empreinte digitale

COMPRENDRE L'HUMAIN, ADAPTER LA TECHNOLOGIE !

Le 8 janvier 2018

MENUISERIE AVENIR

Toutsurmamenuiserie.com

Synthèse architecture technique

EMPREINTE DIGITALE - GROUPE V-TECHNOLOGIES
3 rue Louis Boisramé – 49000 ANGERS 02 41 72 10 75 02 41 72 10 79
contact@empreintedigitale.fr www.empreintedigitale.fr
SARL au capital de 421 150 euros . SIRET 393 267 091 000 71 . APE 6202A

Ce document récapitule l'architecture technique de l'application toutsurmamenuiserie, ainsi que la sécurité et l'étanchéité des données.

1. HÉBERGEMENT

1.1. Serveurs

L'hébergement de la solution est composée de deux serveurs virtuels, un de préproduction et un de production :

- PREPROD/TEST (VT49) :
  Coeurs : 2
  RAM : 2 Go
  Disque : 50Go
- PROD (VT58) :
  Coeurs : 4
  RAM : 4 Go
  Disque : 50 Go

Ces ressources sont amenées à évoluer à la hausse en terme de coeurs et de mémoire notamment pour la production.

La distribution utilisée est **Debian Jessie** (8.x) 64 bits

1.2. Services

Les principaux services qui tournent sur les serveurs sont les suivants :

- Script : **PHP** 7.0.x
- Serveur web : **Nginx** 1.10.x
- Base de données : **MariaDB** 10.1.x
- Surveillance : **Monit** 5.23
- Métrologie : **Munin**

1.3. Firewall

La régle par défaut est de tout fermer en entrée comme en sortie. On ouvre ensuite au cas par cas.

1.3.1. Ouvertures en entrée

- http (80)
- https (443)
- ssh/sftp (888)

1.3.2. Ouvertures en sortie

- http (80) vers 91.223.76.20 (notre serveur git)
- https (443) vers 91.223.76.20 (notre serveur git)

Tous les appels web du serveur vers l'extérieur doivent donc passer par notre proxy squid.

1.4. Sauvegardes

Les bases de données sont sauvées localement toutes les nuits à 04h04 via le script **automysqlbackup**.
De plus, on sauvegarde quotidiennement le répertoire contenant les sources de l'applicatif et le répertoire automysqlbackup précédent sur un autre serveur.

2. ARCHITECTURE TECHNIQUE

L'architecture technique de cette application est basée sur celle de Symfony, un framework PHP basé sur une structure MVC.
Autour de cet outil, nous utilisons l'ORM Doctrine pour gérer les données de l'application. Pour la gestion du back-office, nous avons mis en place le bundle Sonata Admin se basant sur la méthode CRUD.
Pour la partie front-office, nous utilisons le moteur de template Twig.

3. GESTION DES DONNÉES

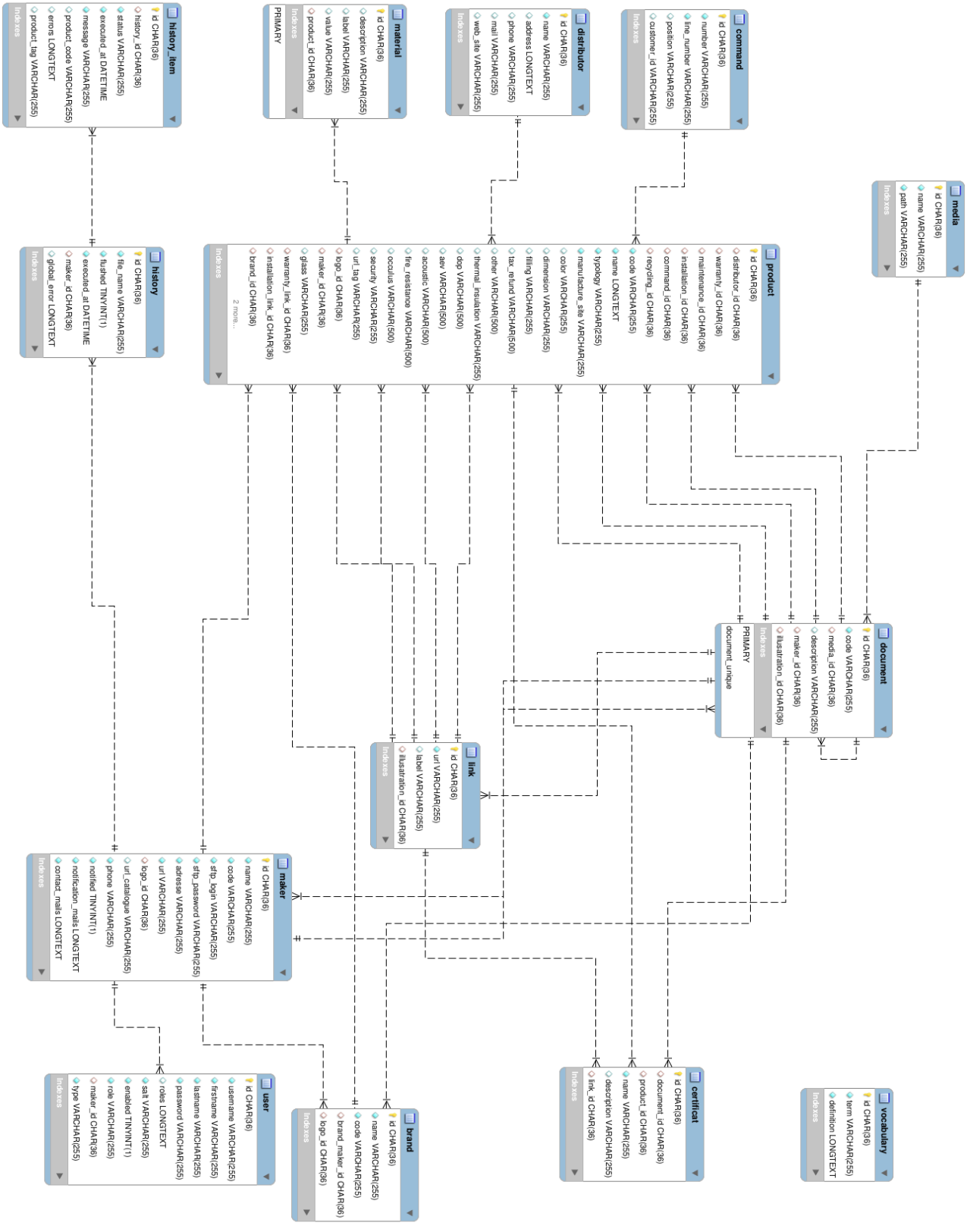
3.1. Liste des clés

TABLE_NAME	CONSTRAINT_NAME	KEY_COMPOSE
document	document_unique	code(document),maker_id(document)
history_item	FK_142E55621E058452	history_id(history)
brand	FK_1C52F958AFEACDD2	brand_maker_id(maker)
brand	FK_1C52F958F98F144A	logo_id(document)
certificat	FK_27448F774584665A	product_id(product)
certificat	FK_27448F77ADA40271	link_id(link)
certificat	FK_27448F77C33F7837	document_id(document)
history	FK_27BA704B68DA5EC3	maker_id(maker)
link	FK_36AC99F19FC657D4	illusatration_id(document)
material	FK_7CBE75954584665A	product_id(product)
user	FK_8D93D64968DA5EC3	maker_id(maker)
maker	FK_C6197FB4F98F144A	logo_id(document)

product	FK_D34A04AD167B88B4	installation_id(document)
product	FK_D34A04AD2D863A58	distributor_id(distributor)
product	FK_D34A04AD2EC1782C	warranty_id(document)
product	FK_D34A04AD33E1689A	command_id(command)
product	FK_D34A04AD44F5D008	brand_id(brand)
product	FK_D34A04AD4A71D145	warranty_link_id(link)
product	FK_D34A04AD68DA5EC3	maker_id(maker)
product	FK_D34A04ADD08DEC6C	recycling_id(document)
product	FK_D34A04ADD31F4768	recycling_link_id(link)
product	FK_D34A04ADD45247E3	maintenance_link_id(link)
product	FK_D34A04ADF2376C0E	installation_link_id(link)
product	FK_D34A04ADF6C202BC	maintenance_id(document)
product	FK_D34A04ADF98F144A	logo_id(document)
document	FK_D8698A7668DA5EC3	maker_id(maker)
document	FK_D8698A769FC657D4	illusatration_id(document)
document	FK_D8698A76EA9FDD75	media_id(media)
brand	PRIMARY	id(brand)
certificat	PRIMARY	id(certificat)
command	PRIMARY	id(command)
distributor	PRIMARY	id(distributor)
document	PRIMARY	id(document)
history	PRIMARY	id(history)
history_item	PRIMARY	id(history_item)
link	PRIMARY	id(link)
maker	PRIMARY	id(maker)
material	PRIMARY	id(material)
media	PRIMARY	id(media)
migration_versions	PRIMARY	version(migration_versions)
product	PRIMARY	id(product)
SCHEDULED_COMMAND	PRIMARY	ID_SCHEDULED_COMMAND(SCHEDULED_COMMAND)
user	PRIMARY	id(user)
vocabulary	PRIMARY	id(vocabulary)
brand	UNIQ_1C52F958F98F144A	logo_id(brand)

certificat	UNIQ_27448F77ADA40271	link_id(certificat)
maker	UNIQ_C6197FB4F98F144A	logo_id(maker)
product	UNIQ_D34A04AD33E1689A	command_id(product)
product	UNIQ_D34A04AD4A71D145	warranty_link_id(product)
product	UNIQ_D34A04ADD31F4768	recycling_link_id(product)
product	UNIQ_D34A04ADD45247E3	maintenance_link_id(product)
product	UNIQ_D34A04ADF2376C0E	installation_link_id(product)
document	UNIQ_D8698A76EA9FDD75	media_id(document)
product	url_unique	url_tag(product)
user	username_idx	username(user)

3.2. MPD


media
id CHAR(36)
name VARCHAR(255)
path VARCHAR(255)
Indexes
command
id CHAR(36)
number VARCHAR(255)
line_number VARCHAR(255)
position VARCHAR(255)
customer_id VARCHAR(255)
Indexes
distributor
id CHAR(36)
name VARCHAR(255)
address LONGTEXT
phone VARCHAR(255)
mail VARCHAR(255)
web_site VARCHAR(255)
Indexes
material
id CHAR(36)
description VARCHAR(255)
label VARCHAR(255)
value VARCHAR(255)
product_id CHAR(36)
Indexes
PRIMARY
history_item
id CHAR(36)
history_id CHAR(36)
status VARCHAR(255)
executed_at DATETIME
message VARCHAR(255)
product_code VARCHAR(255)
errors LONGTEXT
Indexes
product_tag VARCHAR(255)
product
id CHAR(36)
distributor_id CHAR(36)
warranty_id CHAR(36)
maintenance_id CHAR(36)
installation_id CHAR(36)
command_id CHAR(36)
recycling_id CHAR(36)
code VARCHAR(255)
name LONGTEXT
typology VARCHAR(255)
manufacture_site VARCHAR(255)
color VARCHAR(255)
dimension VARCHAR(255)
filling VARCHAR(255)
tax_refund VARCHAR(500)
other VARCHAR(500)
thermal_insulation VARCHAR(255)
dop VARCHAR(500)
aev VARCHAR(500)
acoustic VARCHAR(500)
fire_resistance VARCHAR(500)
oculus VARCHAR(500)
security VARCHAR(255)
url_tag VARCHAR(255)
logo_id CHAR(36)
maker_id CHAR(36)
glass VARCHAR(255)
warranty_link_id CHAR(36)
installation_link_id CHAR(36)
brand_id CHAR(36)
2 more...
Indexes
history
id CHAR(36)
file_name VARCHAR(255)
flushed TINYINT(1)
executed_at DATETIME
maker_id CHAR(36)
global_error LONGTEXT
Indexes
document
id CHAR(36)
code VARCHAR(255)
media_id CHAR(36)
description VARCHAR(255)
maker_id CHAR(36)
illustration_id CHAR(36)
Indexes
PRIMARY
document_unique
link
id CHAR(36)
url VARCHAR(255)
label VARCHAR(255)
illustration_id CHAR(36)
Indexes
maker
id CHAR(36)
name VARCHAR(255)
code VARCHAR(255)
sftp_login VARCHAR(255)
sftp_password VARCHAR(255)
adresse VARCHAR(255)
url VARCHAR(255)
logo_id CHAR(36)
url_catalogue VARCHAR(255)
phone VARCHAR(255)
notified TINYINT(1)
notification_mails LONGTEXT
contact_mails LONGTEXT
Indexes
certificat
id CHAR(36)
document_id CHAR(36)
product_id CHAR(36)
name VARCHAR(255)
description VARCHAR(255)
link_id CHAR(36)
Indexes
vocabulary
id CHAR(36)
term VARCHAR(255)
definition LONGTEXT
Indexes
brand
id CHAR(36)
name VARCHAR(255)
code VARCHAR(255)
brand_maker_id CHAR(36)
logo_id CHAR(36)
Indexes
user
id CHAR(36)
username VARCHAR(255)
firstname VARCHAR(255)
lastname VARCHAR(255)
password VARCHAR(255)
roles LONGTEXT
salt VARCHAR(255)
enabled TINYINT(1)
role VARCHAR(255)
maker_id CHAR(36)
type VARCHAR(255)
Indexes

4. ÉTANCHÉITÉ DES DONNÉES

Pour un utilisateur connecté sur l'application, nous avons personnalisé certains contenus. Effectivement grâce au module d'authentification de Symfony (http://symfony.com/doc/current/components/security/authentication.html) et aux filtres de Doctrine (http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/filters.html), il nous a été possible de personnaliser toutes les requêtes pointant sur des données sensibles.

Exemple, un utilisateur souhaite récupérer la liste de ses documents. D'ordinaire la requête est la suivante : SELECT * FROM document
Dans l'application Toutsurmamenuiserie, la requête exécutée sera différente : SELECT * FROM document WHERE maker_id = 'id_du_maker_lie_a_l_utilisateur'
Dans cet exemple, on ajoute automatiquement une condition à chaque requête.

Cependant, cet exemple ne permet pas d'empêcher une personne d'accéder à un contenu d'une donnée sensible. Il va juste permettre de masquer la donnée à l'utilisateur.
C'est pourquoi, nous avons mis en place une fonctionnalité de Symfony : les voters (https://symfony.com/doc/current/security/voters.html). Cette dernière permet de vérifier si le contenu demandé par l'utilisateur lui appartient. Si ce n'est pas le cas, l'utilisateur reçoit une réponse du serveur HTTP 403 Forbidden. Il ne peut donc pas récupérer les données ne lui appartenant pas.

Toutes les données impactées par ces fonctionnalités sont :

- historique d'import
- gestionnaire de fabricant
- fabricant
- document
- marque
- produit

5. GLOSSAIRE

Symfony : https://symfony.com
Doctrine : http://www.doctrine-project.org/projects/orm.html
Twig : https://twig.sensiolabs.org
Sonata Admin : https://sonata-project.org/bundles/admin/3-x/doc/index.html
MVC : Modèle - Vue - Contrôleur
ORM : Object Relation Mapping
CRUD : Create - Read - Update - Delete