une des raisons est que l'*Interface d'Edition* est intégrée à l'interface publique du site. De ce fait, vous pouvez naviguer sur le site à travers des menus et des liens familiers pour localiser du contenu. Une seconde raison qui fait que l'*Interface d'Edition* est beaucoup plus simple que l'*Interface d'Administration* est la *barre d'outils*. Elle permet d'accéder facilement aux opérations d'édition concernant le contenu que vous êtes en train de visualiser.

L'Interface d'Administration

L'*Interface d'Administration* est l'interface de gestion avancée, mettant à disposition de puissants outils pour la gestion et l'édition du contenu, tels que l'administration, la configuration, la personnalisation et le développement du site. Ce qui rend l'*Interface d'Administration* plus avancée n'est pas seulement ce qu'elle vous permet de faire, mais l'interface elle-même. La présentation visuelle du contenu y est décortiquée, retirant ainsi tous les éléments du design, la mise en page et les styles se rapportant au Siteaccess public. Ceci dit, l'*Interface d'Administration* possède sa propre mise en page, mais elle est limitée à l'organisation structurelle du contenu, avec des menus de navigation, un accès aux opérations de gestion et des interfaces embarquées à but spécifique (pour gérer les versions des contenus ou créer un nouveau compte d'utilisateur par exemple).

Le système de Template

Les *Templates* représentent l'unité fondamentale du design d'un site dans eZ Publish. Par exemple, un Template peut ordonner à une page d'afficher le titre du site en haut et le contenu principal au centre. Lorsqu'on accède à une page, le CMS place le contenu (le titre et le texte) aux emplacements appropriés définis dans le template.

Nous avons vu précédemment que l'assemblage du *Contenu* et du *Design* constituait une *Page Web*. Le Siteaccess spécifie, entre autres, quel jeu de fichiers, relatifs au design, utiliser (feuilles de style, images, templates). Un *Template* eZ Publish est un fichier HTML étendu et personnalisable, qui décrit comment tels ou tels types de contenu doivent être présentés. Pour résumer, les Templates représentent l'élément de transformation qui produit la mise en page du site, aussi bien pour une page Web entière que pour certaines parties de cette page.

Un système axé sur les Composants

Le système de template d'eZ Publish est axé sur les *Composants*. Cela signifie que, dans la plupart des cas, une *Page Web* est créée en combinant plusieurs *Templates*. Au minimum, eZ Publish retourne le Template principal, qui se nomme *pagelayout*. Le pagelayout contient les tags `<html>`, `<head>` et `<body>`; accompagnés de feuilles de style (CSS), qui dirigent l'aspect global d'un site. En d'autres termes, le pagelayout indique le titre, le logo, le menu et le pied de page qui seront présentés pour chaque Page Web que le système générera.

Il existe différents types de Templates qui s'ajoutent au Template principal (le pagelayout). Certains sont intégrés au logiciel alors que d'autres pourront être personnalisés. Il existe des règles qui, selon le contexte, indiquent comment le contenu doit être affiché, sans tenir compte de la façon dont il est structuré et enregistré. Le contexte peut être un Siteaccess spécifique ou un élément intégrant du contenu. Par exemple, une image peut être affichée différemment selon si elle fait partie d'un article ou d'une galerie.

Les tâches des Templates

En plus de la syntaxe HTML standard, il est possible d'utiliser un code spécifique à eZ Publish dans les Templates afin, par exemple, d'extraire et d'afficher du contenu provenant d'une autre partie de l'arborescence des données.

Un modèle de contenu orienté objet

Qu'est-ce que l'orientation objet ?

La structure de contenu d'eZ Publish est basée sur des idées empruntées aux langages de programmation orientés objet tels que Java et C++. Superficiellement, l'*Orienté Objet* (OO) signifie visualiser le monde en terme d'objets. Dans la vie, nous sommes entourés d'innombrables objets, tels que des meubles, des voitures, des animaux de compagnie, et d'autres personnes. Chacun de ces objets possède des traits spécifiques. C'est également de cette manière que le contenu est décrit et géré dans eZ Publish - avec des objets et des attributs (traits).

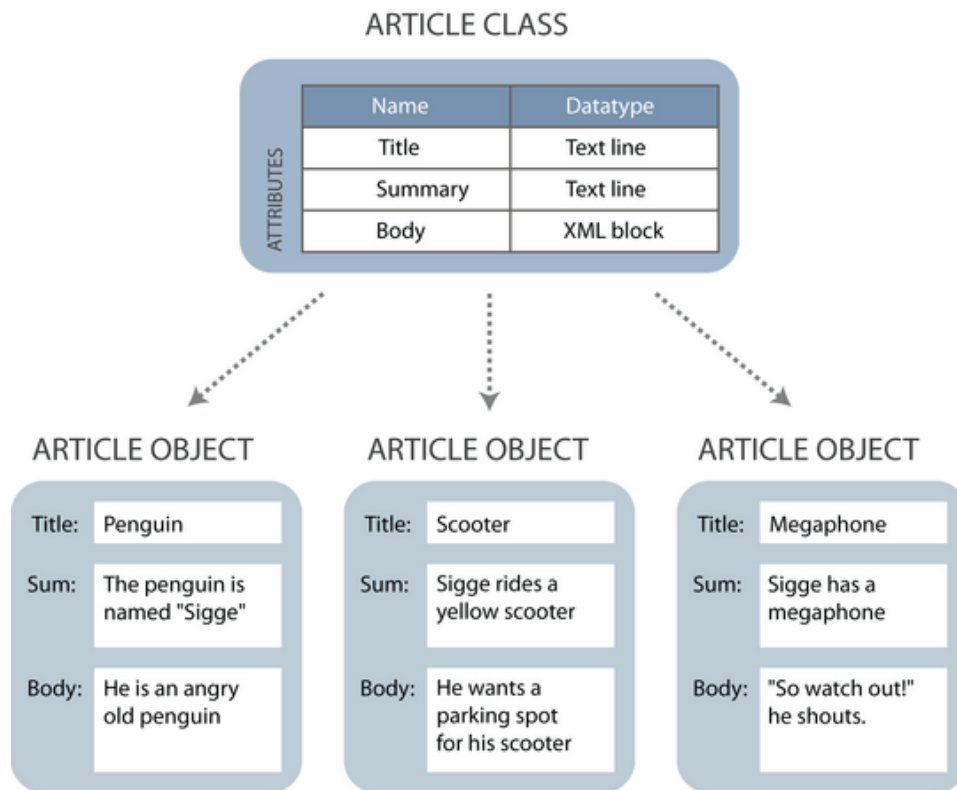
eZ Publish supporte différents types de contenu, tels que des images, des articles, des fichiers multimédia, des forums et des formulaires de contact. Un type de contenu est appelé *Content Class* (ou *Class* pour faire plus court), alors qu'une partie spécifique du contenu est appelé *Content Object* (ou *Object*).

Une structure de données est définie par une *Content Class*, et une *Content Class* est faite d'attributs. Un *Attribut* peut être considéré comme un champ. Les caractéristiques des attributs sont déterminées par les *Datatypes* (types de données). L'illustration qui suit résume et montre la relation qui existe entre les *Datatypes*, les *Attributs*, les *Content Classes* et les *Content Objects*. Les *Content Objects* sont des *Instances* de la même *Content Class*, ce qui signifie qu'ils sont de même type mais contiennent des données différentes.

Content class

Une *Content Class* peut être perçue comme un plan structurel pour un type particulier de contenu. Les propriétés de ce contenu sont désignées comme attributs. Par exemple, la *Content Class* "Article" contient des attributs tels que *Title*, *Name of Author*, *Introduction Text* et *Body Text* (texte de l'article).

Un *Datatype* décrit le type de valeur qui peut être stocké dans un attribut et est la plus petite entité de stockage possible. Il détermine la manière dont un type spécifique d'information doit être validé, enregistré et récupéré.



Des Datatypes, des attributs, une Content Class et des Objets

Content object

Il est important de comprendre qu'une *Content Class* est seulement une définition d'une structure arbitraire. En d'autres termes, la classe elle-même ne stocke aucune donnée réelle. A partir du moment où une *Content Class* a été définie, il est possible de créer des *Instances* de cette classe. Une *Instance* d'une *Content Class* est appelée *Content Object*. Le contenu est stocké dans les objets (par exemple, *folders*, *articles*, *comments*, *employees* et *members*) via les attributs.

Versions et traductions

Un *Content Object* comprend au moins une *Version*. Pour chaque modification effectuée sur un objet et sauvegardée, une *Version* est enregistrée. Si les modifications sont sauvegardées pour la première fois (comparativement à la façon dont l'objet s'affiche sur le site Web), une nouvelle *Version* est créée. Autrement, une *Version* existante sera simplement mise à jour. Cette couche additionnelle permet de posséder différentes versions d'un même contenu. Par ailleurs, chaque *Version* correspond au minimum à une *Traduction*. La couche de *Traduction* permet de représenter la même version du même contenu dans plusieurs langues. C'est cette dernière couche de *Traduction* qui détient les attributs d'un objet.

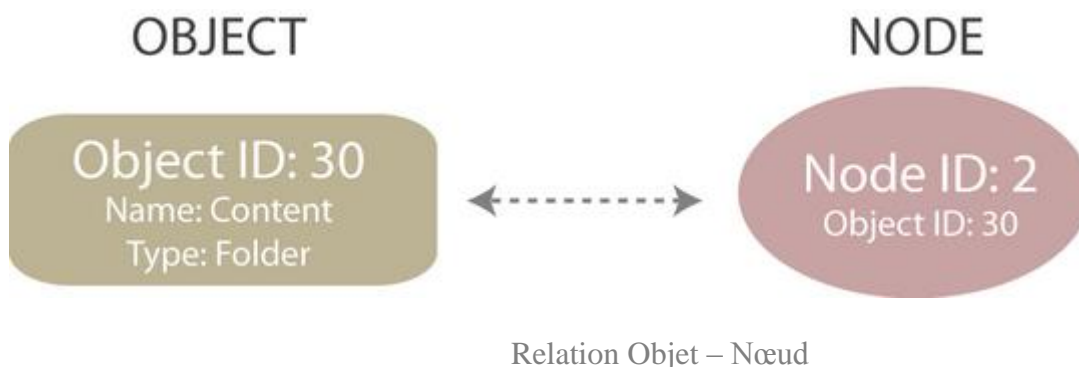
L'arbre des nœuds de contenu

eZ Publish apporte également un autre couche abstraite tirée de l'orientation objet classique qui est le concept de nœud. Abstraite fait référence ici au fait que le système vous permet de penser à un article sans vous préoccuper de la manière dont il sera géré en interne.

Content nodes

Dans eZ Publish, Les *Content Objects* sont englobés (ou *encapsulés*) et structurés en utilisant des *Content Node* (Nœuds de Contenu). Un nœud englobe un *Content Object* particulier afin de fournir un mécanisme d'accès aux données stockées dans l'objet et de l'afficher sur le site. Les objets ne possèdent pas de capacités structurelles propres. En d'autres mots, les *Content Objects* ne peuvent pas se positionner eux-mêmes par rapport aux autres objets. Encapsuler un nœud autour d'un objet le permet. De plus, les nœuds représentent les versions publiées des *Content Objects*.

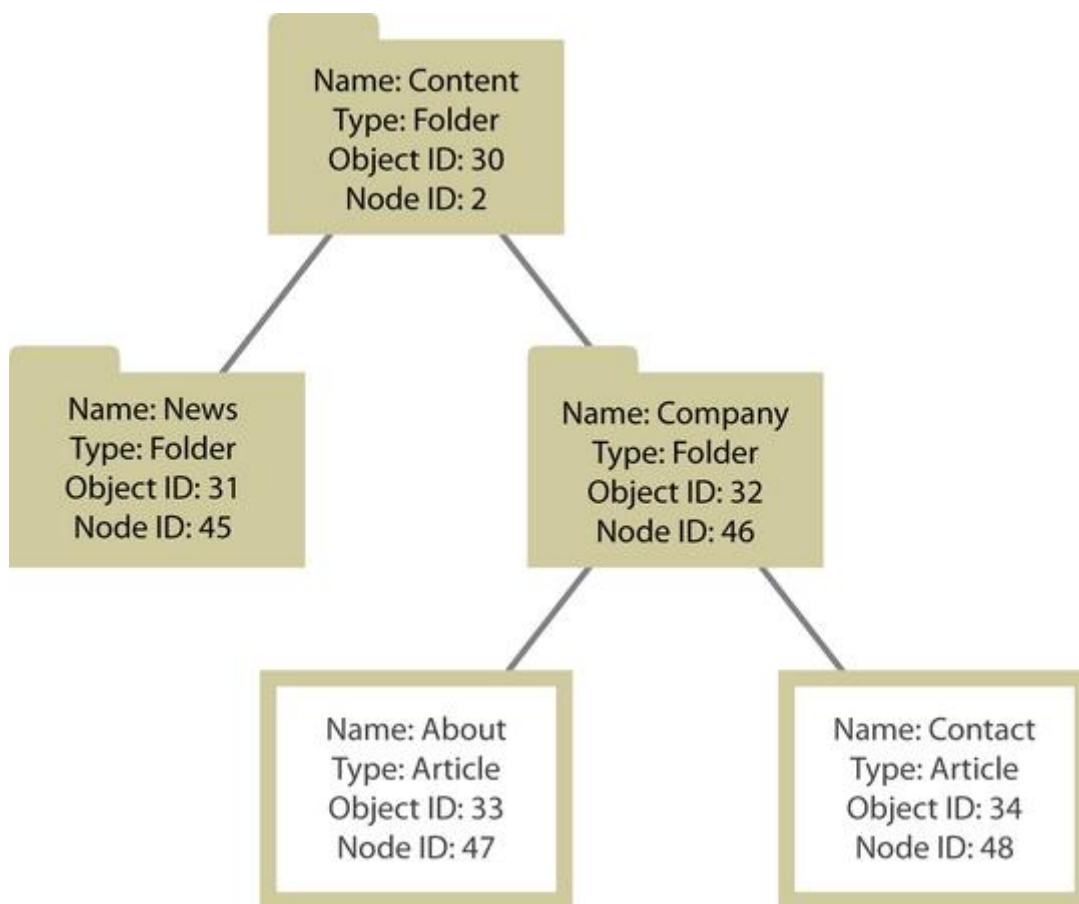
L'illustration suivante affiche un exemple de nœud et de son objet associé tel qu'il serait représenté dans le système.



L'arbre des nœuds (Content node tree)

Les nœuds sont organisés dans un arbre (*Node Tree*). L'arbre est divisé en trois *branches* principales : *Content*, *Media* et *Users* (ainsi que quelques autres parties peu fréquemment utilisées par les éditeurs). La plupart du temps, vous serez amenés à travailler avec la branche *Content* de l'arbre. Le *Node Tree* est également connu sous le nom de *Content Hierarchy* (arborescence des données), qui est un concept important à connaître lors de la création de nouveau contenu.

Les illustrations ci-dessous affichent un exemple simplifié de la façon dont les objets sont référencés par les nœuds pour construire l'arbre des nœuds, d'un point de vue utilisateurs.



Arbre de nœuds

Les *Content Objects* peuvent être à la fois des objets individuels et des conteneurs pour du contenu situé après eux dans la hiérarchie. Par exemple, un ou plusieurs articles sont généralement stockés sous un dossier (Folder). Lorsque le dossier est affiché, il montre automatiquement une liste sommaire des articles stockés sous lui. Chaque *Content Class* possède un Template différent pour afficher ses objets. Toutefois, bien qu'un dossier affiche certains des attributs des objets situés sous celui-ci dans la hiérarchie, un article ne va pas nécessairement en faire de même.

Conclusion et ressources

Ce tutoriel correspond à la version traduite de l'article [An Introduction to eZ Publish Concepts](#) rédigé par [Bergfrid Skaara](#) et donne une idée du contenu présent dans le livre [eZ Publish Content Management Basics](#). Il décrit les concepts clés d'eZ Publish, incluant l'importance de la séparation du contenu et du design, le système de templates et un aperçu du modèle de contenu.

Ce qui suit est une liste des termes clés et des définitions introduites dans cet article :

- Content Management System (CMS): Utilisé pour collecter, gérer et publier du contenu.
- Contenu : Information (données) organisée et stockée dans une structure.
- Design : Présentation visuelle, mise en page et emplacement des données.
- Siteaccess : Indique au système comment reconnaître l'interface à laquelle nous tentons d'accéder.
- Template : Fichier HTML personnalisé qui détermine la mise en page et le design à utiliser en fonction du type de contenu.
- Content object / class : Un type de contenu est appelé *Content Class*, alors qu'une partie spécifique d'un contenu est appelé *Content Object*. Une *Content Class* peut être perçue comme un plan structurel pour un type de contenu particulier. Les propriétés d'une classe sont définies en tant qu'*Attributes*.
- Content node : Dans eZ Publish, les *Content Objects* sont englobés (ou encapsulés) et structurés en utilisant des nœuds. Les nœuds permettent aux objets d'être placés dans l'arbre des nœuds.
- (Content) node tree : Les nœuds sont organisés dans un arbre divisé en trois branches principales : *Content*, *Media* et *Users*. L'arbre est également connu sous le nom *Content Hierarchy*.