

Cours 7 : Les documentations logicielles

Utilisateurs, techniques, marketing...

Documentation logicielle

□ Définition

- texte écrit qui accompagne le logiciel informatique
- Elle explique comment le logiciel fonctionne, et/ou comment on doit l'employer.

□ Remarque

- La documentation constitue une partie importante de l'ingénierie logicielle, qui est trop souvent négligée

Type de documentations

❑ Expression de besoin

- Décrit l'ensemble des besoins du client (fonctionnel, ergonomique, performance ...)

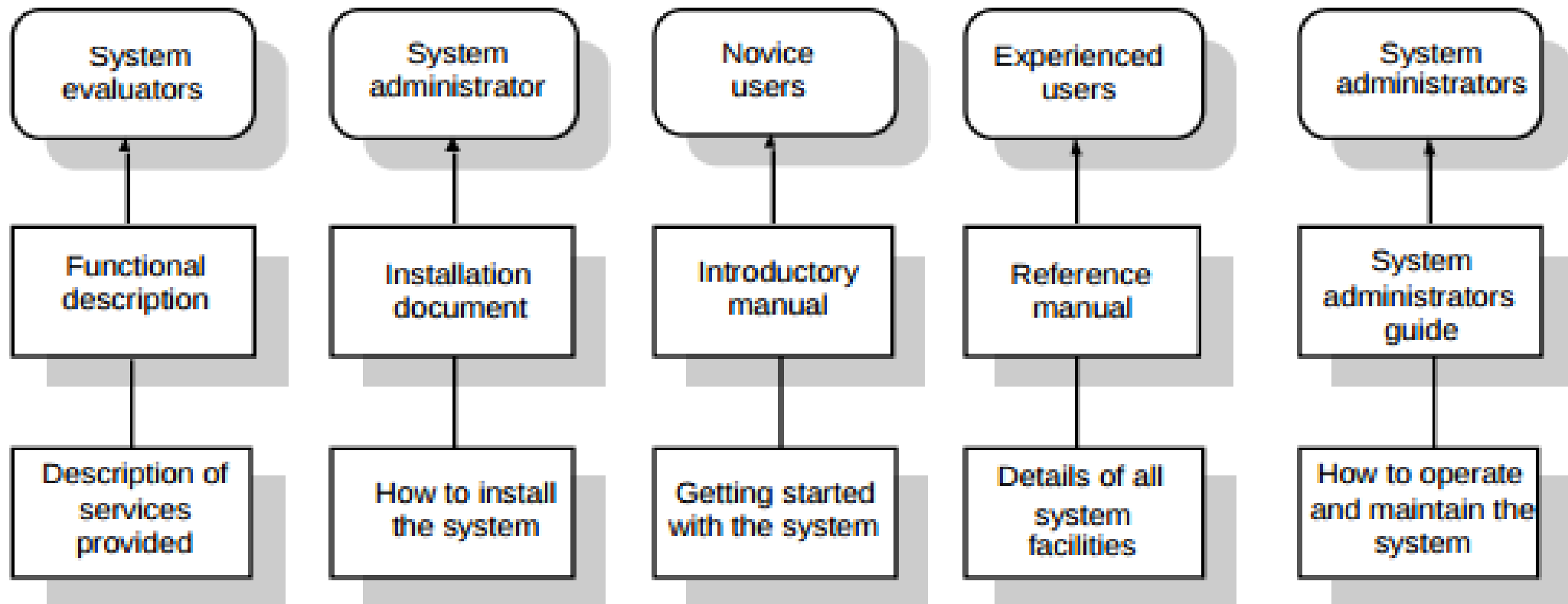
❑ Technique

- Pour les développeurs : Documentation du code, algorithmes, interfaces, et interfaces de programmation (API).
- Pour les architectes : Vue d'ensemble du système d'information. Elle inclut les relations à l'environnement
- Pour les administrateurs systèmes et personnel de support.

❑ Utilisateur

- Manuels pour les utilisateurs, Marketing
- Instructions sur le produit et garantie promotionnelle.

Les différents documents et leurs usages



[Ian Sommerville 2001]

Type de supports

- ❑ **Papier**
- ❑ **HTML**
- ❑ **Vidéo**
- ❑ **Logiciel de démonstration pas à pas (mode démo)**

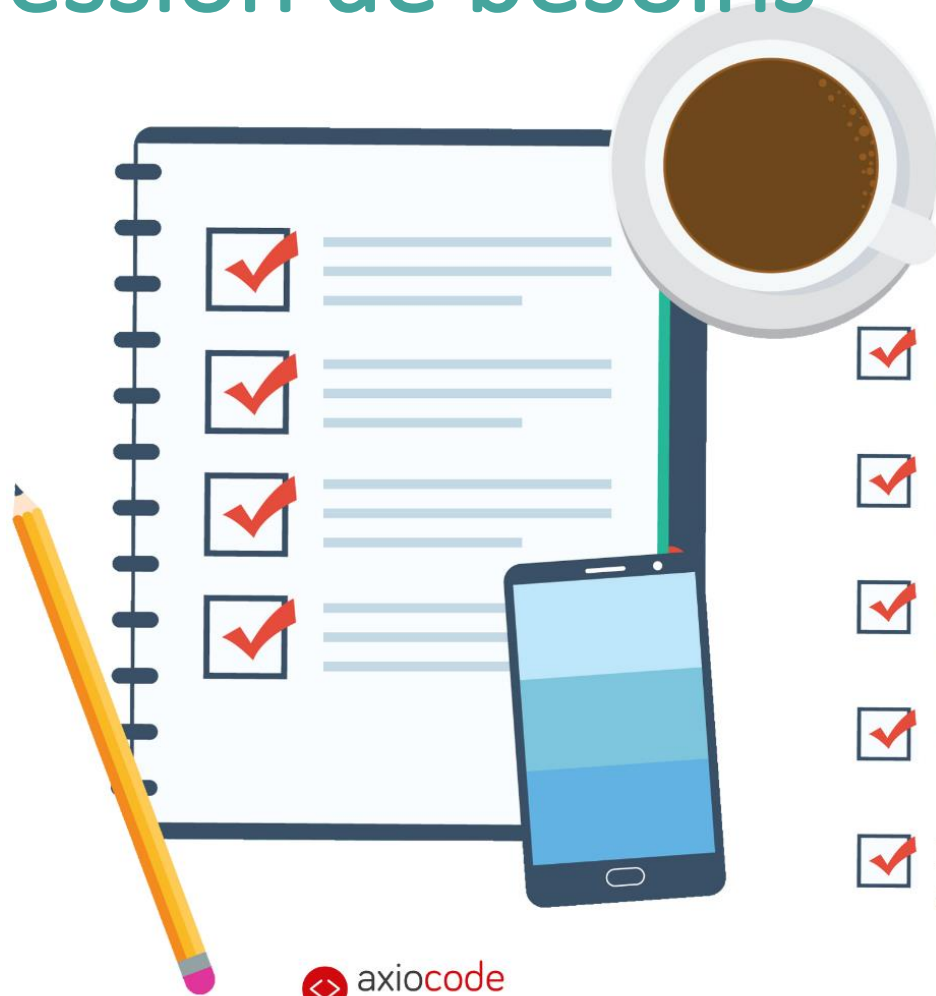
Expression de besoins

Expression de besoins

❑ **Document de spécification des exigences du logiciel**

- Répond à la question « Que doit faire le logiciel ? ».
- Contient des exigences
 - ❑ Reprise d'un existant
 - ❑ Connexion avec d'autres SI
 - ❑ Environnement logiciel

Expression de besoins



Les 10 Commandements d'une Expression de Besoins Efficace

- ☒ Soyez précis : ne laissez pas place à l'interprétation
- ☒ Ecrivez des phrases courtes, ne dépassez pas 5 pages
- ☒ N'anticipez pas la solution et la conception technique
- ☒ Priorisez les points importants
- ☒ Faites des références à des docs existants pour illustrer
- ☒ Listez les fonctionnalités, n'entrez pas dans les détails
- ☒ Décrivez les utilisateurs de votre application
- ☒ Fixez les limites de votre projet : objectifs, ressources, contexte...
- ☒ Précisez les évolutions de périmètre ou fonctionnelles
- ☒ Spécifiez les contraintes et les impacts liés à votre projet

Différence expression de besoins et cahier des charges

❑ Expression de besoins

- rédigée avant le cahier des charges
- Sert de base au cahier des charges

❑ différences

- Une expression de besoins est
 - ❑ non contractuelle.
 - ❑ concise contrairement au cahier des charges qui décrit le plus précisément possible le fonctionnement de votre application
 - ❑ reste ouverte aux changements et aux ajouts de “fonctionnalités”. L’expression de besoins permet à chacun d’apporter des suggestions qui seront conservées ou non dans le cahier des charges fonctionnel.
 - ❑ simple. N’indiquez jamais la solution pour réaliser une fonctionnalité.

Documentations utilisateur

A solid teal-colored horizontal bar spanning the width of the slide, featuring a central V-shaped notch.

Documentations utilisateur

❑ Objectif

- Une documentation utilisateur, dans le cadre de l'utilisation des outils numériques, doit faciliter l'utilisation d'une solution applicative et permettre à l'utilisateur cible de se familiariser avec les fonctionnalités proposées pour atteindre un objectif donné.

❑ Types

- *mode d'emploi*
- guide d'installation,
- guide de démarrage rapide,
- aide au paramétrage,
- fiche fonctionnalités,
- aide mémoire,
- pas à pas
- ensemble de recommandations.

Documentations utilisateur

❑ la documentation utilisateurs

- éloignés du code source du programme
- Décrit chaque caractéristique du programme, et les différentes étapes nécessaires pour l'appeler.
- peut aussi aller jusqu'à fournir une assistance minutieuse en ligne.

❑ Qualité

- pas confus
- à jour
- Être indexé précisément.
- Cohérent
- simple

Documentations utilisateur

- C'est un contrat qui spécifie ce que le logiciel doit faire.
- **Grandes manières d'organiser la documentation utilisateur**
 - Tutoriel
 - Thématique
 - Liste
 - Aide en ligne
- Ces approches sont complémentaires

Mode d'emploi

❑ Classiquement les produits (matériel ou logiciel) sont accompagnés de mode d'emploi

- Avant : format papier, unique, long et détaillé
 - ❑ En France et ailleurs, on ne lit pas les modes d'emploi.
- Maintenant : court, segmenté, et multi-format
 - ❑ Remarque sur les logiciels actuels
 - recherche d'une approche intuitive qui demande peu ou pas de formation
 - Livré sans mode d'emploi

Gigaset A140/Gigaset A240 français

Combiné Gigaset A14

Base

Combiné Gigaset A24

1 Etat de charge du bloc de batteries
2 Numéro interne du combiné
3 Touches écran
4 Touche Décrocher/ Touche Mains-libres (uniquement Gigaset A240)
5 Touche Répertoire
6 Touche Raccrocher et Marche/Arrêt
7 Sonnerie activée/désactivée (appuyer longtemps)
8 verrouillage clavier (appui long)
9 Touche R
- Double appel (Flash)
- Pause interchiffres (appuyer longtemps)
10 Microphone
11 Touche répertoire se-

1 2 3 4 5 6 7 8 9 10 11

SIEMENS

22-35

→ INT MENU

Touche Déclaration/Paging (p. 5/p. 6)

Touches écran
Elles permettent d'appeler l'une des 4 fonctions dont les symboles sont affichés, selon le contexte, en regard au bas de l'écran.

Affichage	Signification
→	Ouvrir la liste des numéros bis (p. 4).
✉	Ouvrir le journal d'appels et de messages (p. 4).
INT	Interne : Ouvrir la liste des combinés inscrits (p. 5).
MENU	Ouvrir le menu principal/sous-menu.
↑ ou ↓	Naviguer vers le haut ou vers le bas ou régler le volume.
← ou →	Déplacer le curseur vers la gauche ou la droite.
↶	Effacer caractère par caractère vers la gauche.
OK	Activer la fonction du menu ou enregistrer l'entrée en la validant.

Consignes de sécurité

⚡ Avant l'utilisation, lire attentivement le mode d'emploi et les consignes de sécurité. Informez vos enfants sur les recommandations qui y figurent et mettez-les en garde contre les risques liés à l'utilisation du téléphone.



Utiliser **uniquement** le bloc-secteur fourni.



Insérez uniquement le **bloc de batteries rechargeables homologué**. Donc, ne jamais utiliser des piles normales (non rechargeables) qui peuvent endommager le combiné, représenter un risque pour la santé et provoquer des blessures. Utilisez le bloc de batteries en respectant ce mode d'emploi (p. 7).



Le combiné peut perturber le fonctionnement de certains équipements médicaux. Tenir compte des recommandations émises à ce sujet par le corps médical.



Le combiné peut provoquer un bourdonnement désagréable dans les aides auditives. De façon générale, ne pas placer d'appareil électronique à proximité immédiate du téléphone. Pendant la sonnerie (arrivée d'un appel), ne pas tenir le combiné directement collé à l'oreille.



Ne pas installer la base dans une salle de bains ou de douche (p. 7). Le combiné et la base ne sont pas étanches aux projections d'eau.



Ne pas utiliser le téléphone dans un environnement présentant un risque d'explosion (par exemple ateliers industriels de peinture).



Lorsque vous confiez votre Gigaset à un tiers, veuillez toujours à joindre le mode d'emploi.



Respectez scrupuleusement les règles de protection de l'environnement lorsque vous mettez au rebut le téléphone et les batteries. Ne pas jeter ces dernières au feu ni les ouvrir ou les mettre en court-circuit.

Lorsque le verrouillage du clavier est activé (p. 3), même les numéros d'appel d'urgence ne peuvent être composés ! Toutes les fonctions décrites dans ce mode d'emploi ne sont pas nécessairement disponibles dans tous les pays et sur tous les réseaux. De même, elles peuvent être assujetties à des abonnements de services à souscrire spécifiquement.

Tutoriel (tuto) et didacticiel, HowTo

- ❑ **Pour un nouvel utilisateur**
- ❑ **Pour une tâche précise**
 - Dans cette méthode l'utilisateur est guidé à chaque étape d'accomplissement des tâches particulières.
 - « **Pousse bouton** »
- ❑ **Format papier, vidéo, logiciel**

Manuel procédural (How to , tuto)

❑ **Un manuel de type procédural**

- Manuel pas à pas.
- N'explique pas la totalité d'un logiciel
- Explique des utilisations précises

❑ **Pour chaque utilisation précise**

- Une explication à partir du début
- Très illustré
 - ❑ Copie d'écran avec légende, vidéo,
- Simple

❑ **Destinataire**

- Utilisateur occasionnel qui n'a pas la maîtrise d'un logiciel complexe

Exemple de tutoriel

- ❑ **Comment construire un diagramme de cas d'utilisation ?**
 - <https://www.youtube.com/watch?v=J8NtoLxhoRc>
 - Du général au particulier
 - Un exemple fil rouge qui permet d'illustrer les concepts décrits.
 - ❑ Simple et bien compris par tous
 - Vocabulaire précis et bien défini



Documentations Thématique

- ❑ **Pour un utilisateur intermédiaire,**
- ❑ **les chapitres ou sections se concentrent sur un domaine d'intérêt particulier**
 - Exemples
 - ❑ Le langage de macro d'Excel
 - ❑ Les images avec transparence (Gimp)
- ❑ **Un ensemble de documents thématiques peuvent former un manuel**

Liste et aide en ligne

□ Liste

- pour des utilisateurs avancés
 - qui connaissent exactement quelle sorte d'information ils recherchent
- les commandes ou les tâches sont simplement listées par ordre alphabétique,
- via des indices croisés

□ Aide en ligne ; touche F1

- des informations de référence sur les commandes ou les lignes de menu

FAQ

❑ **FAQ : frequently asked questions foire aux questions**

- liste de questions faisant la synthèse des questions posées de manière récurrente
 - ❑ Accompagnée des réponses correspondantes,
- Objectifs
 - ❑ éviter que les mêmes questions soient toujours repostées, et d'avoir à y répondre constamment.
 - ❑ un seul couple question / réponse ;

Base de connaissances, wiki

❑ Une base de connaissances sous la forme de wiki autour d'un logiciel est aussi une bonne documentation

▪ Wiki

- ❑ Application web qui permet la création, la modification et l'illustration collaboratives de pages
- ❑ la structure implicite est minimale,
- ❑ tandis que la structure explicite émerge en fonction des besoins des usagers
- ❑ Associé à un moteur de recherche

Manuel pédagogique, Cours et TD

❑ Objectif

- Présenter le logiciel de manière générale
- Acquérir des compétences de plus en plus détaillées
 - ❑ ordre du plus simple et plus utile vers le plus complexe et le moins utilisé
 - ❑ Respect de pré-requis

❑ Inclus

- Des exemples
- Des exercices d'application

❑ Exemple d'un logiciel de traitement de données,

1. chargement des données,
2. traitements simples fréquents
3. sauvegarde
4. impression du résultat
5. Traitements plus complexes...

❑ Format papier ou vidéo

- Formation en ligne
 - ❑ Vidéo exemple : www.khan-academy.fr

Manuel de référence

- ❑ **Ce type de manuel se veut exhaustif. On va le parcourir de A à Z. L'organisation suit alors plus la logique du concepteur que celle de l'utilisateur : on liste les sous-ensembles un par un.**
 - Par exemple, dans le cas d'un logiciel, on va prendre les barres de boutons une par une et décrire chacun des boutons, puis les menus un par un et décrire chacune des options, puis les boîtes de dialogue, ...

Documentations techniques

A solid teal-colored horizontal bar spanning the width of the slide, positioned below the title. It features a central, symmetrical notch or cutout.

Documentations techniques

□ utilisé pour définir et expliquer

- les interfaces de programmation (APIs (*Application Programming Interface*)),
- les structures de données et les variables
- les algorithmes
- Les évolutions

□ Pour qui

- Pour les **développeurs** qui va reprendre le code
- Pour **l'installation** qui va déployer le SI
- Pour **la maintenance** qui va devoir procéder à des maintenances corrective et évolutive

Documentations techniques

❑ Recommandation

- écrire la documentation en même temps et en un même lieu que le code source et à l'extraire par des moyens automatiques.
- Maintenir la documentation à jour en permanence



CommitStrip.com

Exemple de commentaires

❑ Méthode **Variation**

❑ Prend en entrée une suite de valeurs flottantes, et un pourcentage.

- Renvoie un triplet.
 - ❑ le nombre de couples de valeurs consécutives où il y a une décroissance.
 - ❑ le nombre de couples où les valeurs sont stables (variation est inférieure au pourcentage passé en paramètre.
 - ❑ le nombre de couples où les valeurs sont croissantes.

❑ Recommandations

- Noms des variables adaptés
- Création de variables intermédiaires
- Explication des instructions complexes
- Expliquer les cas limites

Exemple de commentaires

```
public static int[] vari(double[] tab, double pourcentage) {  
    int[] resu = {0,0,0}; // nb décroissant, nb stable, nb croissant  
    if(pourcentage<0)      {  
        System.out.println("erreur pourcentage negatif");  
        int[] erreur = {-1,-1,-1};  
        return erreur;    }  
    for (int i=0; i<tab.length-1; i++)      {  
        double borneMin = tab[i]*(1-pourcentage); // la valeur moins le pourcentage  
        double borneMax = tab[i]*(1+pourcentage); // la valeur plus le pourcentage  
        // des bornes ont été utilisées pour éviter les divisions par 0  
        // en effet (tab[i+1]- tab[i]) / tab[i] pose problèmes si tab[i] vaut 0  
        if (tab[i+1]<borneMin)  
            resu[0]++; // décroissant  
        else  
            if (tab[i+1]>borneMax)  
                resu[2]++; // croissant  
            else  
                resu[1]++; //stable  
        }  
    return resu;  
}
```

Documentations techniques

❑ les générateurs de documentations

- Exemple : Doxygen ou JavaDoc
 - ❑ générer automatiquement le document sur le code
 - À partir des commentaires du code source
 - ❑ Organisé sous la forme d'un guide de référence,

JavaDoc



- ❑ **Outil du langage Java**
- ❑ **Basé sur des commentaires spécifiques**
 - commencent par `/**` et se termine par `*/`.
 - Contiennent un texte libre et des balises particulières.
- ❑ **Balises @**
 - Décritent des caractéristiques des classes et des méthodes
- ❑ **Génère des fichiers HTML**
 - Organisé sous la forme d'une arborescence
 - ❑ Projet > package > classe > méthode

Les commentaires et les tags principaux

❑ Le commentaire

- Commence par une description de l'élément décrit (classe, méthode...)
- Possibilité d'utiliser des balises HTML
- Est suivi de tags

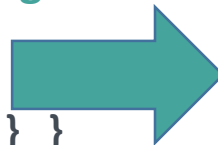
❑ Les tags

- `@author` le ou les auteurs de l'élément
- `@deprecated` précise qu'un élément est déprécié
- `@param` permet de documenter un paramètre de l'élément
- `@return` permet de fournir une description de la valeur de retour
- `@see` permet de préciser un élément en relation avec l'élément documenté
- `@since` permet de préciser depuis quelle version l'élément a été ajouté
- `@version` permet de préciser le numéro de version de l'élément
- `@exception` nom_exception description d'une exception similaire à `@throws`

```
/**
 * Returns an Image object that can then be painted on the screen.
 * The url argument must specify an absolute {@link URL}. The name
 * argument is a specifier that is relative to the url argument.
 * <p>This method always returns immediately, whether or not the
 * image exists. When this applet attempts to draw the image on
 * the screen, the data will be loaded. The graphics primitives
 * that draw the image will incrementally paint on the screen.
 * @param url an absolute URL giving the base location of the
 * image
 * @param name the location of the image, relative to the url
 * argument
 * @return the image at the specified URL
 * @see Image
 */
```

```
public Image getImage (URL url, String name) {
    try { return getImage(new URL(url, name));
    } catch (MalformedURLException e) { return null; } }
```

Fichier
HTML
généré



getImage

```
public Image getImage(URL url,
    String name)
```

Returns an Image object that can then be painted on the screen. The url argument must specify an absolute URL. The name argument is a specifier that is relative to the url argument.

This method always returns immediately, whether or not the image exists. When this applet attempts to draw the image on the screen, the data will be loaded. The graphics primitives that draw the image will incrementally paint on the screen.

Parameters:

url - an absolute URL giving the base location of the image.

name - the location of the image, relative to the url argument.

Returns:

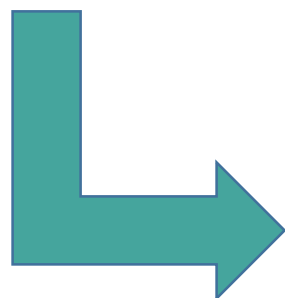
the image at the specified URL.

See Also:

[Image](#)

JavaDoc deuxième exemple

```
01. /**
02.  * Résumé du rôle de la méthode.
03.  * Commentaires détaillés sur le role de la methode
04.  * @param val la valeur a traiter
05.  * @return la valeur calculée
06.  * @since 1.0
07.  * @deprecated Utiliser la nouvelle methode xyz
08.  */
09. public int maMethode(int val) {
10.     return 0;
11. }
```



maMethode

public int **maMethode**(int val)

Deprecated. *Utiliser la nouvelle methode xyz*

Résumé du rôle de la méthode. Commentaires détaillés sur le role de la methode

Parameters:

val - la valeur a traiter

Returns:

la valeur calculée

Since:

1.0

<https://www.jmdoudoux.fr/java/dej/chap-javadoc.htm>

Documents des tests

□ **Description**

- Ces documents accompagnent les tests
 - Intégration
 - Validation
- Regroupe les résultats obtenus

Documents d'architecture

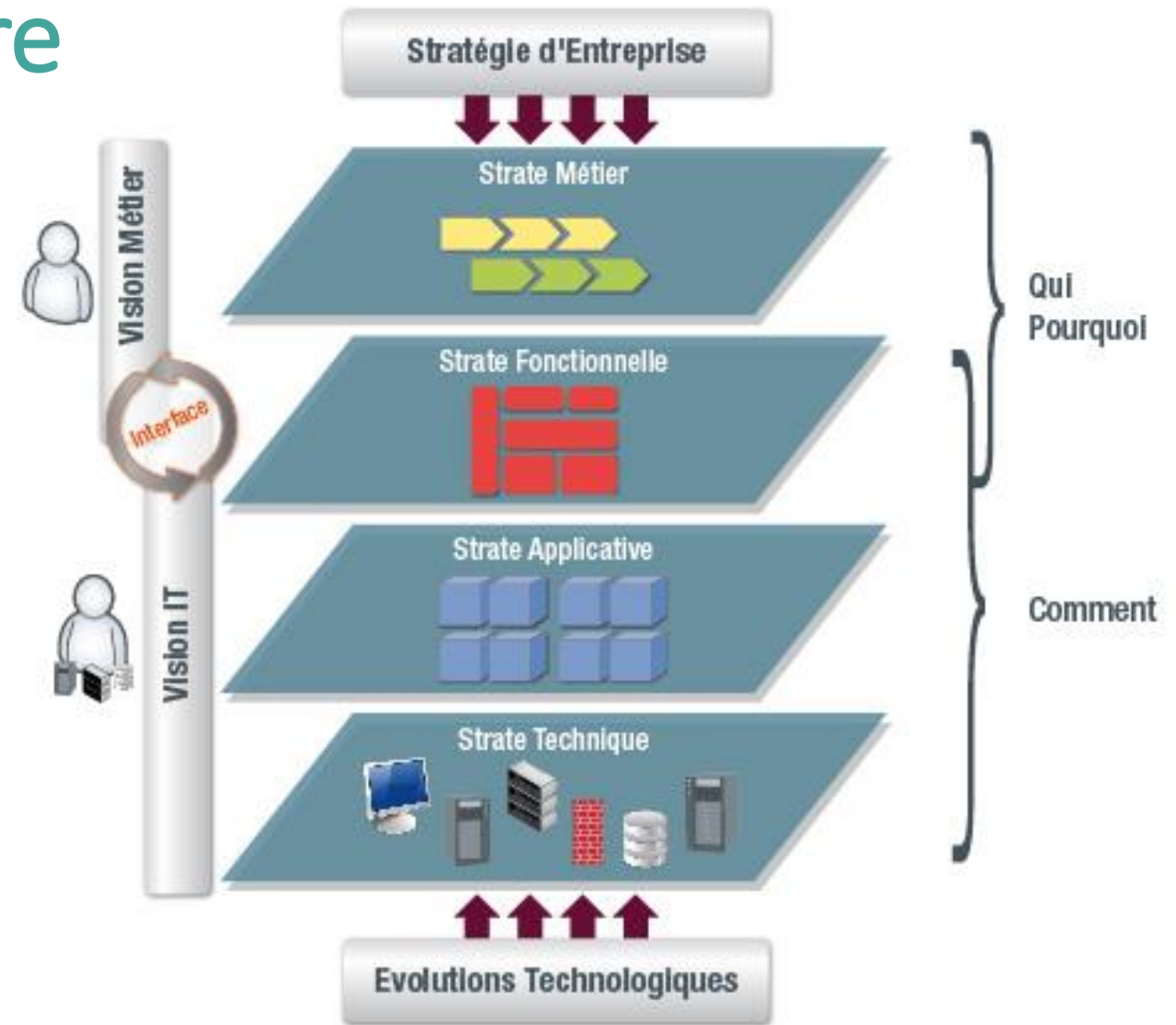
Strates de l'architecture d'un SI

□ 4 strates (couches) selon le CIGREF

(<https://www.cigref.fr/>)

1. Métier
2. Fonctionnelle
3. Logiciel ou applicative
4. Technique

■ Chacune donne une vision différentes du SI

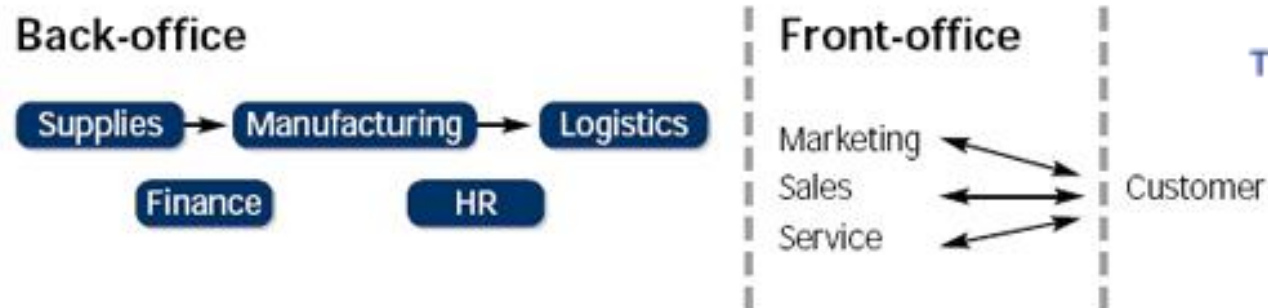


Architecture métier (1/4)

- ❑ **Définit les processus métiers, à l'exclusion de toute technicité**
- ❑ **décrit**
 - les applications informatiques,
 - les principales bases de données du système informatique d'une institution ou d'une entreprise,
 - Les utilisations faites de ces éléments dans le cadre de l'activité de l'institution et leur alignement à l'organisation générale de l'institution.

Exemple de diagrammes d'architecture Métier

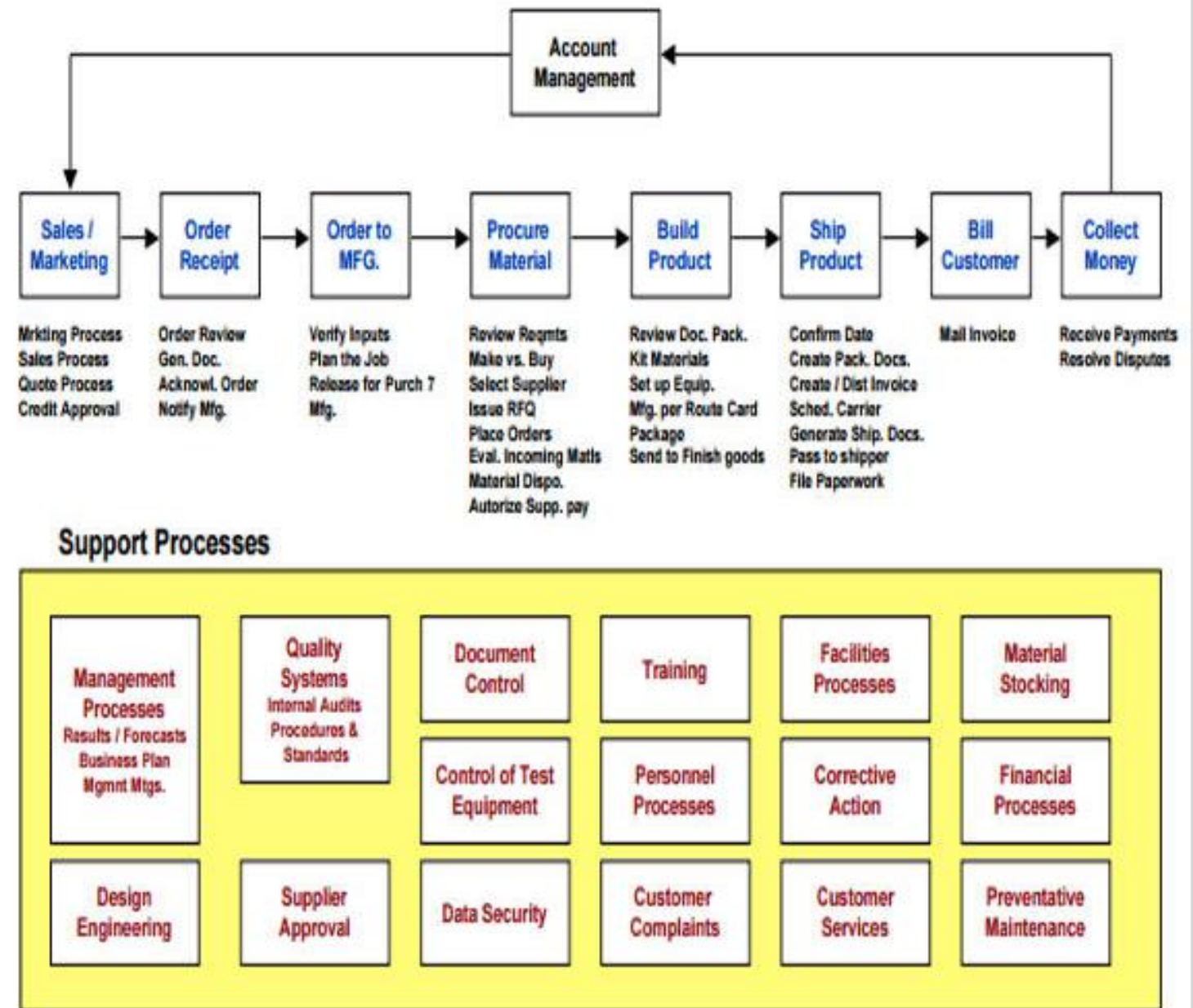
- ❑ **CRM** (customer relationship management) avec une base de données centralisée regroupant l'ensemble des informations sur les clients et les applications clients



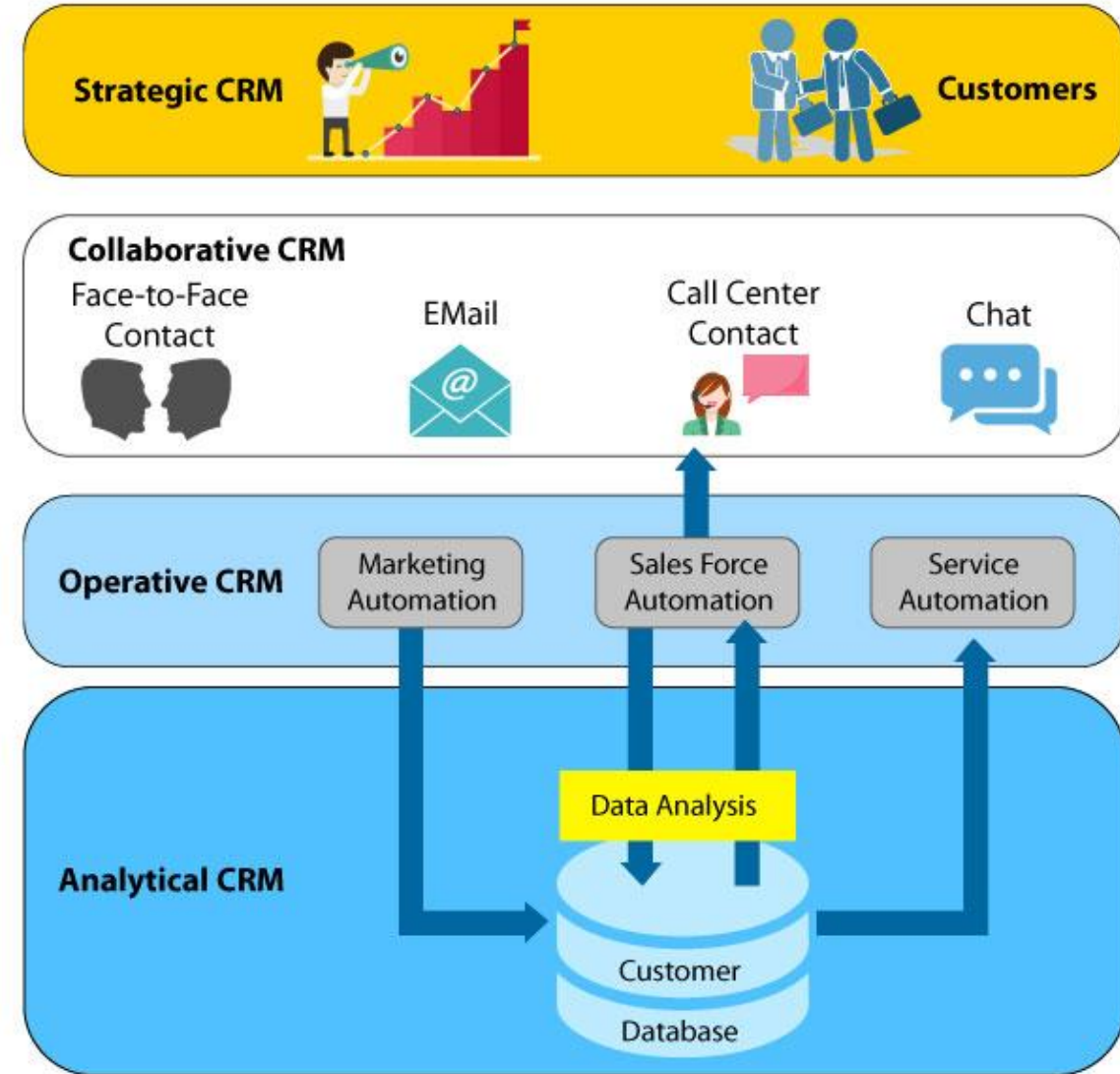
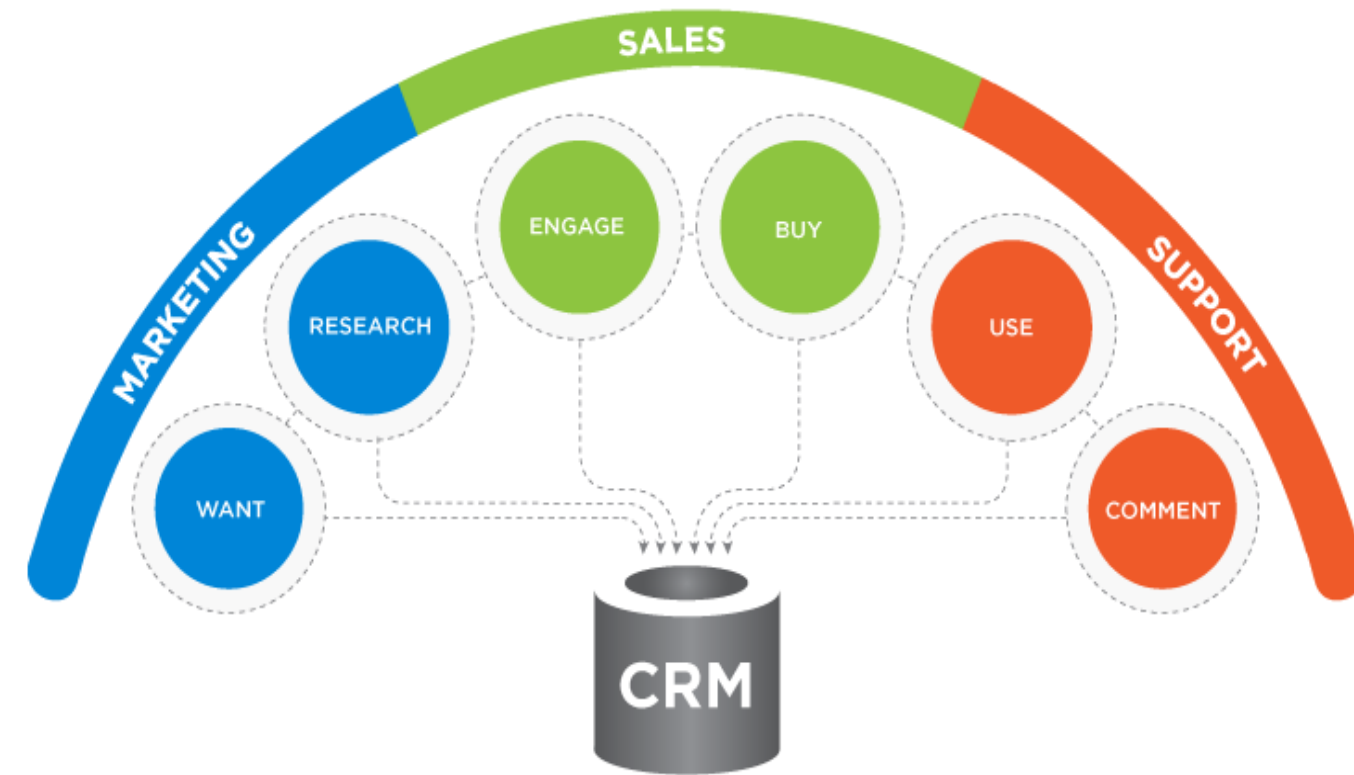
<http://jisc.cetis.ac.uk/crm-tools/what-is-crm.html>

Exemple de diagramme d'architecture Métier

- ❑ <http://antyradar.info/company-flow-chart/company-flow-chart-6-company-flow-chart-templates-6-free-word-pdf-format/>

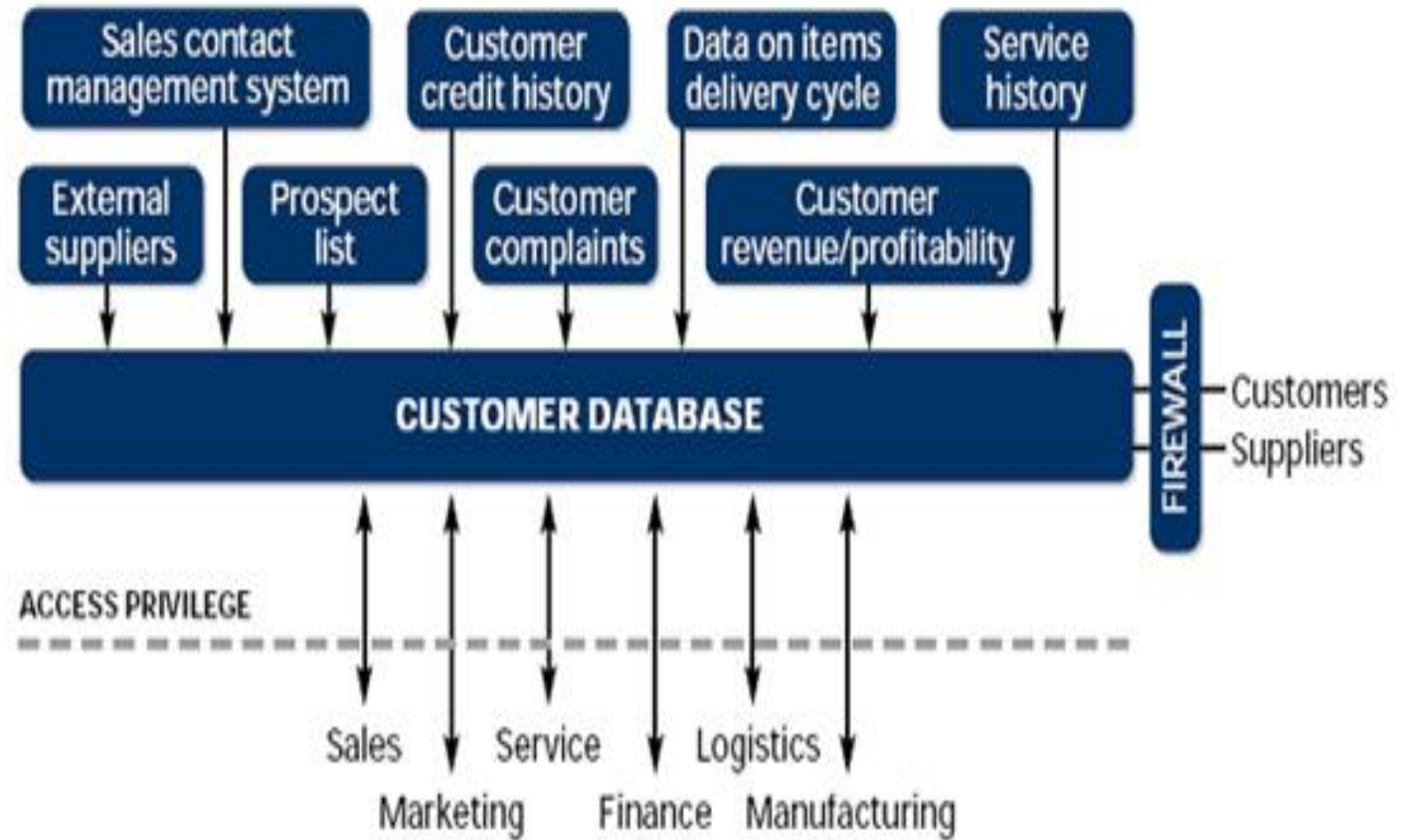


Exemple architecture Métier



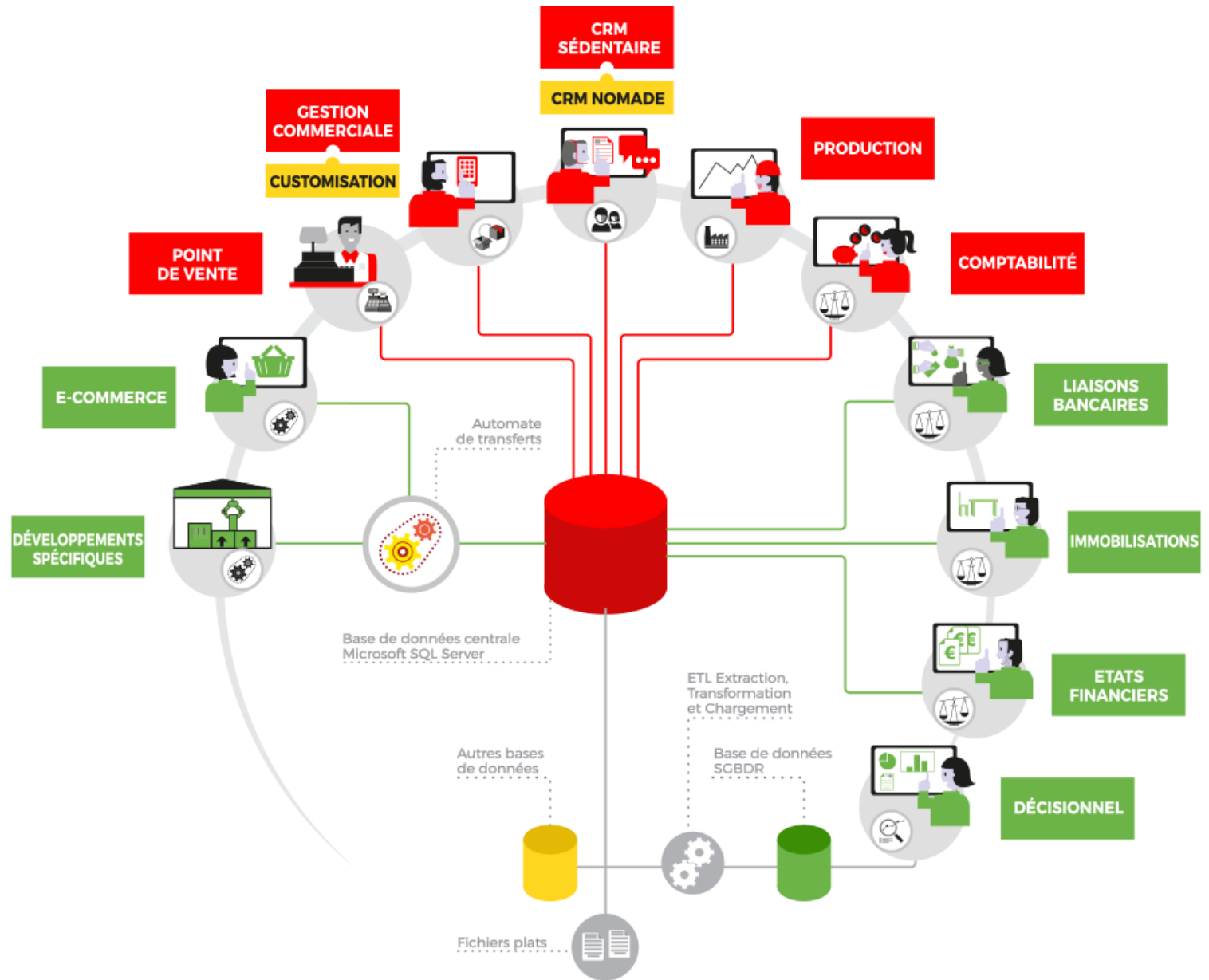
Architecture fonctionnelle (2/4)

- ❑ Les besoins des métiers sont traduits en besoin fonctionnel
- ❑ Représente les **flux de données**
- ❑ Représente l'organisation des différents métiers et fonctions



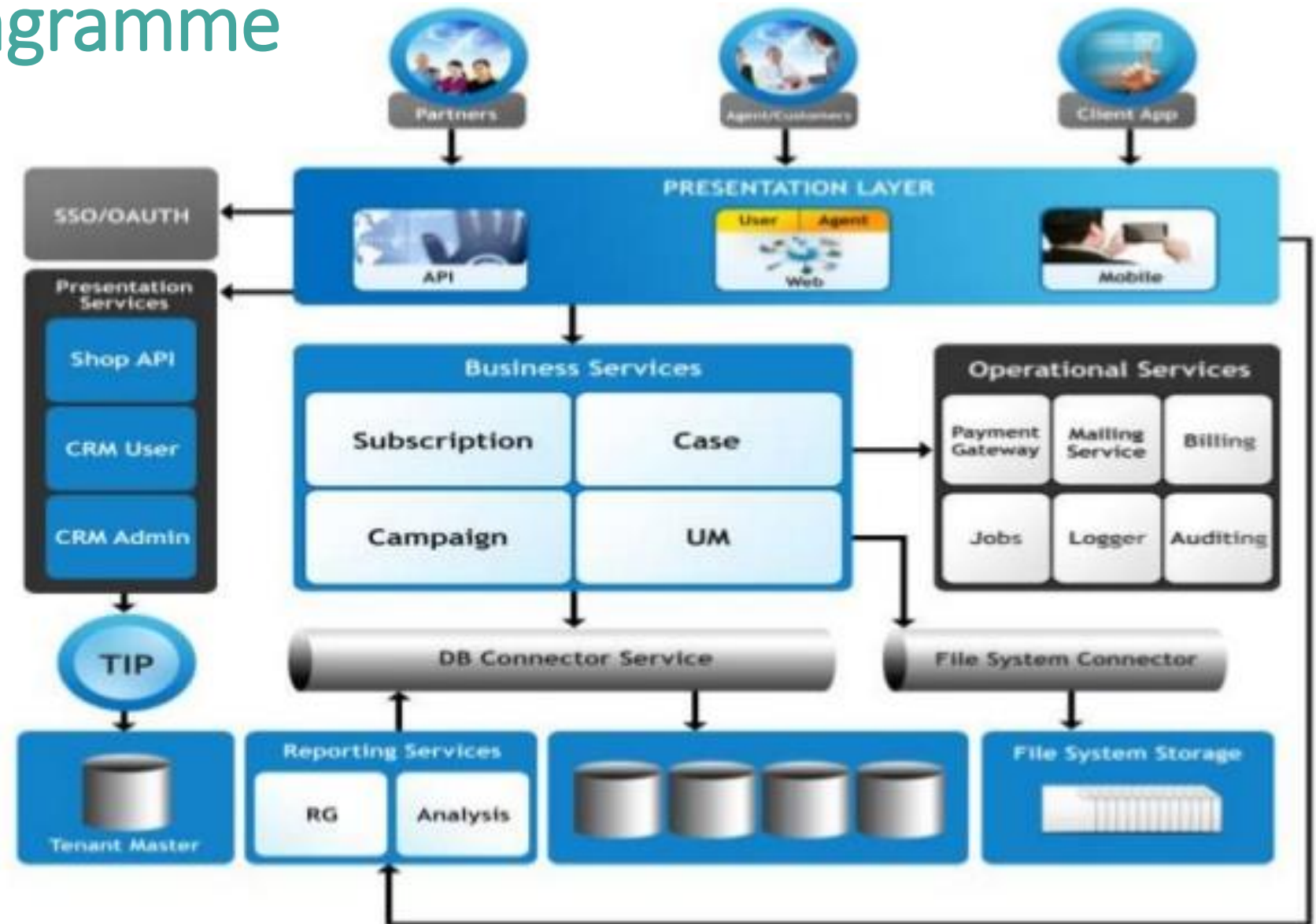
Exemple 1 : diagramme d'architecture fonctionnelle

- ❑ CRM
- ❑ <http://www.wavesoft.fr/votre-gestion-complete-et-simplifiee-en-un-seul-logiciel.html>



Exemple 2 : Diagramme d'architecture fonctionnelle

- ❑ CRM https://www.slideshare.net/CSS_Corp/building-saas-enabled-applications



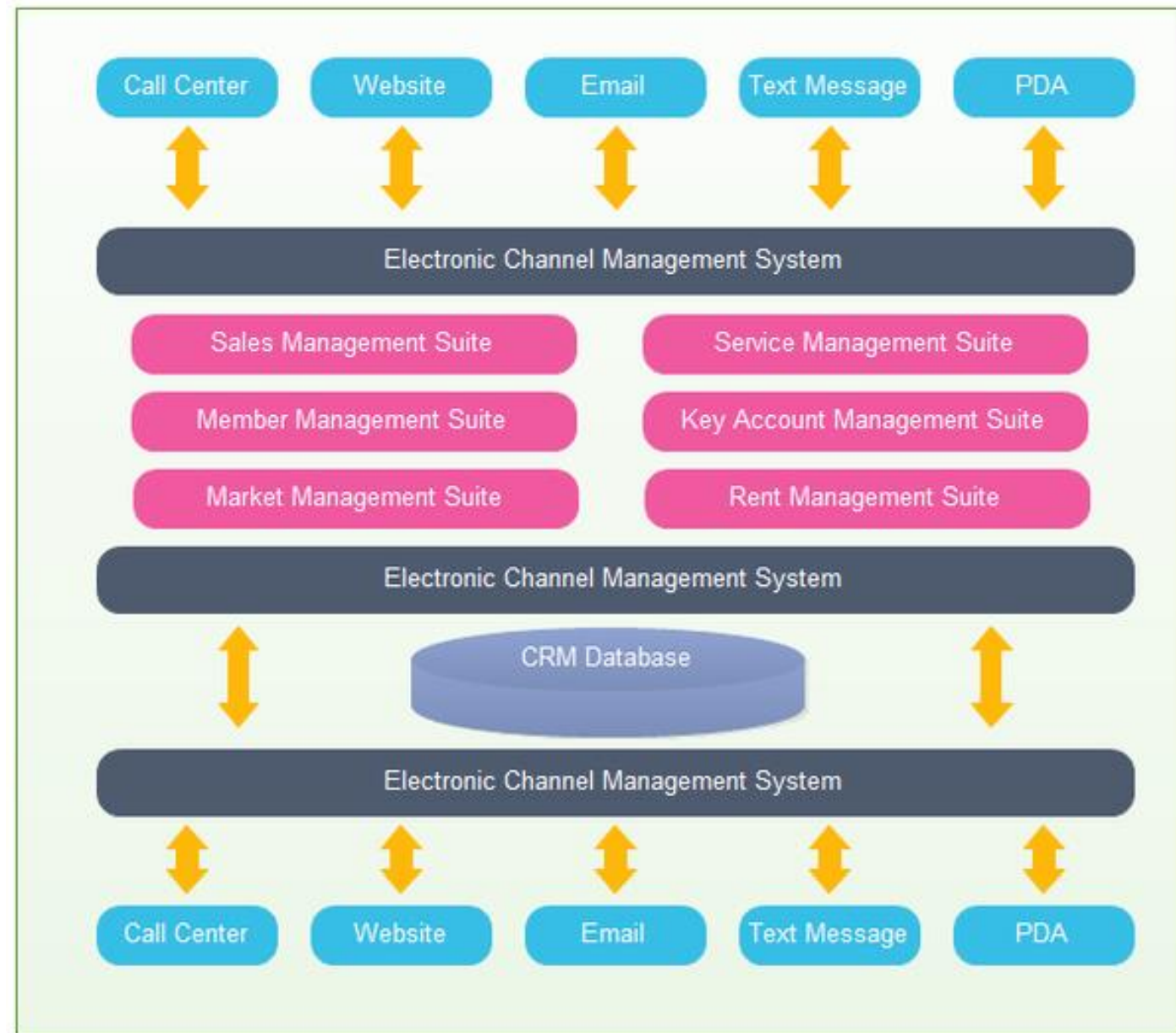
Architecture applicative ou logicielle (3/4)

□ Document orienté logiciel

- l'ensemble des applications qui constituent la partie informatisée du SI
- Les échanges de données et interfaces entre les logiciels
- Comment sont reliés les logiciels, applications serveurs (couches applicatives)

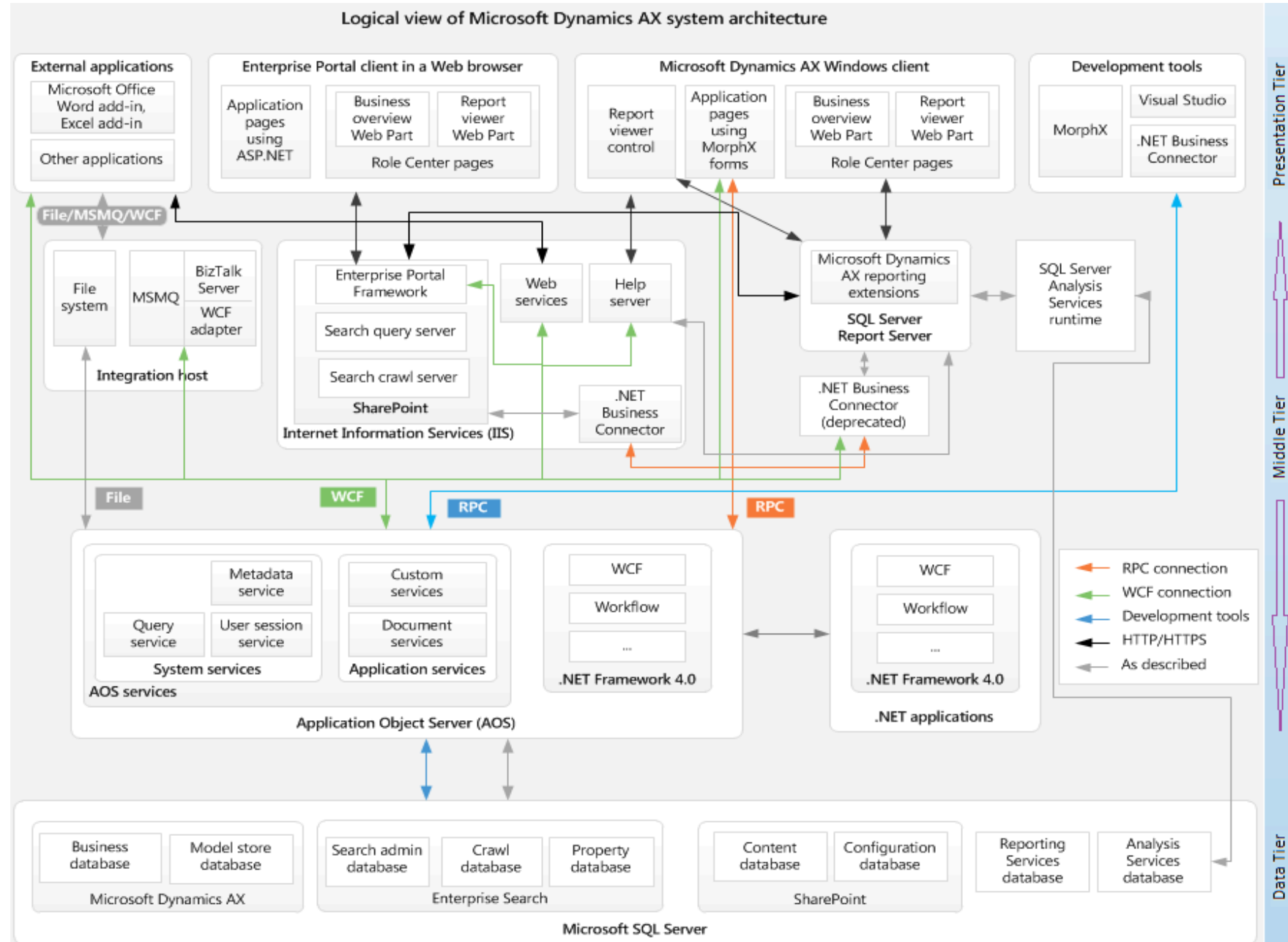
Exemple 1 d'architecture logicielle

- ❑ <https://www.quora.com/How-do-you-make-architectural-diagrams>



Exemple 2 d'architecture logicielle

❑ Vision du CRM selon Microsoft



Architecture Technique (4/4)

□ Organisation logique du SI

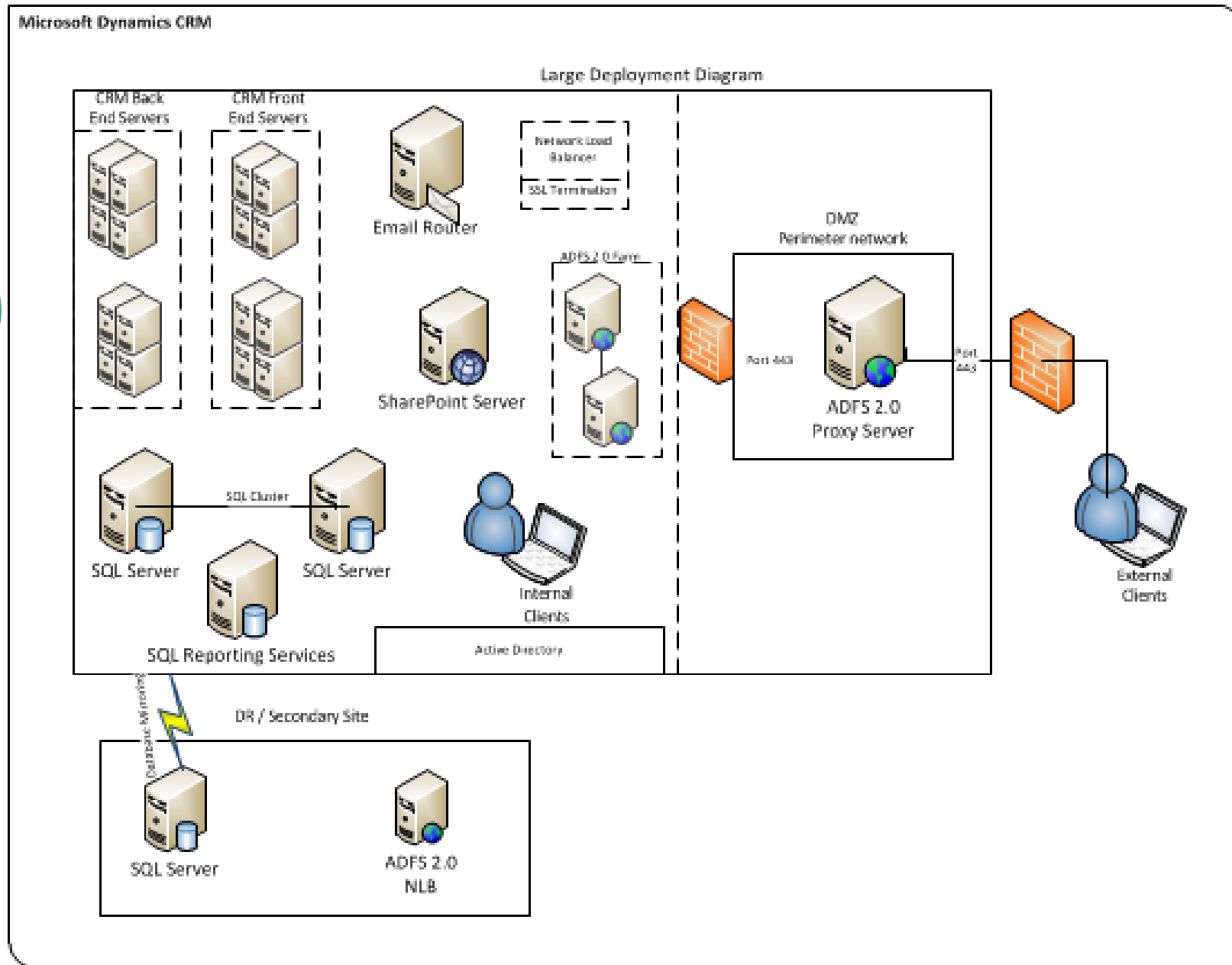
- Matériel informatique : Clients, serveurs, routeurs, firewall,
- les logiciels,
- Les systèmes,
- Les middlewares,
- Les réseaux de télécommunication
- les relations entre ces différents éléments

□ UML

- Diagramme de déploiement

Exemple d'architecture Technique (4/4)

- <https://crmbook.powerobjects.com/system-administration/server-installation/multiple-server-deployment-2/>



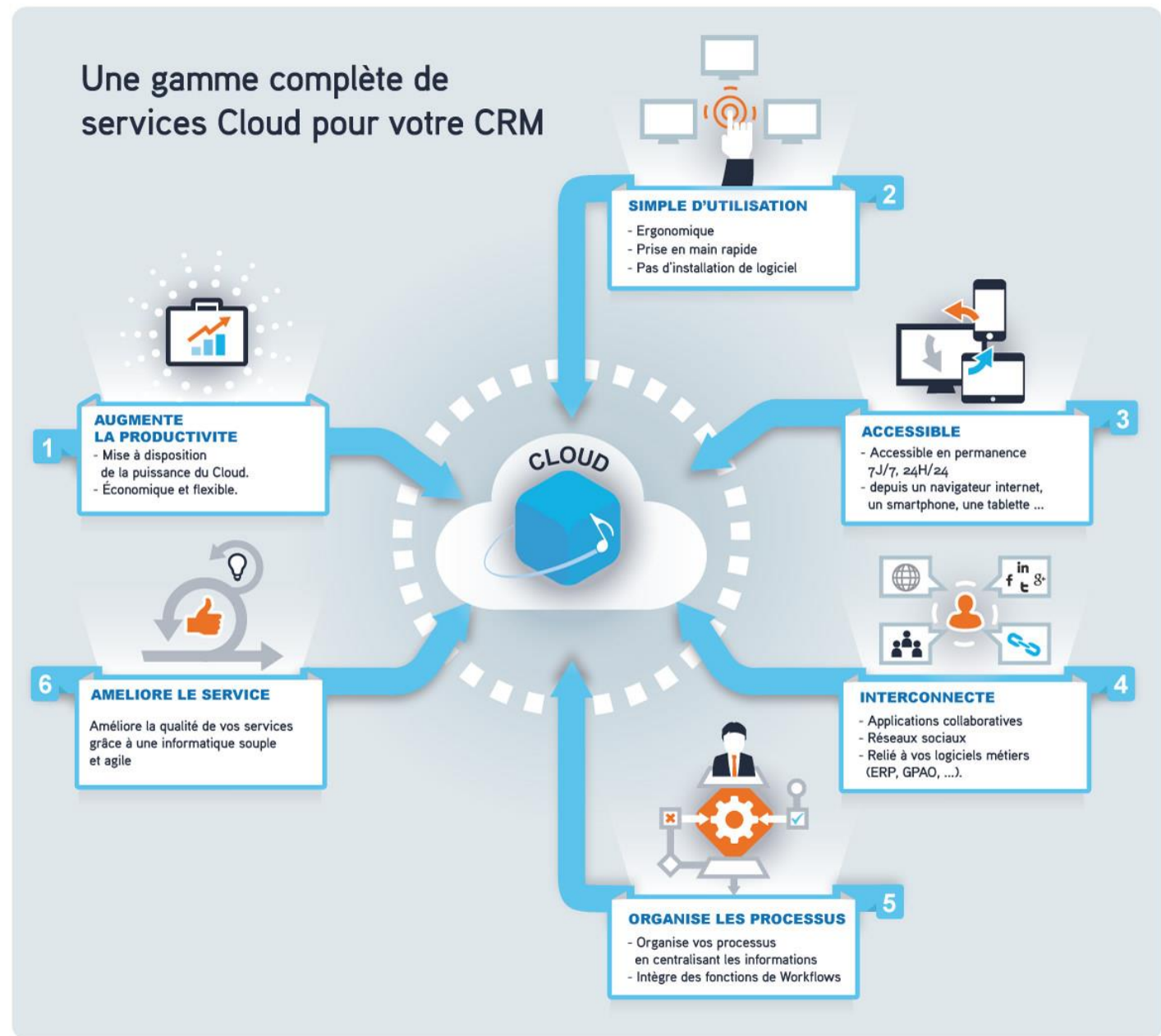
Autres documentations

Autres documentations

- ❑ **Documentations financières**
- ❑ **Documentations politiques, stratégiques**
- ❑ **Documentations marketing**

Documentation marketing

- ❑ Exemple CRM (version cloud)
- ❑ <https://www.bluenote-systems.com/services-crm/crm-cloud-computing.html>



Références

Références

- ❑ **Comment écrire une documentation logicielle**
 - <https://fr.wikihow.com/%C3%A9crire-une-documentation-logicielle>
- ❑ **Comment Rédiger une Expression de Besoins Efficace ?**
 - <https://www.axiocode.com/10-points-infaillibles-rediger-expression-de-besoins-efficace/>
- ❑ **Rédiger une documentation utilisateur**
 - [https://documentation.unistra.fr/Catalogue/Production et diffusion/DocUtilisateur/Dokiel/co/TypeDoc.html](https://documentation.unistra.fr/Catalogue/Production%20et%20diffusion/DocUtilisateur/Dokiel/co/TypeDoc.html)
- ❑ **Ian Sommerville (2001) Software Documentation**
 - www.literateprogramming.com/documentation.pdf
- ❑ **Partons ensemble à la découverte de l'Architecture SI !**
 - <http://www.emsi-histoireinformatique.fr/13736/>
- ❑ **La documentation logicielle**
 - <https://medium.com/@nioperas06/la-documentation-logicielle-79cf962a593b>

Références

❑ JavaDoc Open Class Rooms

- <https://openclassrooms.com/fr/courses/1115306-presentation-de-la-javadoc>

❑ How to Write Doc Comments for the Javadoc Tool

- <https://www.oracle.com/technetwork/java/javase/index-137868.html>

❑ Développns en Java

- <https://www.jmdoudoux.fr/java/dej/chap-javadoc.htm>

❑ What is a CRM

- <http://jisc.cetis.ac.uk/crm-tools/what-is-crm.html>