

# CH-III: La démarche de pilotage et de mise en œuvre d'un projet ERP

# Qu'est ce qui est le plus important dans une entreprise ?

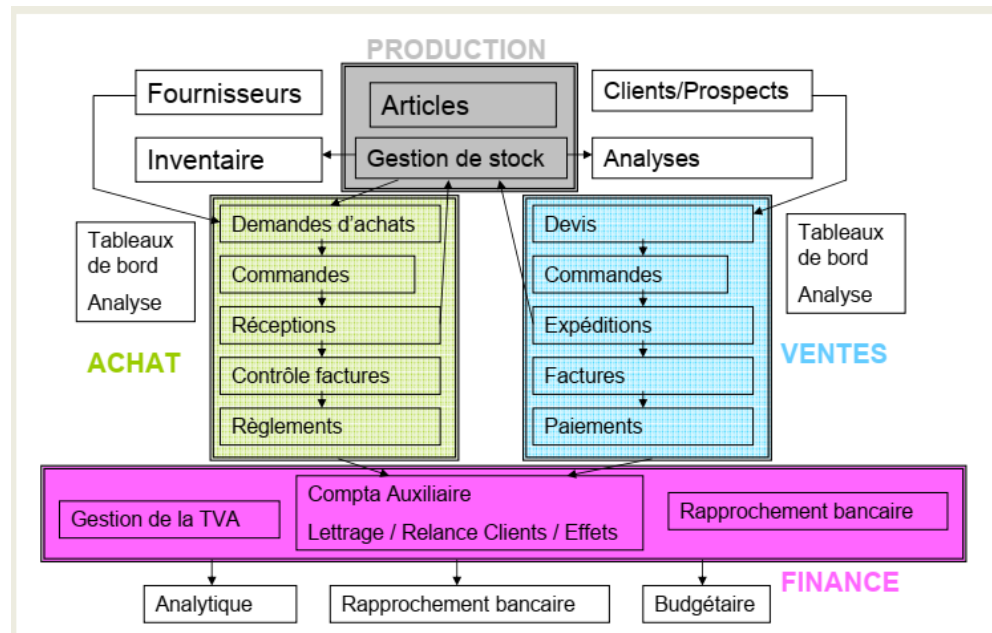
- Facturer ses clients et de se faire payer.
- Payer les employés et fournisseurs.
- Maximiser Le profit & Minimiser les charges

# Quels sont les métiers qui se ressemblent le plus?

Les ventes et les achats.

les ventes : on retire du stock pour livrer à une adresse.

Les achats : on entre en stock dans un magasin de l'entreprise.

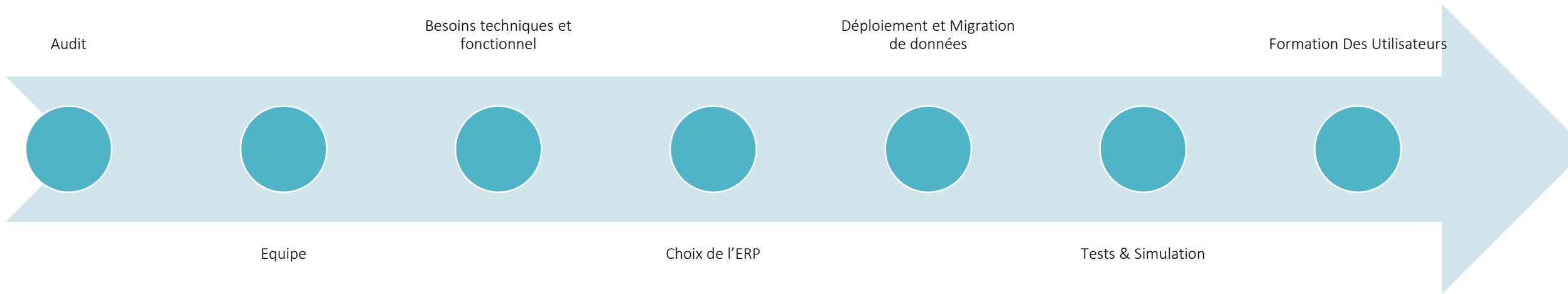


# Quels sont les fichiers de base ?

- Le plan comptable,
- Les articles,
- Les sites de stockage et les magasins,
- Les clients,
- Les tarifs,
- Les fournisseurs,

# Les différentes phases d'implémentation d'un projet ERP

Un projet ERP est une démarche de développement stratégique au sein d'une entreprise qui implique différentes phases pour sa préparation et sa réalisation. Voici les différentes étapes à suivre pour garantir la réussite du projet.



# Réaliser un audit de l'organisation de l'entreprise

L'évaluation des besoins de l'entreprise en système d'information est primordiale pour déterminer si l'investissement dans l'implantation d'un ERP est nécessaire.

Pour cela, il faut analyser le flux d'informations au sein de l'entreprise, les pratiques comptables et commerciales ainsi que les méthodes de gestion des stocks, des ventes...

À la suite de cela, il faut se poser la question de l'utilité d'un ERP et déterminer les différents objectifs à atteindre.

# Mettre en place une équipe en charge du projet

Lorsqu'il est avéré que l'implantation d'un progiciel ERP dans le système d'information constituera un levier de développement de l'entreprise, il faut mettre en place une équipe en charge de l'étude et de la réalisation du projet.

Si le chef de projet est dans la majorité des cas issu du département technique et informatique, d'autres ressources issues d'autres départements sont nécessaires pour garantir la bonne intégration des différents besoins



# Evaluer les besoins et mettre en place un cahier des charges techniques et fonctionnel

Lorsque l'équipe est sur pied, il faut évaluer avec précision la configuration actuelle du système d'information et les besoins de l'entreprise à travers le nouveau logiciel ERP. C'est à partir des différents besoins identifiés que les différents modules du progiciel seront sélectionnés.

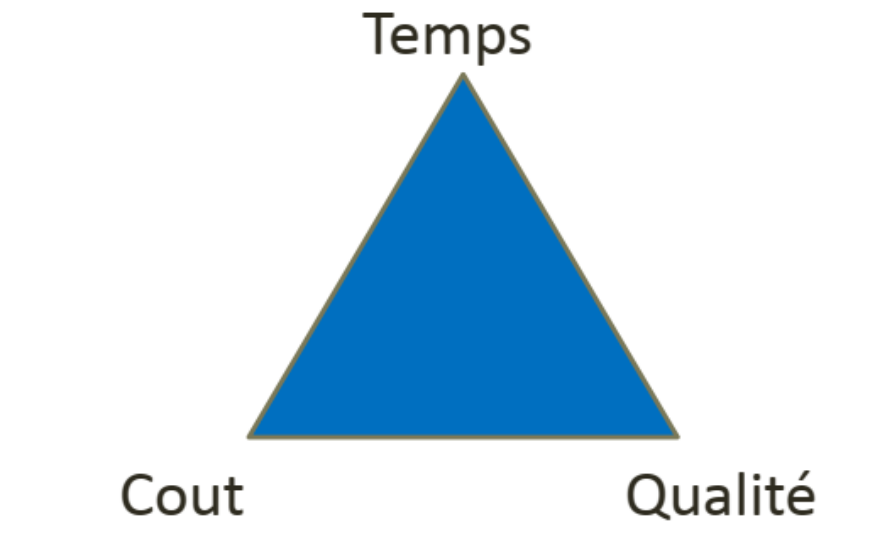
Le cahier des charges technique et fonctionnel regroupera toutes les données nécessaires tout au long du projet.

La direction de l'entreprise, animée d'une vision à long terme, doit s'impliquer dans l'évaluation des besoins, afin que le projet ERP fasse partie intégrante de la politique de croissance.

# Choix d'ERP

# Qu'est ce qui guide le choix ?

- Les fonctionnalités que les utilisateurs de l'entreprise ont besoin, appelé le périmètre (Qualité).
- Le budget pour changer (Cout).
- Le délai de mise en place (Temps).



La solution qui correspondra  
en tous points n'existe pas.

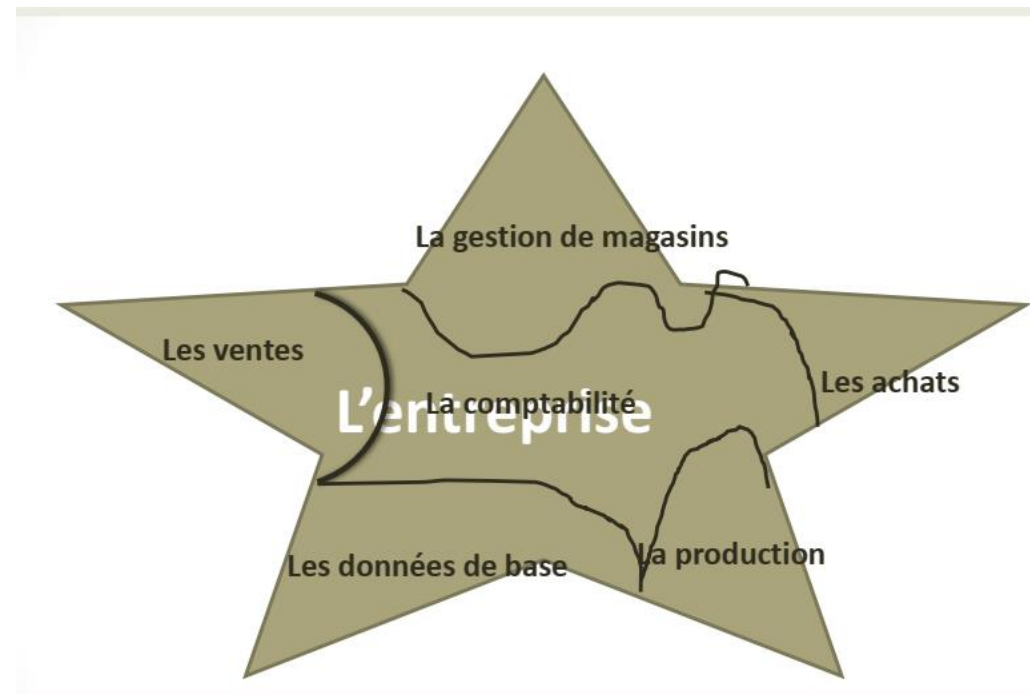
# Comment procéder?

On identifie les métiers dans l'entreprise et les personnes qui connaissent bien leur métier, on les appelle les « Utilisateurs clés » ou « keys users ». Comptable, assistante commerciale, planificateur, acheteur, magasinier...

On demande aux utilisateurs les processus et les fonctionnalités dont ils ont besoins.

# On découpe le projet

On identifie les processus Metier



On consulte chaque solution et on compare avec chacun de nos processus.  
Pour chaque processus et sous processus  
on applique le **MoSCoW**

**M** comme Must have, doit avoir.

**S** comme Should have, devrait avoir.

**C** comme could have, pourrait avoir

**W** comme Won't have, ne doit pas avoir.

# On établit un comparatif des solutions face aux fonctionnalités

Processus actuel	description	ERP X	ERP Y	ERP Z
ACH02_saisie_ordre_achat_0.doc	Saisie de l'ordre d'achat à un fournisseur habituel pour un article géré en stock	Ok	Ok	Ok
ACH03_Reception_d_un_ordre_achat_0.doc	Réception de l'ordre d'achat en 1 ou plusieurs fois,	Ok	Ok	Ok
ACH04_Facturation_de_la_réception_d_un_ordre_achat_0.doc	Facturation de la réception	Ok	Ok	Ok
ACH05_Reception_quantite_recue_sup_qte_commande_0.doc	Réception de vrac 'sable' 'ferraille' la quantité peut être supérieur à la quantité commandé	Non	Ok	Pas vu
ACH06_Reception_de_plusieurs_ordres_achat_0.doc	Plusieurs OA du meme fournisseur.	Non	Ok	Ok
ACH07_Facturation_ajout_frais_transport_0.doc	Frais de transports et d'emballage en plus.	Transport Oui	Non	Si dans la commande
ACH08_Facturation_de_réceptions_d_ordres_achat_0.doc	Facturation des réceptions en X fois.	Non	Ok	Non
ACH10_ajustement_de_stock_0.doc	Mouvement de stock issu de la réception	Ok	Ok	Ok
ACH11_conditionnement-achat-nomenclature_0.doc	Je gère à l'unité mais je recoit en carton de 12 unités	Ok	Non	Non
ACH12_création-fournisseur_0.doc	Création fournisseur Français, de UE, Hors UE	Ok	Ok	Ok
ACH13_reference_et_prix_du_fournisseur_0.doc	La référence de mon article chez mon fournisseur, son prix selon quantité	Ok	Non	Ok
ACH15_Réévaluation de stock_0	Valeur de mon stock pour fin de mois	Ok	Ok	Ok



# Elimination

On fait une première élimination.

- Fonctionnelle,
- Technique,
- Budgétaire.

Le but est de garder 2 ou 3 solutions en short liste

# Elimination

On revoit les 3 solutions de façon approfondie avec les keys users.

On prend un rôle d'observateur, on consigne les questions posées et les réponses pour compléter les fonctionnalités.

# Elimination

On demande la confirmation des fonctionnalités en standards de l'ERP et des adaptations à réaliser.

Toute adaptation doit être chiffrée par l'intégrateur.

# Choix de la solution

On fait une présentation à la direction, et aux key users des solutions finales.

La direction choisi ou fait choisir par son équipe de la solution à retenir.

On implante la solution.  
Ce n'est pas la meilleure ! Mais la  
moins mauvaise !  
De toute façon, celle qui est  
choisie.

# Le déploiement et la migration des données

Le déploiement consiste en l'intégration de l'ERP dans le système d'information. Le choix est large entre une installation en local, l'hébergement dans un serveur distant dédié ou l'installation en système cloud mutualisé.

Après l'installation, il faut procéder à la migration des différentes données dans la base unique de l'ERP. Il est primordial de faire une sauvegarde avant le basculement.

# La phase de test et de paramétrage

La phase de test et de paramétrage est critique dans la mesure où elle conditionne le bon fonctionnement de l'ERP tout au long de la vie de l'entreprise.

À ce moment, il est toujours possible d'ajouter ou de personnaliser les modules en fonction des différents besoins identifiés tout au long du processus de préparation et d'installation.

# La formation des utilisateurs

La phase de formation est une étape indispensable qui sera plus ou moins longue en fonction de la complexité de l'interface de l'ERP. Une formation optimale garantit une utilisation au quotidien qui tire profit de toutes les potentialités de l'ERP.



# TP : Débuter Avec Odoo

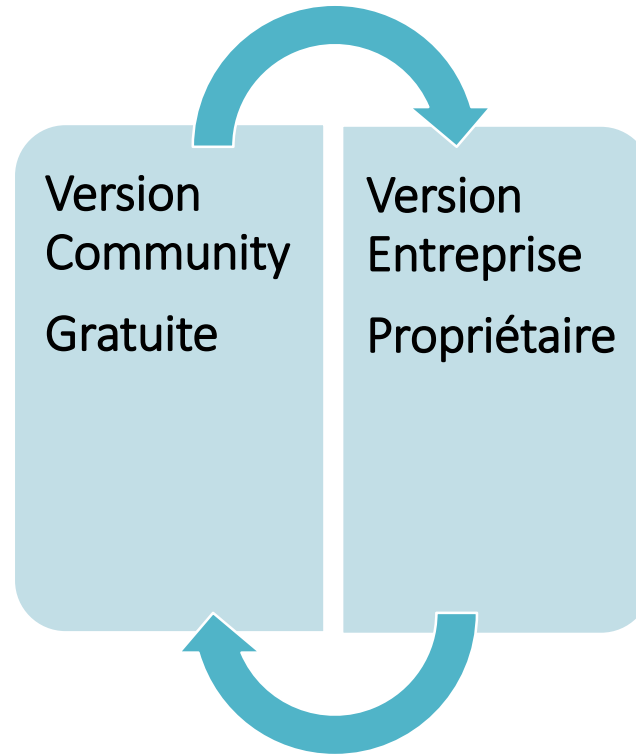
Installation & Configuration

# Odoo Ex. OpenERP / TinyERP

Odoo, anciennement OpenERP et Tiny ERP, est initialement un progiciel open-source de gestion intégré comprenant de très nombreux modules permettant de répondre à de nombreux besoins de gestion des entreprises (ERP), ou de gestion de la relation client (CRM). Le logiciel est utilisé par plus de quatre millions d'utilisateurs pour gérer leurs entreprises à travers le monde. Odoo est le système ERP open-source le plus populaire.



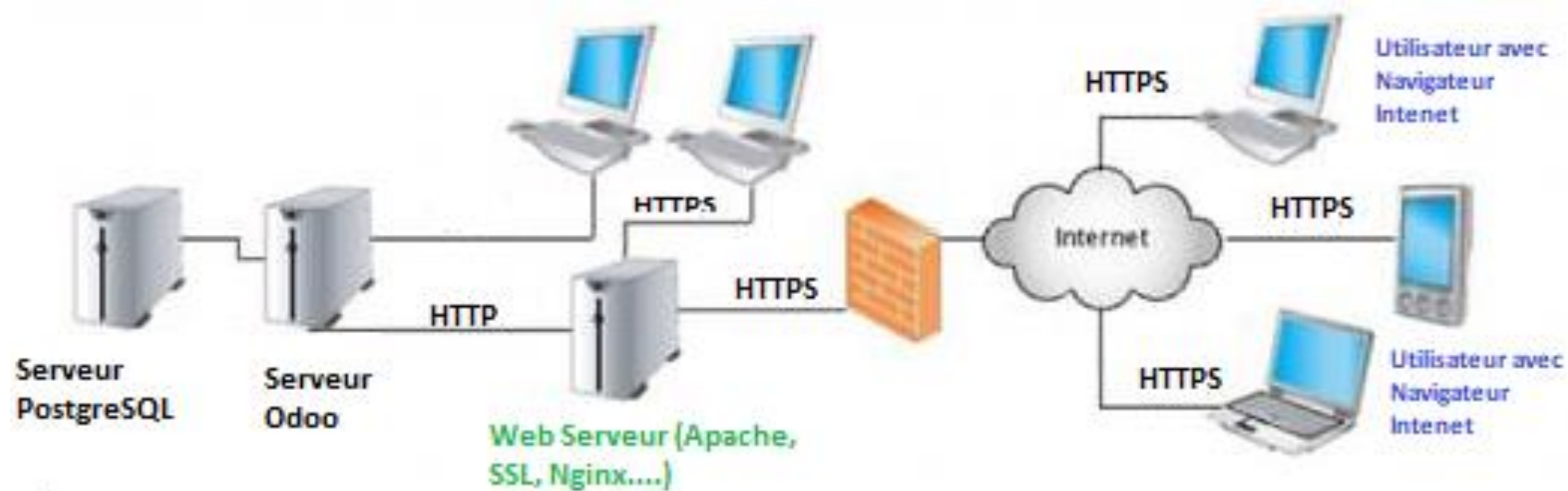
# Version Odoo



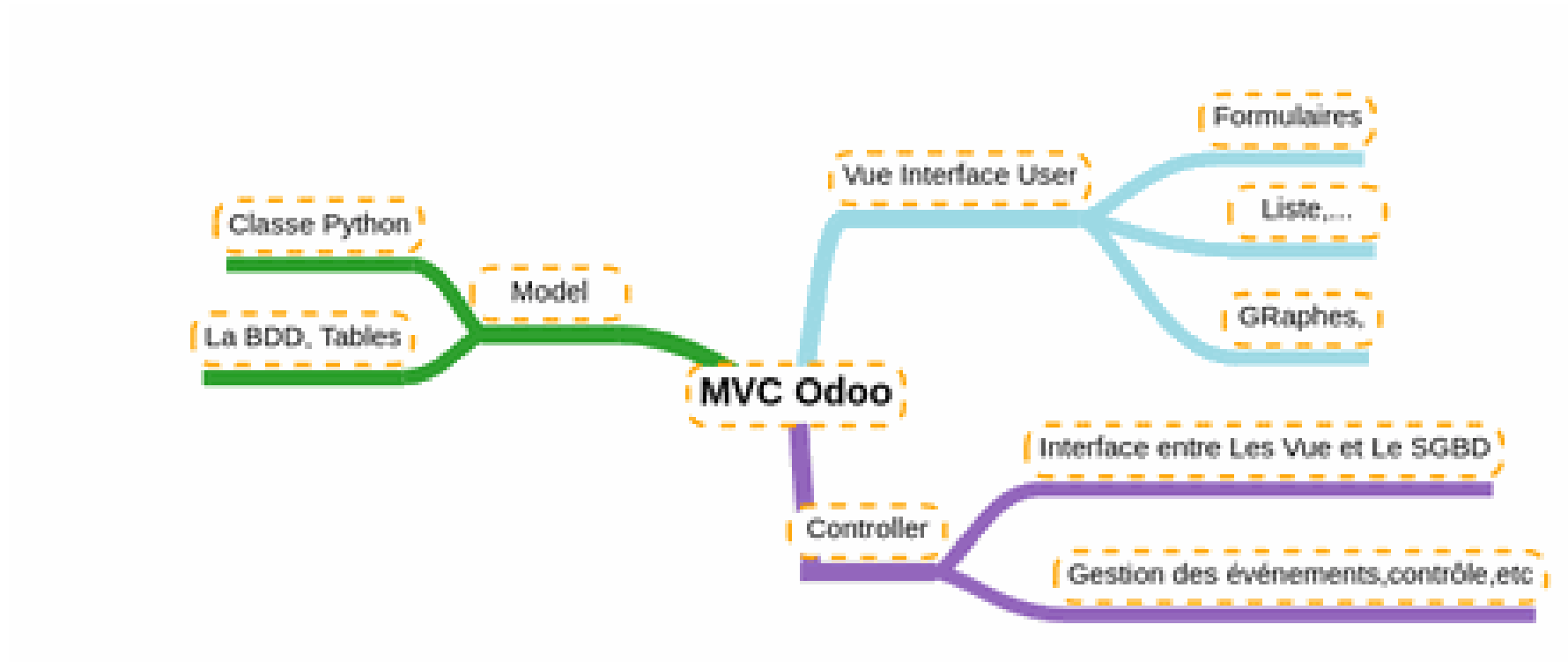
# Architecture Technique

○ L'architecture du système Odoo est composée de 3 tiers :

- **Un serveur de base de données PostgreSQL.**
- • **Un serveur d'application** (contenant les objets de gestion, le moteur de workflow, le générateur d'édition, etc.).
- **Un serveur de présentation** qui permet à l'utilisateur de se connecter à Odoo avec n'importe quel navigateur internet( Google chrome, firefox,...)



# Le Modèle de Conception MVC



# Le Modèle de Conception MVC

Le Framework technique d'odoo est basé sur le modèle de conception MVC (Modèle Vue Contrôler) qui consiste à décomposer l'architecture du programme en trois parties complémentaires et dépendantes :

- **Modèle** : qui contient toutes les objets odoo déclarés, les tables de données : La partie base de données est gérée par le SGBD PostgreSQL
- **Vue** : c'est les différentes vues et interfaces utilisés par l'utilisateur (Vue Formulaire, kanban view, Calendar, ....) , cette partie est gérée principalement par XML
- **Contrôleur** : C'est la partie contrôle qui consiste aux codes python exécutés pour assurer le contrôle , la synchronisation et la gestion des événements.

# Architecture des Dossiers et modules Odoo

Le principe de base d'odoo est d'utiliser une structure modulaire, et indépendante qui permet à la fois d'améliorer régulièrement les modules existants, et d'Autre part, avoir une souplesse de modification ou de suppression des modules qu'on ne veut pas utiliser, sans avoir à toucher tout le système.

Par défaut, il contient plusieurs modules prêt à être déployés tel quels , ou à les personnaliser selon vos besoins. Les modules sont contenu dans un répertoire nommé / addons .

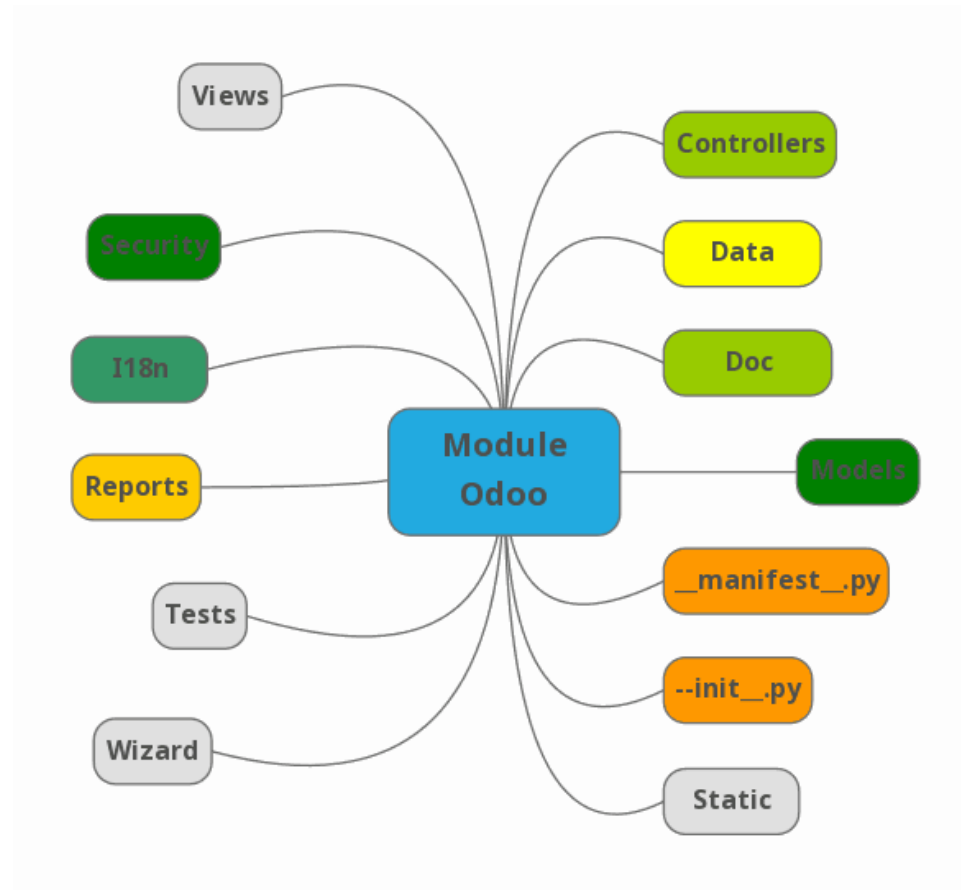


## FOLDERS

- Odoo 13.0
  - nssm
  - python
  - server
    - .github
    - addons
    - debian
    - doc
    - odoo
      - \_\_pycache\_\_
      - addons
        - \_\_pycache\_\_
        - account
        - account\_analytic\_default
        - account\_analytic\_default\_hr\_expense
        - account\_analytic\_default\_purchase
        - account\_bank\_statement\_import
        - account\_check\_printing
        - account\_debit\_note
        - account\_facturx
        - account\_lock
        - account\_payment
        - account\_tax\_python
        - account\_test
        - analytic
        - association
        - attachment\_indexation
        - auth\_ldap
        - auth\_oauth
        - auth\_password\_policy
        - auth\_password\_policy\_signup

```
<<  _manifest_.py  x
1  # -*- coding: utf-8 -*-
2  # Part of Odoo. See LICENSE file for full copyright and licensing details.
3  {
4      'name' : 'Invoicing',
5      'version' : '1.1',
6      'summary': 'Invoices & Payments',
7      'sequence': 15,
8      'description': """
9  Invoicing & Payments
10  =====
11  The specific and easy-to-use Invoicing system in Odoo allows you to keep track of your accounting, even whe
12
13  You could use this simplified accounting in case you work with an (external) account to keep your books, an
14  """,
15      'category': 'Accounting/Accounting',
16      'website': 'https://www.odoo.com/page/billing',
17      'images' : ['images/accounts.jpeg', 'images/bank_statement.jpeg', 'images/cash_register.jpeg', 'images/cha
18      'depends' : ['base_setup', 'product', 'analytic', 'portal', 'digest'],
19      'data': [
20          'security/account_security.xml',
21          'security/ir.model.access.csv',
22          'data/data_account_type.xml',
23          'data/account_data.xml',
24          'data/digest_data.xml',
25          'views/account_menuitem.xml',
26          'views/account_payment_view.xml',
27          'wizard/account_accrual_accounting_view.xml',
28          'wizard/account_unreconcile_view.xml',
29          'wizard/account_move_reversal_view.xml',
30          'views/account_move_views.xml',
31          'wizard/setup_wizards_view.xml',
32          'wizard/pos_box.xml',
33          'views/partner_view.xml',
34          'views/account_view.xml',
```

# Structure d'un Module odoo :



# Structure d'un Module odoo :

Un Module sous odoo est représenté comme un répertoire, contenant des sous-répertoires et des fichiers avec une convention bien définie.

Pour pouvoir créer et développer ses propres modules odoo, il va falloir comprendre la signification des fichiers et répertoire d'un module odoo, composant l'architecture technique d'un module

# Structure d'un Module odoo :

1. **\_\_init\_\_.py** : (Required : nécessaire pour la création d'un module)

C'est le fichier python d'initialisation du module contenant tous les autres fichiers python à importer .

2. **\_\_manifest\_\_.py** :— (Required : nécessaire pour la création d'un module)

C'est le fichier de manifestation du module , il contient tous les informations concernant le module comme le nom, une description, la version, l'auteur, le site web, les fichiers de données, démos, sécurité ,....

# Structure d'un Module odoo :

**3. models :—(Required :** nécessaire pour la création d'un module)

C'est un répertoire qui va contenir tous les fichiers pythons de votre modules, ce répertoire doit être déclaré dans le fichier d'initialisation `__init__.py`

**4. views :—**

C'est le répertoire qui va contenir les fichiers xml décrivant la couche présentation ou interface, tel que les vues tree(liste), formulaires (form), graph, pivot, kanban , action, menu, rechercher (search),....

**5. data:—**C'est un répertoire qui va contenir les données prédéfinies nécessaire lors de l'installation du module .

**6. demo:—**Ce répertoire va contenir les données de démonstrations, qui vont être chargé lors de l'installation du module.

# Structure d'un Module odoo :

**7. static** :—Ce répertoire va contenir tous les fichiers relié au site web, tel que les fichiers img, js (javascript), css, font ,...

## (1) src

**css**:— contient tous les fichiers css de conception

**img**:— contient tous les fichiers images

**js**:— contient tous les fichiers javascript

**xml**:— contient tous les fichiers .xml utilisé pour les view/qweb templates

**(2). description**:—Contient un fichier html nommé “Index.html” qui permet une démonstration et présentation sur votre module , contenant des aperçus et des imprimés écrans des fonctionnalités de votre module .Il contient aussi le fichier icon.png qui sera utilisé comme l’icône de votre module .

# Structure d'un Module odoo :

**8. wizard** :— C'est un répertoire qui va contenir les classes de transitions nécessaire à l'affichage des assistants et des fenêtres qui servent à aider l'utilisateur à introduire les données nécessaires pour l'obtention des résultats souhaités (états de sorties, calcul, ...) Ces modèles seront automatiquement supprimés après utilisation.

**9. Report** :- C'est le répertoire qui va contenir les descriptions qweb et xml, des rapports de sortie (en pdf) de votre module

**10. security** :- Répertoire contenant les fichiers de base définissant les règles de sécurité, les droits d'accès, les groupes, etc.

**Le fichier csv** des droits d'accès (ir.model.access.csv) : pour assigner les droits de lecture et d'écriture , à un utilisateur d'un groupe .

**security.xml** : qui va contenir la création des groupes et les autres règles d'accès aux enregistrements .

# Structure d'un Module odoo :

**11. controller** :— Ce répertoire va contenir les fichiers nécessaires des requêtes http et leurs réponses à partir d'un navigateur internet , il concerne la partie développement web qui permet d'étendre ou personnaliser le module de base website proposé par odoo

**12. i18n** :- Il comprend la traduction de votre module dans les différents langues .

principalement , il contient le fichier template de traduction (.pot) et le fichier .po qui contient la traduction actuelle du module

**13. Doc** :- C'est un répertoire qui va contenir la partie documentation concernant votre module les fonctionnalités, les explications et démos,..)

**14. Test** :- C'est un répertoire qui va contenir les fichiers .py (python) de test.



# Écosystème technique odoo :

La communauté odoo, est l'une des facteurs importante de progression et du développements d'odoo.

Composée de développeurs, utilisateurs et partenaires, a fait exploser l'utilisation d'odoo dans le monde entier, en quelques années seulement.

Il est nécessaire donc de savoir comment utiliser la communauté pour se compléter en terme de compétences et se former régulièrement.

# Installation & configuration

# Cas d'étude

Entreprise de vente des Meubles de bureau

Produits ( Tables, Chaises, Bureau)

- Achat des produits d'un ( ou plusieurs fournisseur)
- Stockage des produits dans le warehouse
- Vente des Produits aux clients avec Facture

# Procédure d'achat - Fournisseur

- I. Demande de Prix
- II. Réception des Devis
- III. Validation de la commande
- IV. Réception des produits et validation
- V. Réception de facture
- VI. Règlement

# Procédure De vente - Client

- I. Réception de demande de prix
- II. Envoie du Devis
- III. Réception des Bon de Commande
- IV. Livraison des Produits avec le bon de livraison
- V. Facturation du client
- VI. Règlement Client

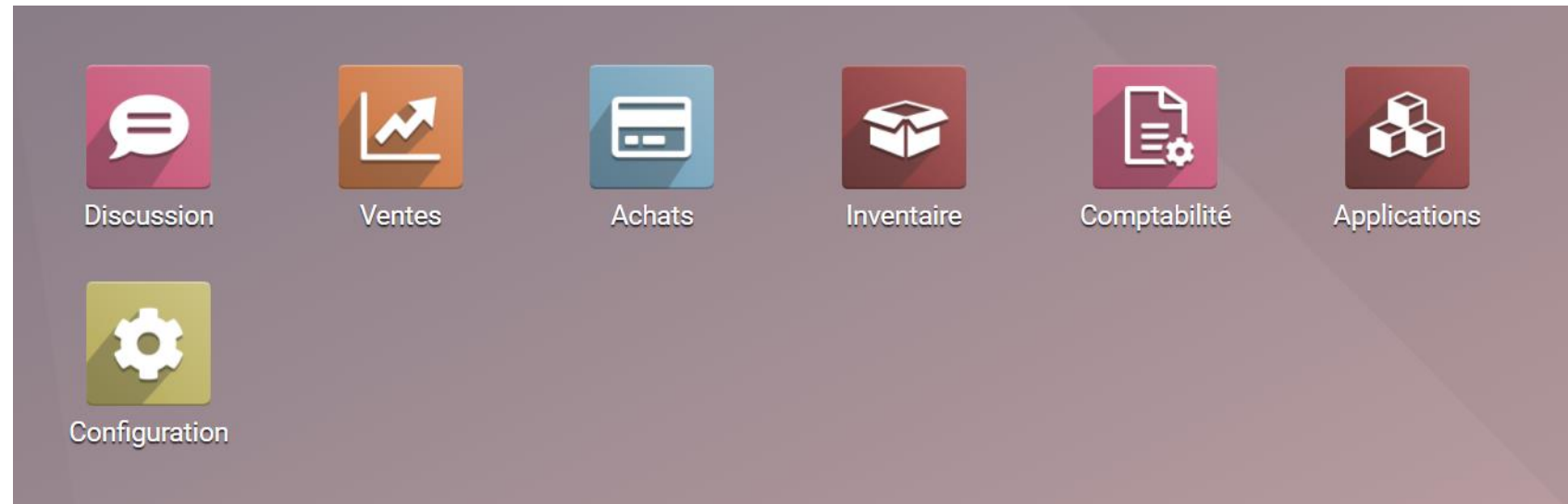
# Processus d'Inventaire

La société a un seul « Warehouse » afin de stocker les produits reçus des fournisseurs et les livrer aux clients

# Installation & configuration

# Liste des Modules à installer

- Vente
- Facturation
- Comptabilité
- Inventaire
- Achat





Changement de mot de passe