

Rapport DU Projet SOA

H4103 2014

Etude et conception d’architectures de services dans un SI urbanisé (Banque)

[Conception globale de l’architecture applicative 1](#_Toc402943037)

[Modèles Conceptuels de données 1](#_Toc402943038)

[Blocs applicatifs 1](#_Toc402943039)

[Clients & Produits : 1](#_Toc402943040)

[Commercial : 1](#_Toc402943041)

[Cycle de vie des objets metiers 5](#_Toc402943042)

[Choix de l’environnement technique 5](#_Toc402943043)

[Conception fonctionnelle détaillée 6](#_Toc402943044)

[Conception applicative détaillée 7](#_Toc402943045)

[Architecture technique 8](#_Toc402943046)

[Contacts 9](#_Toc402943047)

# Conception globale de l’architecture applicative

## Modèles Conceptuels de données

L’architecture applicative de notre SI s’appuie sur des modèles conceptuels bien précis décrivant l’ensemble des objets métiers du domaine.  
Ces modèles décrivent les entités interagissant avec les agents (en plus du chef d’agence). Tout d’abord les entités commerciales : celles-ci se focalisent sur le management du capital Temps de l’agent ainsi que ses Contacts (les différents états de cet objet-métier (OM) seront détaillés ultérieurement). Ensuite, nous avons les entités Clients et Produits qui traitent les différentes interactions entre l’ensemble des clients, des produits et des entités constitutives de la Banque.   
Afin de respecter les règles d’urbanisation, nous procèderons au découpage de ces modèles en blocs et systèmes applicatifs. Ces blocs seront capables d’assurer un maximum de cohérence tout en maintenant un minimum de couplage entre les différents Objets-Métiers.

## Blocs applicatifs

### Clients & Produits :

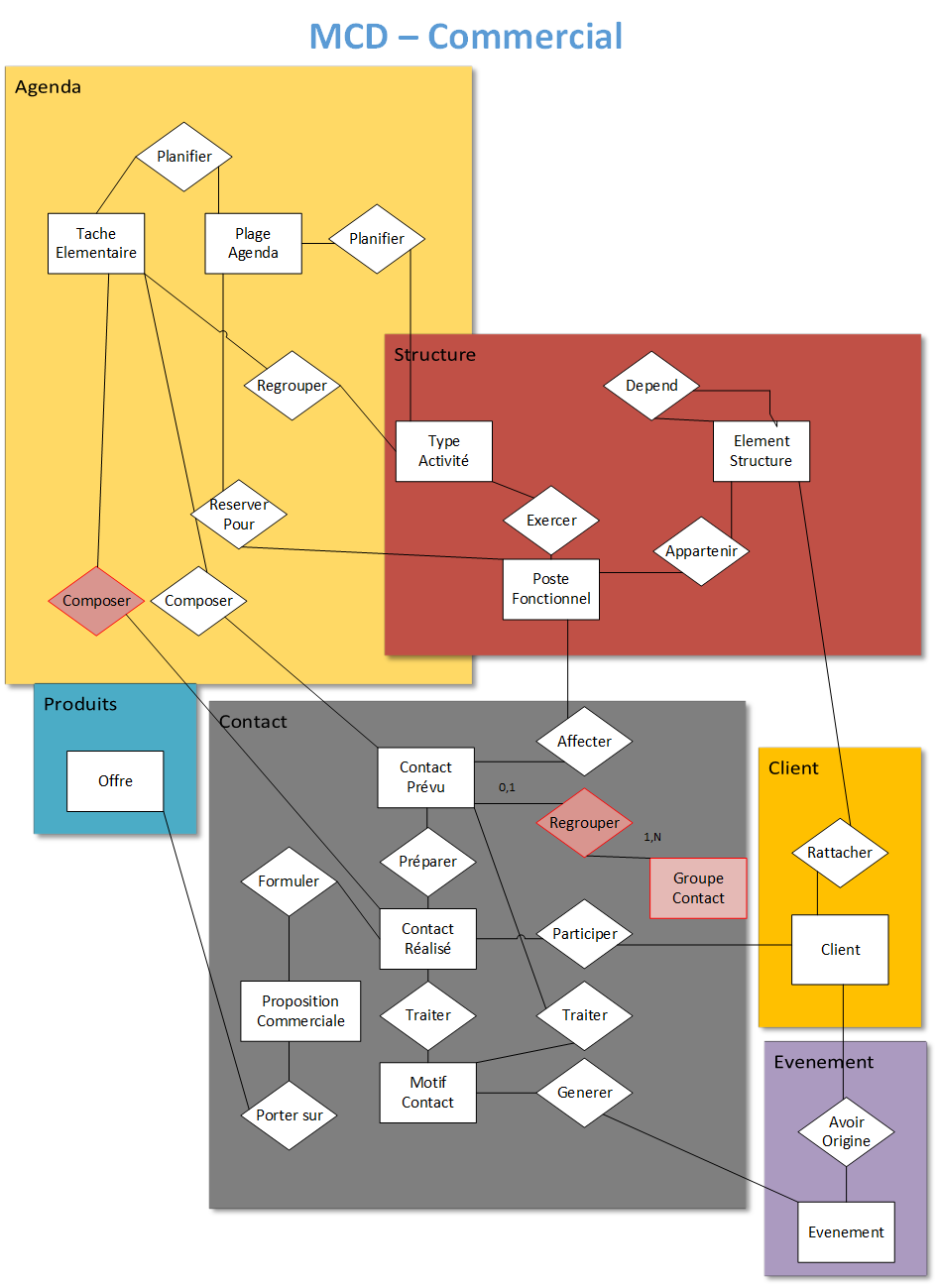
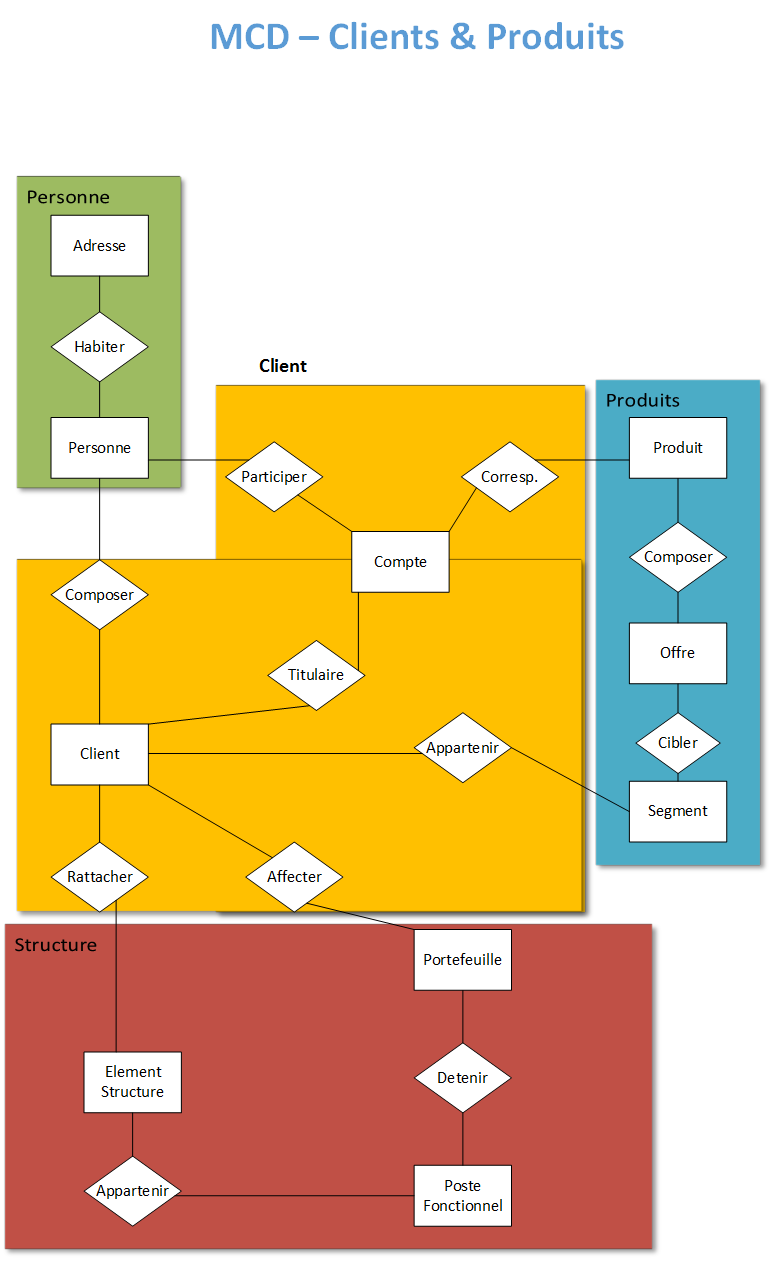
Au cours de notre analyse, nous avons opté pour un découpage en 4 blocs applicatifs distincts :

* Le bloc **Client** : ce bloc regroupe 2 OM fondamentaux dans le SI de l’entreprise, à savoir l’objet *Client* et l’objet *Compte.* Ce dernier représente une « instance » d’un produit bancaire quelconque lié à un unique *Client*, expliquant le choix de l’intégrer dans ce bloc applicatif.
* Le bloc **Personne :** ce bloc rassemblant 2 OM, *Personne* et *Adresse*. Cette dernière contribue aux données signalétiques de toute personne, physique ou morale.
* Le bloc **Produit** : ce bloc regroupe *Produit, Offre* ainsi que *Segment*. Ces OM englobent toute l’offre bancaire disponible pour les *Clients*.
* Le bloc **Structure**: ce bloc réunit toutes les entités propres à la structure de l’entreprise, à savoir *le Portefeuille* (l’ensemble des clients assignés à un *Agent*)*,* *Le Poste Fonctionnel* (désignant l’ensemble des salariés de l’entreprise) et *L’Elément Structure* (contenant l’ensemble des entités constitutives de l’entreprise : Siège, Agences…).

### Commercial :

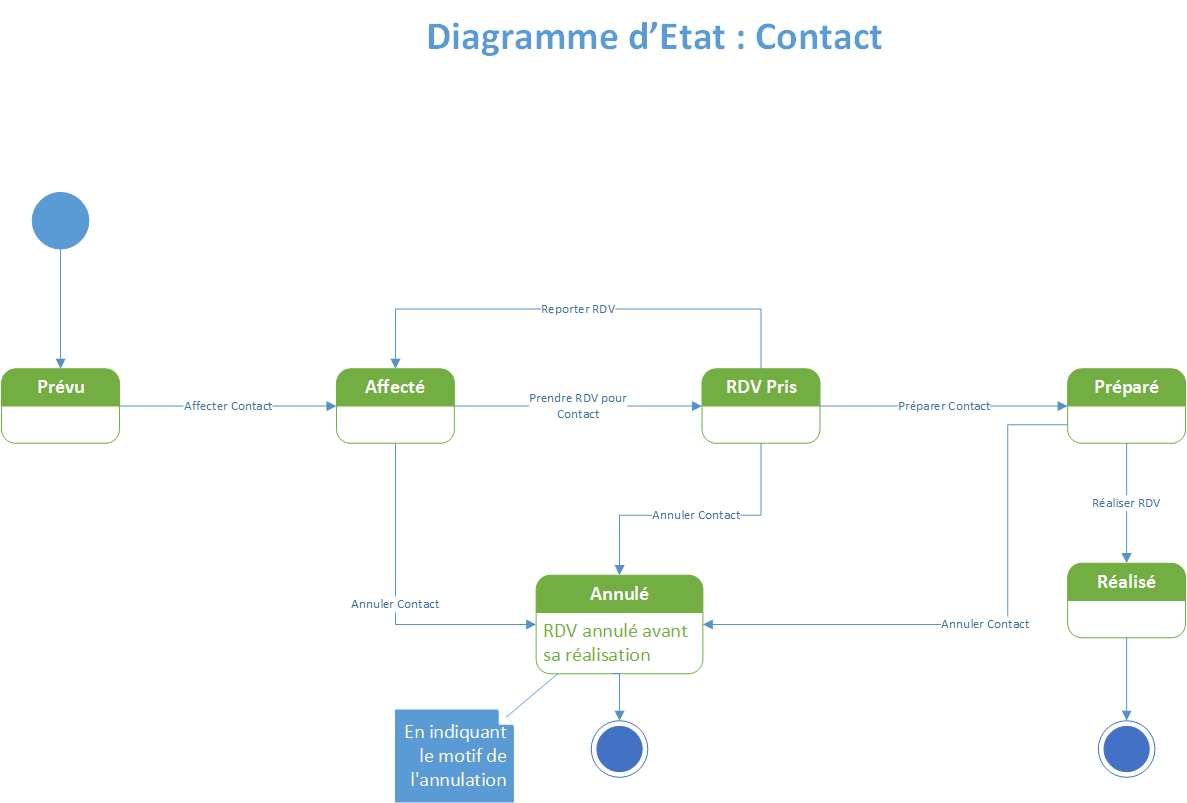
Ce modèle conceptuel sera découpé en 6 blocs distincts, reprenant des blocs précédemment définis.

* Le bloc **Agenda**: ce bloc regroupe d’un côté les *Plages Agenda* qui précisent un type d’activité pour une plage horaire d’un Agent, et de l’autre les *Taches Elémentaires* qui symbolisent entre autres les rendez-vous ainsi que diverses actions pouvant être menées par l’Agent.
* Le bloc **Structure**: ce bloc précédemment défini se retrouve enrichi par un OM, le *Type D’activité*. Ce dernier désigne les différentes fonctions que peut exercer un salarié durant son jour ouvré (Commercial, Accueil …).
* Le bloc **Contact :** ce bloc englobe les entités traduisant les *Contacts* prévus ou réalisés par les Agents concernant leurs *Clients* affectés.
* Le bloc **Evènement : …**
* Les blocs **Client** et **Produit** participants dans ce modèle conceptuel restent identiques à ceux précédemment définis.



## Cycle de vie des objets metiers

Nous nous concentrerons dans cette section sur le cycle de vie de l’OM *Contact*.   
Cet objet est initialement à l’état **Prévu**. Lorsqu’il est affecté à un agent, le contact passe à l’état **Affecté.** A partir de là, on peut prendre un RDV pour basculer à l’état **RDV Pris**., état qui sera en attente de la préparation de l’agent pour transiter à l’état **Préparé.** Notons qu’à tout moment, les 3 derniers états (Affecté, RDV Pris et Préparé) peuvent basculer à l’état final **Annulé,** marquant un RDV annulé avant sa réalisation. Enfin, un second état final est prévu pour l’OM, **Réalisé**, activé par la réalisation d’un RDV préparé.

**

## Choix de l’environnement technique

Conformément aux plans de la MOA de l’entreprise, l’architecture technique globale mise en œuvre sera une architecture **Client/Serveur n-tiers.** Cette dernière bénéficiera des avantages du modèle Client/Serveur, à savoir une centralisation globale des données (la gestion des clients/produits reste sur le site central), simplifiant ainsi les contrôles de sécurité et les mises à jour. Les serveurs supportent toute la charge de calcul, allégeant au maximum la charge des terminaux « Agent ». Ainsi, nous opterons pour une solution **Web** en adoptant des **clients légers** pour la couche présentation.

De plus, l’aspect multi-tiers apporte davantage de flexibilité et de performance dans la mesure où la multiplicité des niveaux permettra une spécialisation plus accrue des serveurs.

# Conception fonctionnelle détaillée

## Diagrammes d’activité & SMA :

## Spécification IHM centrée OM :

### EDF :

### Description des Fenêtres :

### SMA :

# Conception applicative détaillée

## Services et dynamique de l’architecture

## Specification des services (IHM contact)

### Consulter Contacts de l’agence :

|  |  |
| --- | --- |
| SM2 : GetListeContactsPrévus  SOM Liés : GetListeClients (SOM3), GetContactsPrévus (SOM4) | |
| Entrée : idAgence | **Sortie** : liste<ContactPrévu,MotifContact,Client> |
| Ce service métier renvoie une liste de contacts liés à une agence, chaque contact est accompagné d’un motif ainsi que les clients concernés. | |
| Procédure :   * Lister les clients de l’agence (SOM3) * Pour chaque client, lister ses contacts prévus en plus des motifs (SOM4) | |

|  |  |
| --- | --- |
| SOM3 : GetListeClient | SOM4 : GetContactsPrévus |
| Bloc : Client  Entités : Client | **Bloc** : Contact  **Entités** : ContactPrévu || MotifContact |
| Entrée : idAgence | Sortie : liste<Client> | **Entrée** : IdClient | **Sortie** : liste<ContactPrévu,MotifContact> |
| Renvoyer la liste de clients gérés par l’agence | Renvoyer la liste des contacts prévus liés à un client particulier |

### Affecter Agent à Contact :

|  |  |
| --- | --- |
| SM5 : AffecterAgent  SOM Liés : AffecterAgent (SOM7), MAJinfosAgent (SOM8) | |
| Entrée : idAgence, idContact | **Sortie** : Bool |
| Ce service métier se charge de l’affectation d’un contact à un agent. | |
| Procédure :   * Affecter le client à l’agent concerné, création de la relation Affecter (SOM7) * Mettre à jour les informations de l’agent (SOM8) | |

|  |  |
| --- | --- |
| SOM7 : AffecterAgent | SOM8 : MAJinfosAgent |
| Bloc : Contact  Entités : Contact | **Bloc** : Structure  **Entités** : Poste Fonctionnel |
| Entrée : idContact, idAgent | Sortie : Bool | **Entrée** : IdAgent | **Sortie** : Bool |
| Affecter le client à l’agent concerné en alimentant la relation « Affecter ». Un booléen de confirmation est renvoyé. | Mettre à jour les informations de l’agent concerné par l’affectation.Un booléen de confirmation est renvoyé. |

# Architecture technique

Conformément aux plans de la MOA de l’entreprise, l’architecture technique globale mise en œuvre sera une architecture **Client/Serveur n-tiers.** Nous détaillerons dans cette section du rapport l’architecture technique tout en spécifiant les différents niveaux logiques.

## Couches présentation & Logique applicative

Afin de bénéficier d’un maximum de versatilité et d’autonomie, les différentes IHM de l’application de gestion de l’entreprise seront déployées sur des clients légers. Ainsi, on sépare l’application cliente de la couche présentation. Plus concrètement, nous nous baserons sur des **Applications Java EE**. De ce fait, les IHM représenteront la section Conteneur Web des applications Java EE qui intègre à la fois l’aspect Présentation (Vues Web) et l’aspect Logique Applicative (Composants Contrôleurs).

## Couche Logique Metier & Acces aux ressources

Conformément aux choix précédemment établis, les applications Java EE permettent de détenir le noyau applicatif dans le Conteneur EJB. Ce dernier intègre les composants métiers nécessaires à la logique métier et les composants d’accès aux données.

## Couche Ressources

Etant donné la nature critique des données manipulées par l’ensemble de la structure, la couche Ressources doit satisfaire des critères de sécurité renforcée et de performance limitant erreurs et incohérences. De ce fait, cette couche se basera sur un **Système de Gestion de Bases de Données Relationnels et Distribué (Oracle Database, Microsoft SQL Server …)**. Ce système est destiné à assurer l'indépendance des données et à offrir les moyens de contrôler la cohérence et d'éviter la redondance.

# Contacts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Yassine Moreno  Chef de projet |  | Meryem BenchakrounE Reponsable Qualité |  | Mehdi kitane Responsable conception |
| Sample person image |  | Sample person image |  | Sample person image |
|  |  |  |  |  |
| AMINE ELRHAZI RESPONSABLE COMMUNICATION |  | KARIM BENHMIDA ERGONOME |  | ABDELALIM TRIBAK RESPONSABLE TECHNIQUE |
| **Sample person image** |  | **Sample person image** |  | **Sample person image** |
|  |  |  |  |  |