



# **PROJET DE REFONTE DU SYSTÈME D'INFORMATION**

-

## **DOCUMENT DE DÉFINITION D'ARCHITECTURE**



# DOCUMENT DE DÉFINITION D'ARCHITECTURE

## TABLE DES MATIÈRES

<b>PORTÉE</b>	<b>4</b>
<b>OBJECTIFS ET CONTRAINTES</b>	<b>5</b>
Objectifs	5
Contraintes du projet	5
Techniques	5
Ressources humaines	5
Budget	6
Temps	6
Lignes directrices du projet	6
<b>PRINCIPES D'ARCHITECTURE</b>	<b>7</b>
<b>BASELINE ARCHITECTURE</b>	<b>9</b>
<b>MODÈLES D'ARCHITECTURE POUR L'APPROCHE ARCHITECTURALE</b>	<b>10</b>
Architecture métier	10
Architecture de données	10
Architecture applicative	10
Architecture technique	11
<b>JUSTIFICATION DE L'APPROCHE ARCHITECTURALE</b>	<b>12</b>
Coûts	13
<b>RÉUTILISATION DES ÉLÉMENTS EXISTANTS</b>	<b>16</b>
<b>GAP ANALYSIS</b>	<b>17</b>
Langages de programmation	17
Gestion des données	17
Résumé	19
<b>ÉVALUATION DES IMPACTS</b>	<b>20</b>
<b>ARCHITECTURES DE TRANSITION</b>	<b>21</b>
Définition des différentes étapes de transition	21
Etape 1 - Mise en place de RabbitMQ sur Amazon MQ	21
Etape 2 - Mise en place de Zendesk en SaaS	21
Etape 3 - Mise en place des composants du Service Client	21
Etape 4 - Mise en place des composants du Département Légal	22



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

Etape 5 : Mise en place des composants du Département Vente	23
Etape 6 - Mise en place des composants du Département Facturation	23
Architecture applicative pour chaque étape de transition	24
Architecture technologique pour chaque étape de transition	24
Architecture des données pour chaque étape de transition	24
Architecture métier pour chaque étape de transition	25
<b>ANNEXES</b>	<b>26</b>
Diagramme de l'architecture applicative initiale	27
Diagramme de l'architecture technique initiale	28
Diagramme de l'architecture des données initiale	29
Diagramme de l'architecture métier initiale	30
Diagramme de l'architecture applicative cible	31
Diagramme de l'architecture technique cible	32
Diagramme de l'architecture des données cible	33
Diagramme de l'architecture métier cible	34
Diagramme de transition applicative	35
Diagramme de transition technologique	36
Diagramme de transition des données	37
Diagramme de transition métier	38
Diagramme des bénéfices	39
Catalogue des principes d'architecture	40
Catalogue des principes métiers	41
Catalogue des principes de données	42
Catalogue des principes applicatifs	43
Catalogue des principes technologiques	44
Catalogue des principes de sécurité	45



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

# PORTÉE

LAE (Les Assureurs Engagés) est une entreprise d'assurance spécialisée dans les assurances-vie. Depuis 30 ans, la réputation de l'entreprise s'est bâtie grâce à son engagement auprès des clients, notamment vis-à-vis :

- de la satisfaction des clients par rapport au service client
- de la sécurité des données, en particulier bancaires et à caractère privé.

Le système d'information (SI) de l'entreprise s'est développé petit à petit depuis la création de l'entreprise. Jusqu'à présent, l'entreprise a toujours répondu aux besoins court-termistes des différents services.

Ceci a conduit à l'élaboration d'un système en patchwork dans lequel cohabitent des technologies anciennes et modernes et ayant des conséquences négatives sur les engagements de LAE auprès de ses clients.



## OBJECTIFS ET CONTRAINTES

### *Objectifs*

Les principaux objectifs de la refonte du SI sont :

- Maintenir l'engagement de LAE auprès de ses clients
- Accélérer les temps de traitement
- Simplifier les processus en supprimant les actions manuelles et en les automatisant
- Fiabiliser les données en évitant leurs redondances entre les différents services
- Sécuriser les données privées
- Réduire les frais de maintenance engendrés par une pile technologique vieillissante et inadaptée aux besoins actuels et futurs de l'entreprise

### *Contraintes du projet*

#### Techniques

Le service client impose l'utilisation d'un outil de ticketing externe. Outre cet aspect, vous êtes libre d'utiliser les technologies de votre choix — sous réserve que ce choix repose sur une argumentation convaincante vis-à-vis des coûts et de la dette technique (frais de maintenance notamment).

#### Ressources humaines

En dehors de la collaboration entière de toute l'équipe, LAE met à disposition un ingénieur généraliste en informatique à temps plein (38,5 h/semaine). Toute ressource humaine supplémentaire, comme un développeur d'application pour un langage spécifique, devra être compris dans la limite du budget (voir paragraphe ci-dessous).



# DOCUMENT DE DÉFINITION D'ARCHITECTURE

## Budget

200 000 euros ont été alloués à la mise en œuvre de ce projet. Ce budget doit couvrir l'établissement des spécifications logicielles et matérielles détaillées, le développement, les tests, l'intégration, l'achat de nouveaux matériels et/ou logiciels, ainsi que les besoins en formation.

## Temps

Le système devra être opérationnel d'ici sept mois à compter de la validation de votre solution. L'intégration devra se faire de façon continue, de sorte à ne pas perturber le travail des collaborateurs. Le non-respect de ces conditions entraînera des indemnités compensatrices. Elles pourront être égales au temps payé et non travaillé des collaborateurs en cas d'incapacité de travail introduite par l'intégration de votre solution. Notez également que chaque journée de retard par rapport à l'échéance entraînera également une indemnité de 200 €.

## *Lignes directrices du projet*

Le projet de refonte du système doit être fait en accord avec les processus en vigueur dans les différents services.



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

# PRINCIPES D'ARCHITECTURE

L'ensemble de principes d'architecture devant être respectés pour la mise en place de la nouvelle architecture sont présentés dans les catalogues de principes d'architecture suivants :

- [Catalogue des principes d'architecture](#)
- [Catalogue des principes métiers](#)
- [Catalogue des principes de données](#)
- [Catalogue des principes applicatifs](#)
- [Catalogue des principes technologiques](#)
- [Catalogue des principes de sécurité](#)

Ces principes sont résumés ci-dessous :

La méthode architecturale utilisée pour le projet de refonte du système d'information de LAE se basera sur une architecture orientée événements (ou Event Driven Architecture - EDA en anglais) s'appuyant sur une communication asynchrone entre les services via un agent de messages (ou message broker en anglais).

L'architecture orientée événements (EDA) est un modèle d'architecture logicielle qui favorise la production, la détection, la consommation et la réaction à des changements importants dans l'état d'un système (appelés événements). Ceci est appliqué à travers la conception et la mise en œuvre d'applications et de systèmes qui transmettent des événements entre des composants logiciels et des services faiblement couplés.

Cette architecture sera couplée à une refonte de l'organisation des bases de données tant au niveau des données elles-mêmes qu'au niveau des systèmes de gestion des données.

De plus, un outil de ticketing sera implémenté dans le SI afin de fiabiliser et faciliter les relations inter-services (notamment celles entre le Service client et le Département Légal).



## **DOCUMENT DE DÉFINITION D'ARCHITECTURE**

Le ticketing est un outil de gestion des problèmes rencontrés par les clients. Il définit le niveau d'importance d'une requête, la priorise et la transmet aux personnes concernées pour qu'elles la résolvent rapidement. Cette solution contribue à une meilleure organisation dans le cadre de la relation client, permettant un contact permanent avec les clients afin de garantir un niveau de satisfaction élevé..

Enfin, la pile technologique sera réduite afin de réduire les coûts de maintenance, de faciliter les évolutions futures du SI en se séparant de langages de développement ou de système de gestion de bases de données relationnelles obsolètes ou inadaptées à une utilisation d'hébergement hybride (à la fois en cloud et en local).





## DOCUMENT DE DÉFINITION D'ARCHITECTURE

# BASELINE ARCHITECTURE

L'architecture actuelle (\*1) est organisée en 3 applications indépendantes exploitant chacune une base de données dédiée. Ces applications sont celles des services :

- Vente
- Facturation
- Légal

Le 4ème service (Client) utilise les applications des services Vente et Facturation afin de mettre à jour leurs bases de données. Il utilise également une base de données dédiée sans application afin de stocker les données clients nécessaires à son activité.

Les données stockées dans les différentes bases (\*2) sont des données clients uniquement, réparties en plusieurs types :

- Informations de contact
- Demandes clients
- Informations commerciales
- Informations comportementales
- Informations contractuelles
- Informations bancaires
- Historique de facturation

Chaque service utilise sa propre base de données et les informations de contact sont présentes dans chacune des bases

L'architecture technique (\*3) s'appuie sur une pile technologique disparate et en partie obsolète ou non adaptée aux besoins et objectifs de l'entreprise. Chaque service utilise ses propres technologies.

(\*1) Voir annexe [Diagramme de l'architecture applicative initiale](#)

(\*2) Voir annexe [Diagramme de l'architecture des données initiale](#)

(\*3) Voir annexe [Diagramme de l'architecture technique initiale](#)



# MODÈLES D'ARCHITECTURE POUR L'APPROCHE ARCHITECTURALE

### *Architecture métier*

Le diagramme de l'architecture métier cible (\*1) montre les différents services utilisateurs du SI ainsi que leurs actions et responsabilités.

(\*1) Voir annexe [Diagramme de l'architecture métier cible](#)

### *Architecture de données*

Le diagramme de l'architecture de données (\*1) montre la nouvelle organisation des données au sein des différentes bases de données utilisées par les services.

Nous pouvons voir que, désormais, les données clients de type informations de contact ne sont plus stockées que dans la base de données clients. De plus, les demandes clients ne sont plus stockées dans la base de données du service client.

(\*1) Voir annexe [Diagramme de l'architecture des données cible](#)

### *Architecture applicative*

L'architecture applicative peut être représentée par le diagramme d'architecture applicative en annexe (\*1).

(\*1) Voir annexe [Diagramme de l'architecture applicative cible](#)



# DOCUMENT DE DÉFINITION D'ARCHITECTURE

## *Architecture technique*

L'architecture technique peut être représentée par le diagramme d'architecture technique (\*1).

Dans ce document, nous retrouvons les composants techniques suivants ;

- Utilisation de RabbitMQ comme agent de message afin d'automatiser les tâches des processus inter-services.
- Implémentation d'un agent de message permettant d'automatiser les tâches des processus inter-services
- Utilisation PostgreSQL comme nouveau système de gestion de bases de données relationnelles fonctionnant en cloud et en local et permettant la réplication automatique de la base de données du département Légal
- Généralisation de la pile Vue.js / PHP / Symfony pour les applications et backend des services
- Utilisation de Zendesk comme outil de ticketing en remplacement d'une redirection manuelle des appels et de la base de données Access.

Cette architecture permet de limiter les responsabilités sur les données, de supprimer les doublons de données (pour les informations de contact), d'harmoniser et limiter la pile technologique, d'automatiser toutes les tâches manuelles entre les services et au sein des services eux-mêmes, de sécuriser les données en supprimant les technologies obsolètes

La mise à jour régulière des langages est nécessaire pour mettre en place une nouvelle base d'architecture technique sur laquelle on puisse s'appuyer pour les évolutions futures.

(\*1) Voir annexe [Diagramme de l'architecture technique cible](#)



# JUSTIFICATION DE L'APPROCHE ARCHITECTURALE

L'approche architecturale est justifiée par les éléments suivants :

- La centralisation des données clients de type informations de contact dans une seule base de données permet de :
  - Accélérer les temps de traitement car les informations ne sont modifiées qu'une seule fois
  - Fiabiliser les données en empêchant les oublis de mises à jour multiples de ces données clients
  - Réduire la dette technique en harmonisant les responsabilités sur les données en 'affectant" à chaque service des données dont lui seul est habilité à les gérer.
- L'utilisation d'un nouveau SGBDR unique dans le SI permet de :
  - Automatiser les sauvegardes de la base de données du département Légal
  - Diminuer les coûts de maintenance liés aux multiples SGBDR
  - Réduire la dette technique en supprimant l'utilisation de SGBDR archaïques ou non adaptés au besoin
- L'utilisation d'un outil de ticketing permet de :
  - Répondre à la demande du service Client
  - Sécuriser les données en n'utilisant plus de répertoire partagé
  - Réduire la dette technique en supprimant les actions manuelles sujettes à risque d'oubli
- L'utilisation d'une architecture orientée événements permet de ;
  - Automatiser et fiabiliser les actions inter-services
  - Accélérer les temps de traitement car les informations ne sont modifiées qu'une seule fois
- L'harmonisation des langages de programmation permet de :



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

- Réduire la dette technique en supprimant l'utilisation de langages obsolètes non maintenus
- Sécuriser les applications en supprimant l'utilisation de langages manquant de sécurité

L'ensemble de ces mesures permet également d'envisager l'avenir sereinement en s'appuyant sur des outils et technologies modernes et répandues sur le marché.

### Coûts

L'outil de ticketing est facturé par mois et par utilisateurs. Pour notre besoin, il nous faut 16 utilisateurs; Le coût mensuel pour un utilisateur est de 79 € (pour la Suite Growth) soit un coût total par an de 15168 €

L'agent de message est facturé mensuellement en fonction de l'utilisation. Le tarif pour notre besoin (utilisation de 2 courtiers de messages) sera facturé 91 € par mois soit 1092 € par an.

Le système de gestion de bases de données PostgreSQL est, lorsqu'il est utilisé en cloud sur AWS facturé mensuellement en fonction de l'utilisation. Pour notre besoin, le tarif sera de 1240 € par mois soit 14880 € par an.

La version locale (on premise) de PostgreSQL est sous licence OSI et est donc gratuite. Nous ré-utiliserons les serveurs de données de l'ancienne architecture pour l'hébergement local.

Le reste des coûts concerne l'aspect développement et gestion de projet. Le développement est ici chiffré pour les coûts unitaires suivants :

- 490 € par J/H pour les développements front par un(e) [développeur\(se\) front-end senior](#)
- 560 € par J/H pour les développements back par un(e) [développeur\(se\) back-end senior](#)
- 595 € par J/H pour la gestion de projet par un(e) [architecte logiciel senior](#)



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

Coûts "front" :

Item	JH	Coût unitaire	Coût total
Nouveau front Service Client + hébergement	10		
Nouveau front Département Légal	10		
Nouveau front Département Facturation	10		
<b>Total</b>	<b>30</b>	<b>490</b>	<b>14700</b>

Coûts "back" :

Item	J/H	Coût unitaire	Coût total
Implémentation RabbitMQ	5		
Implémentation Zendesk	5		
Nouveau backend Service Client + hébergement	12		
API Service Client	4		
Nouveau backend Département Légal	12		
Nouveau backend Département Facturation	12		
Nouvelle base de données Service Client PostgreSQL + hébergement	8		
Nouvelle base de données Département Légal PostgreSQL + Mise en place du backup	8		
Nouvelle base de données Département Vente PostgreSQL	8		
Nouvelle base de données Département Facturation PostgreSQL	8		
Migration des donnés	10		
<b>Total</b>	<b>92</b>	<b>560</b>	<b>51520</b>



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

Coûts "projet" :

Item	J/H	Coût unitaire	Coût total
Analyse (ce projet) - 10 jours mais ne comptent pas dans le total des jours puisque phase amont	10	595	5950
Rédaction des spécifications	15	595	8925
Formation outil ticketing Service Client (4 pers)	2	595	1190
Formation outil ticketing Département Légal (4 pers)			
Formation outil ticketing Département Vente (3 pers)			
Formation outil ticketing Département Facturation (3 pers)			
Formation outil ticketing Autres Services (2 pers)			
<b>Total</b>	<b>27</b>		<b>16065</b>
<b>Total à prendre en compte</b>	<b>17</b>		<b>16065</b>

Le total des coûts est estimé à 82285 € pour une durée de 139 jours soit 6 mois et 19 jours en ETP. Les contraintes de 200000 € et 7 mois sont donc respectées.



# RÉUTILISATION DES ÉLÉMENTS EXISTANTS

Certains éléments de l'architecture initiale ont été conservés :

- Le service d'hébergement d'AWS est conservé et généralisé pour toutes les applications et bases de données (hormis la BDD du département Légal)
- L'application du département Vente (front et back) est globalement conservée dans la nouvelle architecture. Seule la technologie utilisée pour la base de données fait l'objet d'une modification.
- Les serveurs hébergeant les données du département Légal sont conservés.
- Le système de gestion des utilisateurs de chaque service/département est conservé car il ne fait pas partie du périmètre du projet. Cependant, il convient de le faire évoluer dans le futur car il participe à la dette technique en n'ayant pas prévu dès le début de la mise en place de la gestion des utilisateurs du SI une approche plus centralisée et commune à tous les services.





# GAP ANALYSIS

L'architecture de base nécessite de supprimer les composants de la pile technologique ne permettant pas d'atteindre les objectifs de l'entreprise.

Il convient également de supprimer autant que faire se peut les actions manuelles des utilisateurs qui présentent des risques pour la qualité des données et des processus métier.

### *Langages de programmation*

Les langages de programmations Java et COBOL sont à supprimer de la pile technologique :

- COBOL est un langage obsolète (n'est plus maintenu depuis 2014) qui n'est pas adapté aux approches architecturales modernes et qui, de part la difficulté à trouver de la main d'oeuvre pour la maintenance des applications, rend l'architecture actuelle fragile
- Java notamment via l'utilisation de Java Applet qui n'est plus maintenu depuis 2016 est à considérer comme obsolète dans l'architecture actuelle

L'obsolescence de ces 2 langages rend l'architecture actuelle fragile et ne permet pas l'évolutivité de celle-ci. La sécurité des données que nous stockons n'est de ce fait plus assurée.

### *Gestion des données*

La pile technologique est composée d'un ensemble de systèmes de gestion de bases de données hétéroclites.

Access est un système de gestion de bases de données relationnel (SGBDR) qui est utilisé dans notre contexte avec sa propre application pour gérer les données (formulaires) et est associé avec un fichier de bases de données stocké dans un répertoire partagé qui est synchronisé avec la



## **DOCUMENT DE DÉFINITION D'ARCHITECTURE**

version local du même fichier présent sur le PC de chaque utilisateur l'exploitant.

L'utilisation de cette synchronisation inclut un risque important en termes de fiabilité car des interruptions de réseau peuvent altérer la qualité et la sécurité des données. De plus, c'est une opération qui consomme généralement beaucoup de temps ne permettant pas aux utilisateurs d'être pleinement efficaces.

La multiplicité des SGDBR utilisés dans l'entreprise génèrent des coûts de maintenance élevés. L'harmonisation des ces systèmes est donc nécessaire afin de fiabiliser et sécuriser les données des nos clients et réduire ces frais de maintenance.

L'organisation des données clients et l'architecture utilisée oblige les utilisateurs à réaliser des opérations de double saisie des données de contact des clients. Ceci présente un risque d'intégrité des données car les opérations de double saisie peuvent faire l'objet d'oubli et certaines données peuvent être saisies différemment et présenter des erreurs. De plus, c'est une opération chronophage qu'il convient d'automatiser.

L'organisation manuelle de la sauvegarde de la base de données du département Légal (contenant les données les plus sensibles), n'est pas une méthode fiable et suffisamment robuste qui nécessite d'être automatisée.



# DOCUMENT DE DÉFINITION D'ARCHITECTURE

## Résumé

Tous les composants obsolètes nécessitent donc d'être remplacés par des composants encore maintenus ou plus adaptés.

Les SGBDR doivent être unifiés par un seul et même système fonctionnant de manière hybride (en cloud et en local) et contenant nativement un outil de sauvegarde automatique.

La refonte des responsabilités sur les données permet d'augmenter la qualité des données clients en supprimant les opérations de double saisie par le service Client.

L'ajout d'un agent de message va permettre de compenser le manque de fiabilité de l'ancienne architecture qui nécessitait des actions manuelles dans les relations entre services/départements.

L'ajout d'un outil de ticketing permet de compenser les lacunes de l'ancienne architecture au niveau des relations entre le service Client et le département Légal.



# ÉVALUATION DES IMPACTS

L'évolution de l'architecture aura des impacts négatifs mineurs sur l'utilisation du nouveau système d'information.

Les principaux impacts pour les utilisateurs seront principalement liés à l'utilisation de l'outil de ticketing. Pour compenser cet impact, une formation de 2 jours pour tous les utilisateurs concernés devra être organisée.

Les différentes étapes de transition devraient permettre de minimiser les impacts sur la stabilité de l'architecture et sur la continuité d'utilisation du système d'information.

Cette nouvelle architecture aura également des impacts positifs :

- Simplification et automatisation des processus
- Facilitation des relations inter-services
- Fiabilisation, sécurisation et meilleure robustesse du SI
- Remboursement de la dette technique

Les bénéfices qui seront tirés de cette nouvelle architecture sont présentés dans le [diagramme des bénéfices](#) en annexe.



# ARCHITECTURES DE TRANSITION

### *Définition des différentes étapes de transition*

#### Etape 1 - Mise en place de RabbitMQ sur Amazon MQ

Mise en place de RabbitMQ sur Amazon MQ

#### Etape 2 - Mise en place de Zendesk en SaaS

Mise en place de Zendesk en mode SaaS (Software As A Service ou logiciel en tant que service).

#### Etape 3 - Mise en place des composants du Service Client

- Développement d'un nouveau backend avec PHP / Symfony et hébergement sur AWS
- Développement de l'API REST pour l'interrogation de la base de données client
- Mise en place de la base de données PostgreSQL sur Amazon RDS et migration des données de Access vers PostgreSQL
- Développement d'un nouveau frontend avec Vue.js et hébergement sur AWS
- Tests

Les données de type informations de contact (présentes dans toutes les anciennes bases de données) qui feront foi seront désormais celles de la BDD du service Client.



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### Etape 4 - Mise en place des composants du Département Légal

- Développement d'un nouveau backend avec PHP / Symfony incluant la gestion des messages pour l'exploitation de l'agent de message
- Hébergement en local en ré-exploitant l'ancien serveur utilisé pour l'application COBOL
- Mise en place de PostgreSQL en local en ré-exploitant le serveur local utilisé pour la BDD SQLite et migration des données (comportementales et contractuelles) de SQLite vers PostgreSQL
- Mise en place de la sauvegarde automatique quotidienne (disponible sur PostgreSQL)
- Développement d'un nouveau frontend avec Vue.js et hébergement en local sur l'ancien serveur utilisé pour l'application COBOL
- Tests

Lors de cette étape, les données de type informations de contact de la BDD du département Légal seront supprimées.

Lors de cette étape de transition, les échanges entre le département Légal et les départements Vente et Facturation continueront à se faire comme avant (sans passer par l'agent de message).



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### Etape 5 : Mise en place des composants du Département Vente

- Développement d'une fonction permettant la gestion des messages pour l'exploitation de l'agent de message
- Mise en place de la base de données PostgreSQL sur AWS et migration des données (commerciales) de MySQL vers PostgreSQL
- Tests

Lors de cette étape, les données de type informations de contact de la BDD du département Vente seront supprimées.

Lors de cette étape de transition, les échanges entre le département Vente et les services Légal et Facturation continueront à se faire comme avant (sans passer par l'agent de message).

### Etape 6 - Mise en place des composants du Département Facturation

- Développement d'un nouveau backend avec PHP / Symfony incluant la gestion des messages pour l'exploitation de l'agent de message
- Mise en place de la base de données PostgreSQL sur AWS et migration des données (bancaires et historiques de facturation) de MySQL vers PostgreSQL
- Développement d'un nouveau frontend avec Vue.js
- Tests

Lors de cette étape, les données de type informations de contact de la BDD du département Facturation seront supprimées.

Lors de cette étape de transition, les échanges entre tous les services/départements se feront par l'agent de message. Il n'existera plus d'échange "manuels".



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### *Architecture applicative pour chaque étape de transition*

Le diagramme de transition applicatif (\*1) montre à quelles étapes les différentes applications et services (anciens et nouveaux) seront implémentés dans la nouvelle architecture.

(\*1) Voir annexe [Diagramme de transition applicative](#)

### *Architecture technologique pour chaque étape de transition*

Le diagramme de transition technologique (\*1) montre à quelles étapes les différentes technologies seront implémentées dans la nouvelle architecture.

(\*1) Voir annexe [Diagramme de transition technologique](#)

### *Architecture des données pour chaque étape de transition*

Le diagramme de transition des données (\*1) montre à quelles étapes les différentes bases de données (anciennes et nouvelles) seront implémentées dans la nouvelle architecture.

(\*1) Voir annexe [Diagramme de transition des données](#)





## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### *Architecture métier pour chaque étape de transition*

Le diagramme de transition métier (\*1) montre à quelles étapes les fonctions métiers seront implémentées dans la nouvelle architecture.

(\*1) Voir annexe [Diagramme de transition métier](#)



## ANNEXES

[Diagramme de l'architecture applicative initiale](#)

[Diagramme de l'architecture technique initiale](#)

[Diagramme de l'architecture des données initiale](#)

[Diagramme de l'architecture métier cible](#)

[Diagramme de l'architecture applicative cible](#)

[Diagramme de l'architecture technique cible](#)

[Diagramme de l'architecture des données cible](#)

[Diagramme de transition applicative](#)

[Diagramme de transition technologique](#)

[Diagramme de transition de données](#)

[Diagramme de transition métier](#)

[Diagramme des bénéfices](#)

[Catalogue des principes d'architecture](#)

[Catalogue des principes métiers](#)

[Catalogue des principes de données](#)

[Catalogue des principes applicatifs](#)

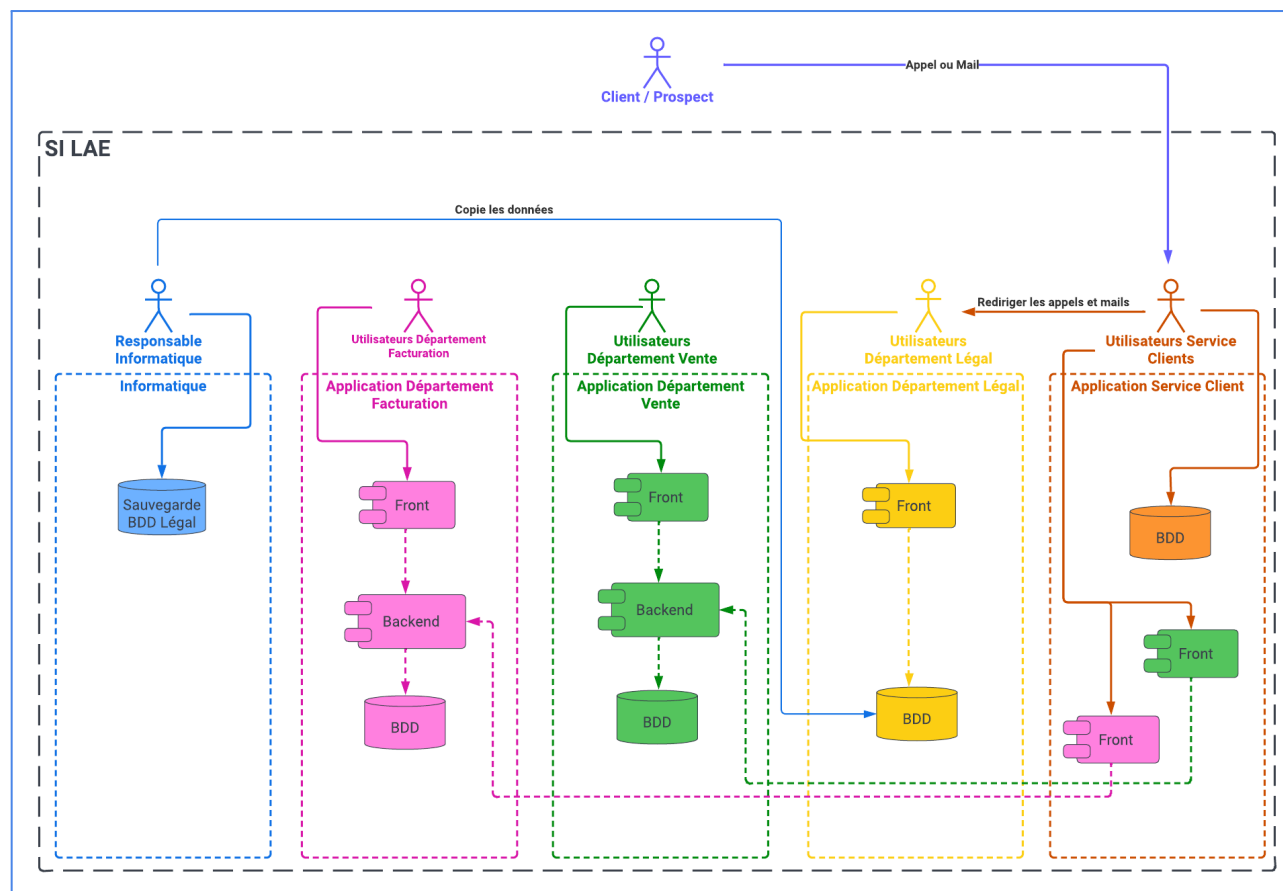
[Catalogue des principes technologiques](#)

[Catalogue des principes de sécurité](#)



# DOCUMENT DE DÉFINITION D'ARCHITECTURE

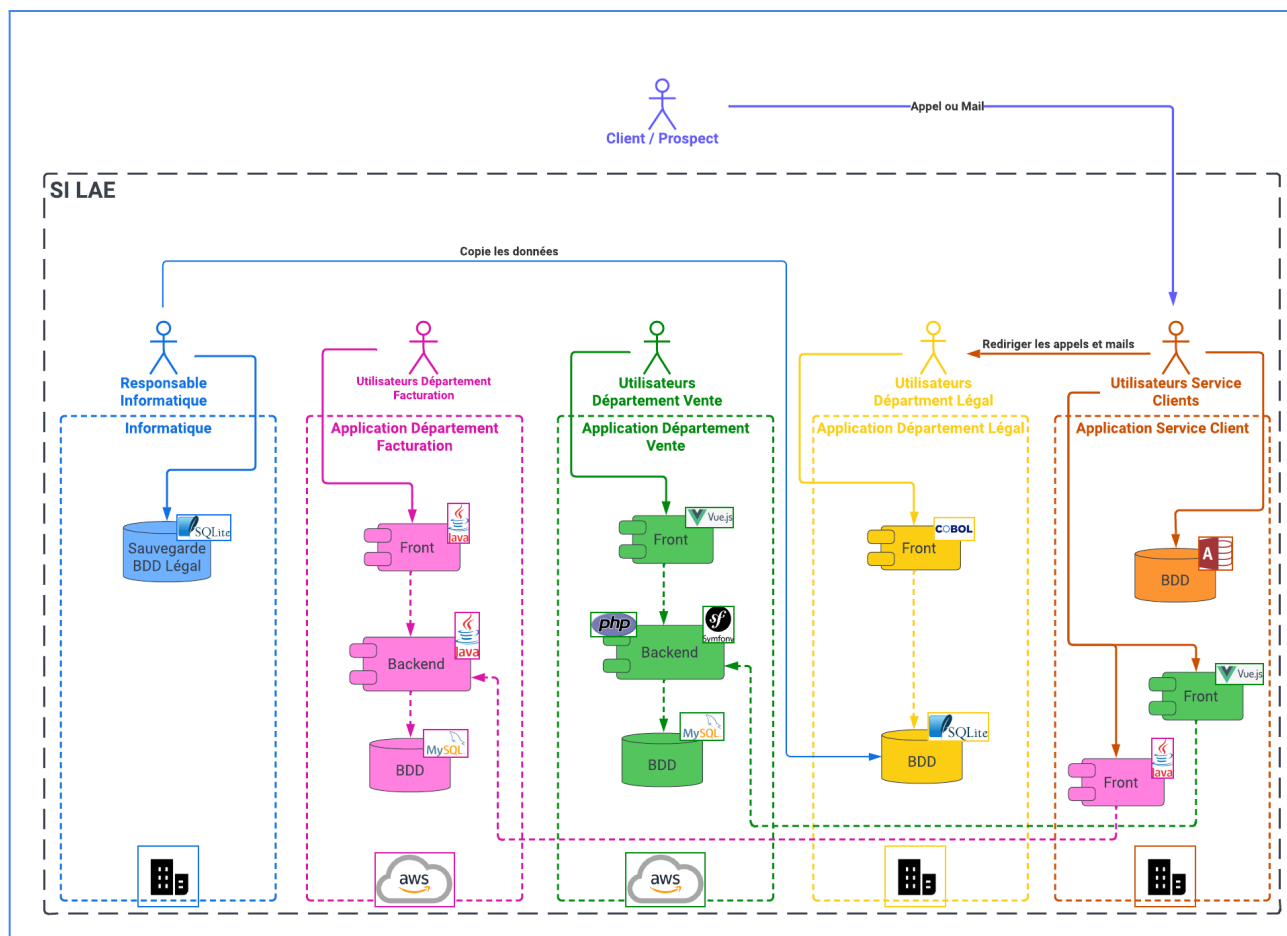
## Diagramme de l'architecture applicative initiale





# DOCUMENT DE DÉFINITION D'ARCHITECTURE

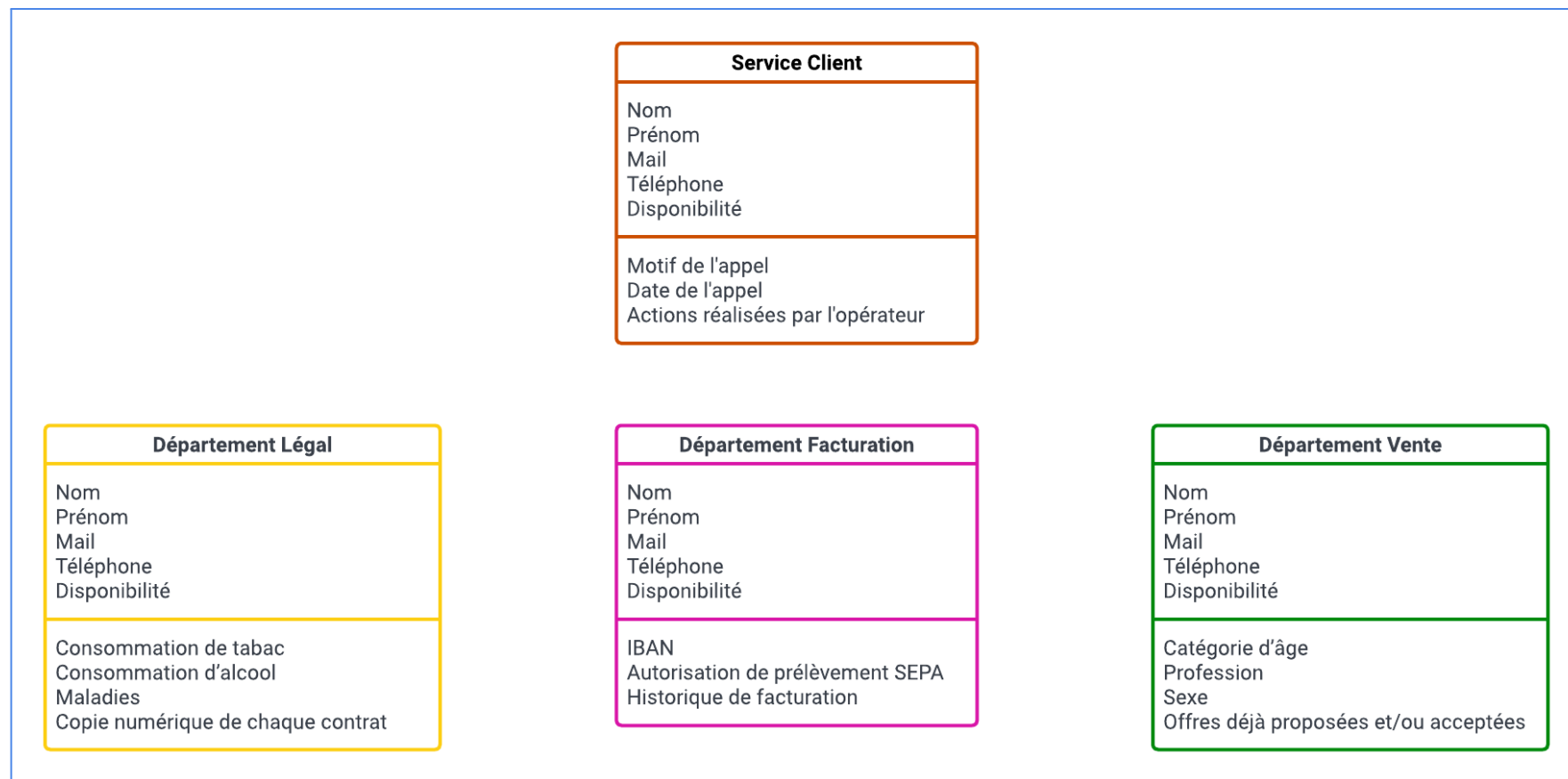
## Diagramme de l'architecture technique initiale





## DOCUMENT DE DÉFINITION D'ARCHITECTURE

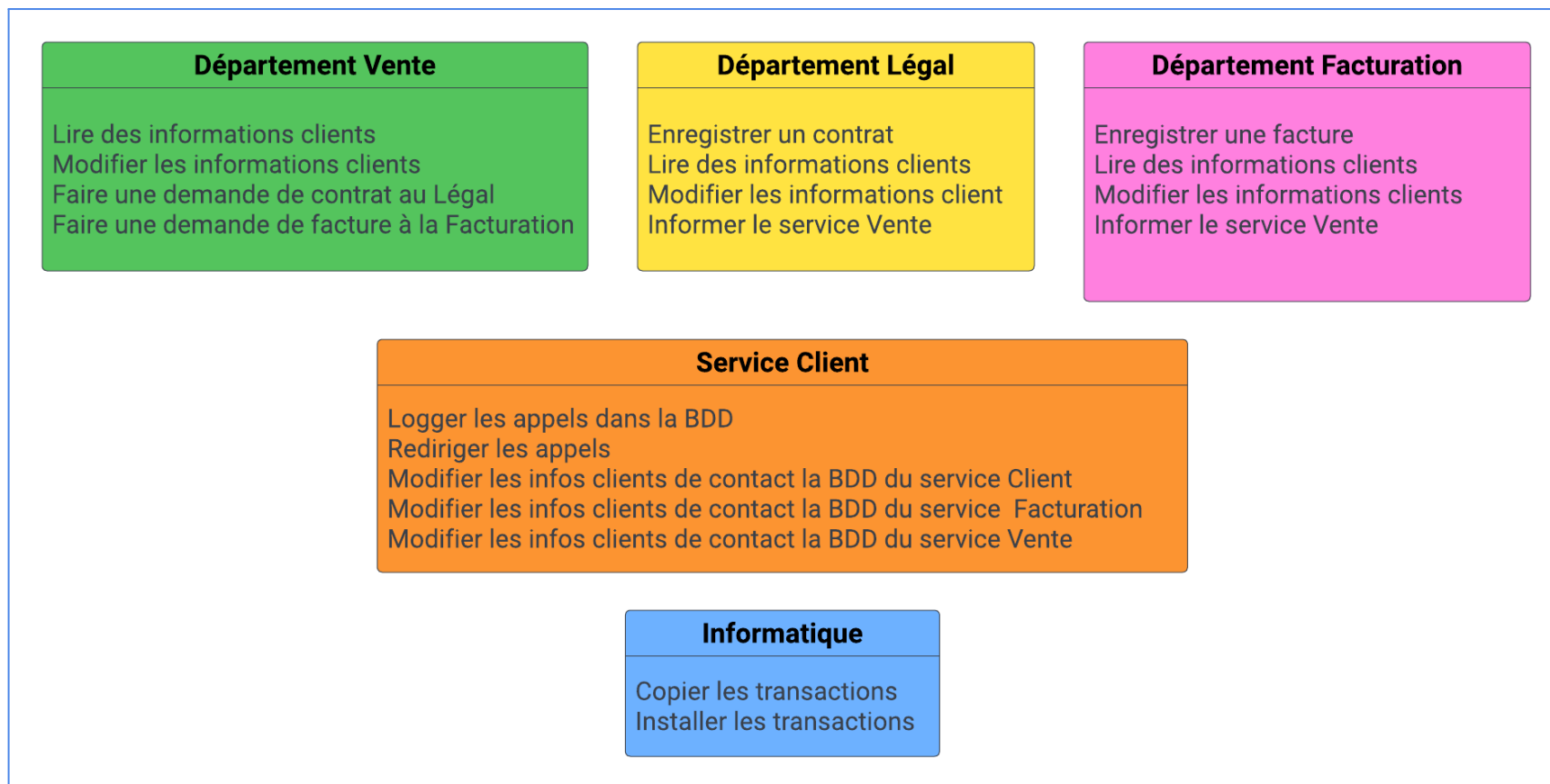
### *Diagramme de l'architecture des données initiale*





## DOCUMENT DE DÉFINITION D'ARCHITECTURE

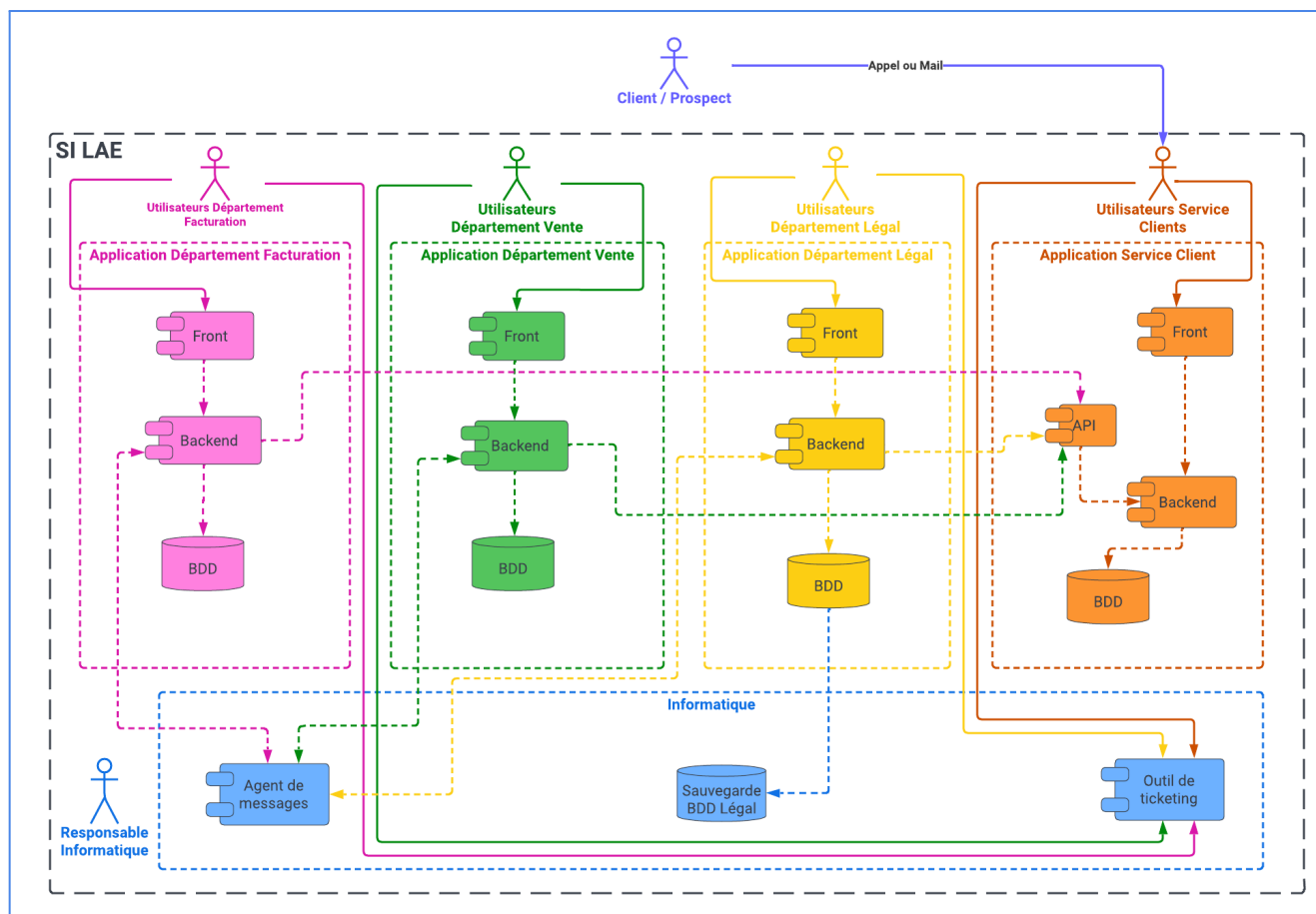
### *Diagramme de l'architecture métier initiale*





# DOCUMENT DE DÉFINITION D'ARCHITECTURE

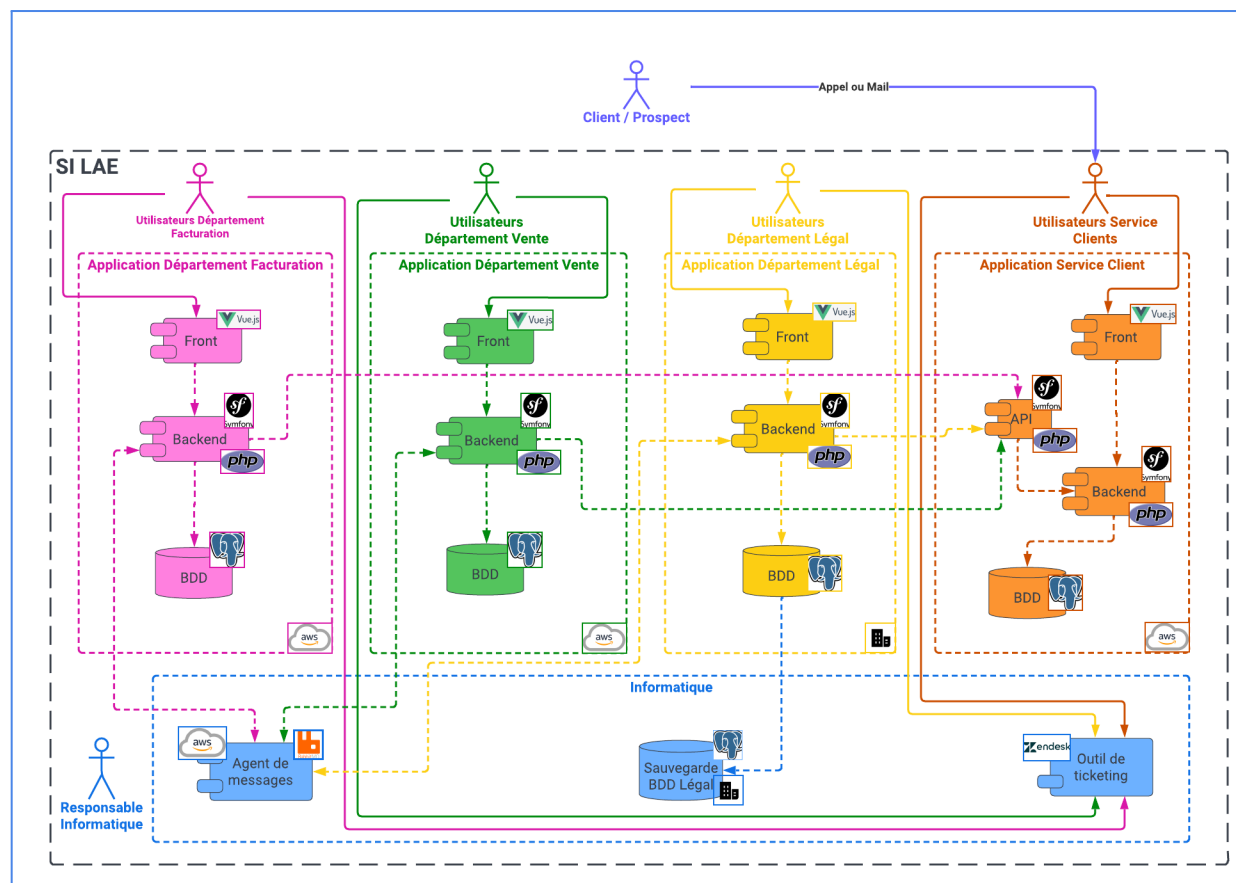
## Diagramme de l'architecture applicative cible





# DOCUMENT DE DÉFINITION D'ARCHITECTURE

## Diagramme de l'architecture technique cible

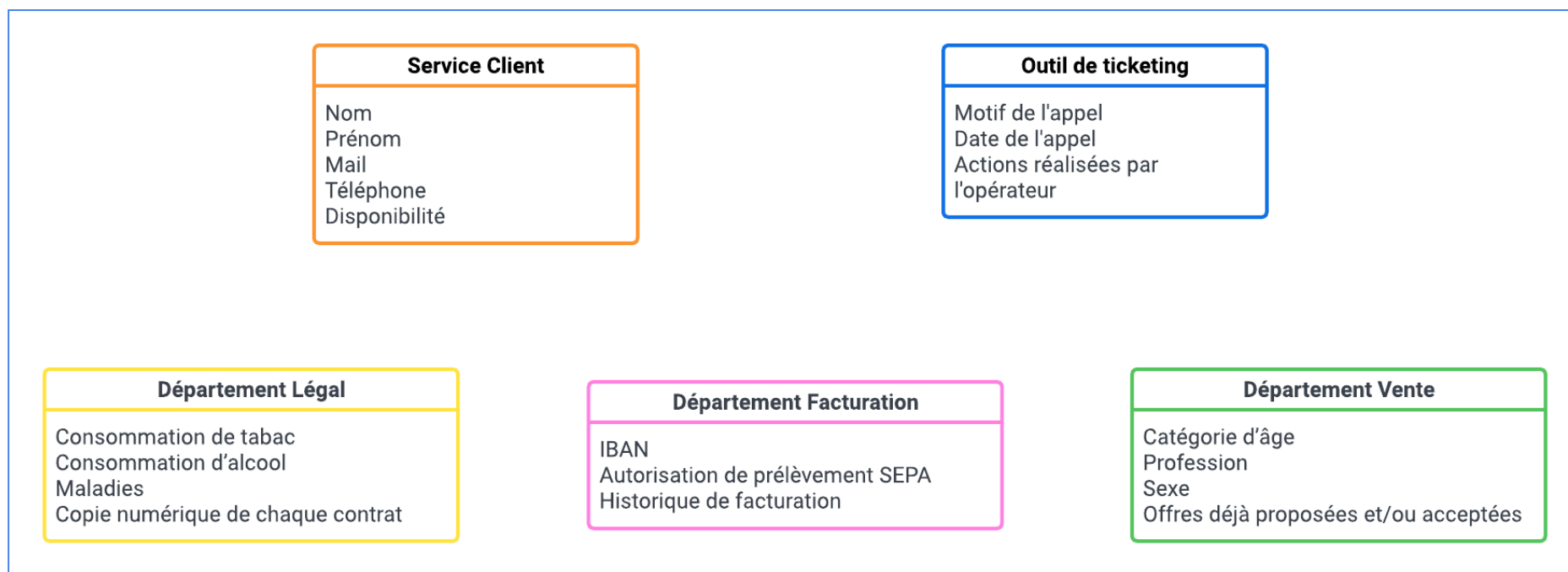






## DOCUMENT DE DÉFINITION D'ARCHITECTURE

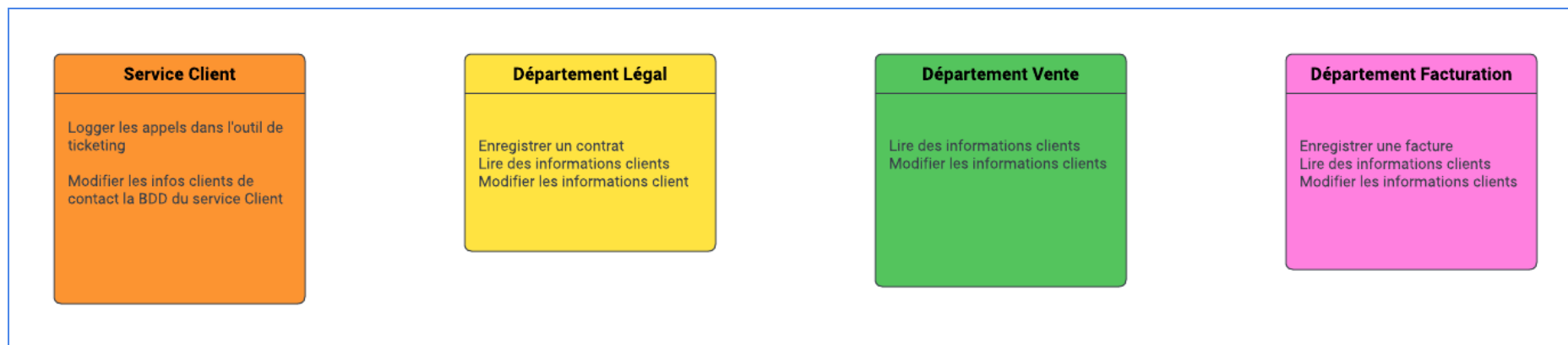
### *Diagramme de l'architecture des données cible*





## DOCUMENT DE DÉFINITION D'ARCHITECTURE

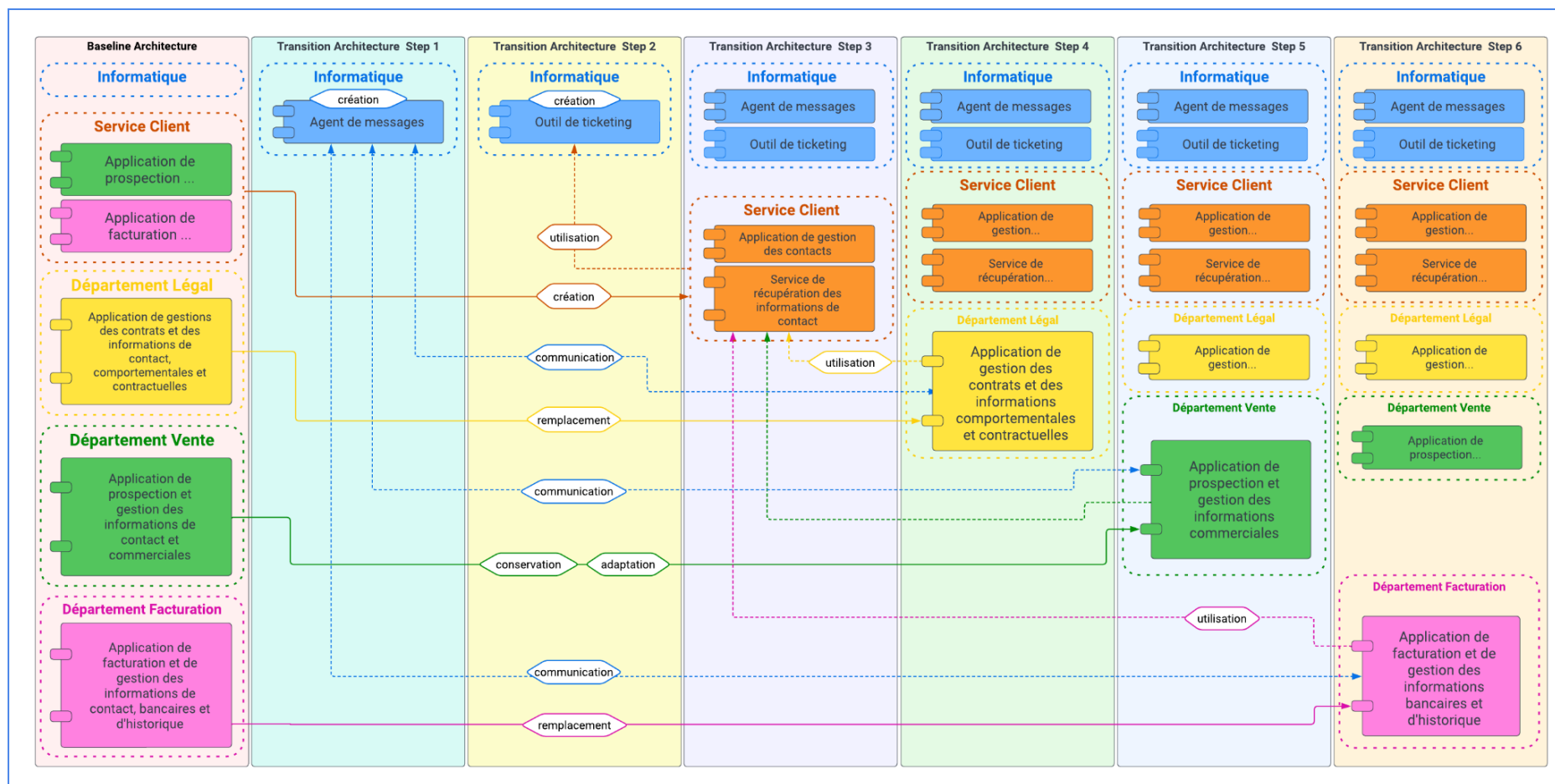
### *Diagramme de l'architecture métier cible*





## DOCUMENT DE DÉFINITION D'ARCHITECTURE

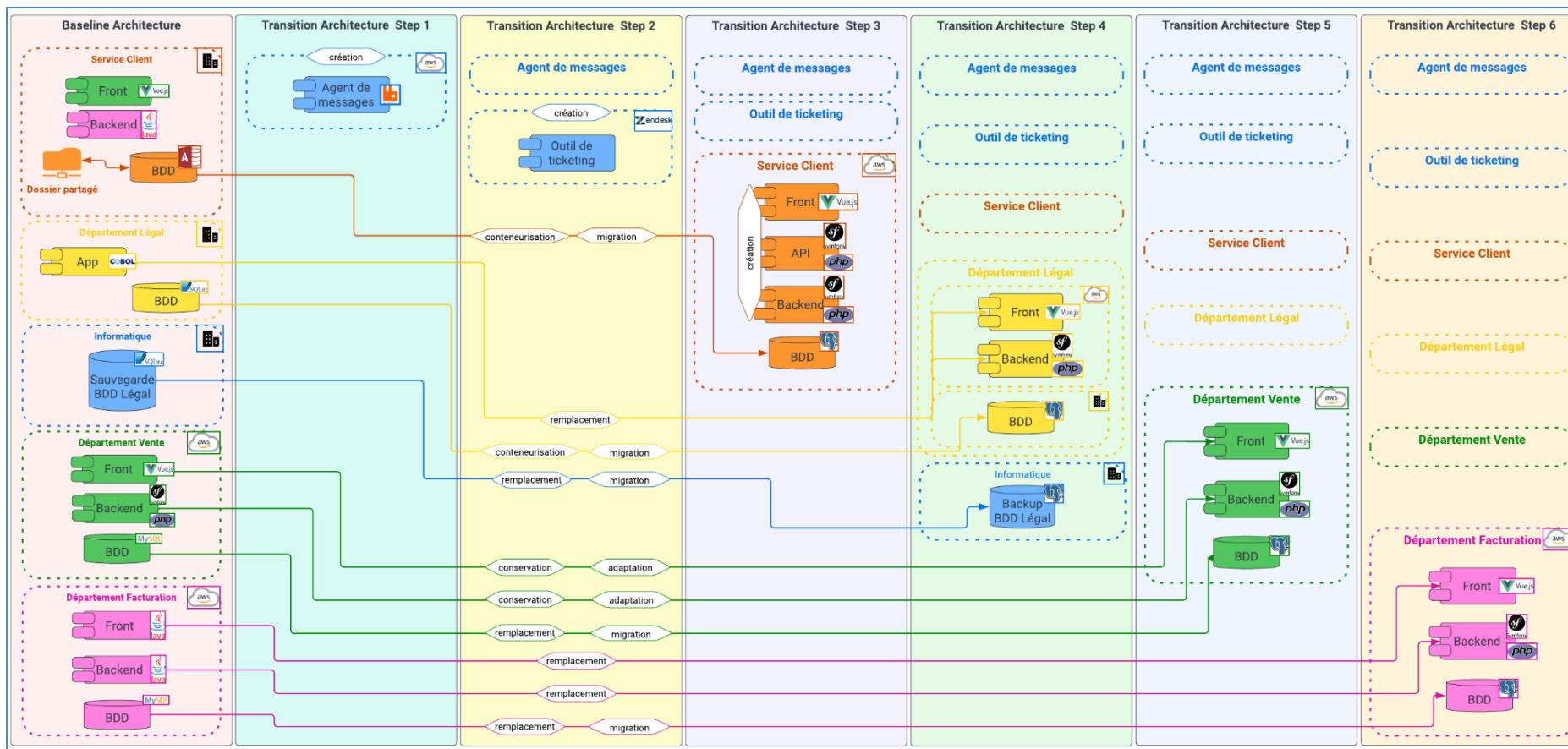
### Diagramme de transition applicative





# DOCUMENT DE DÉFINITION D'ARCHITECTURE

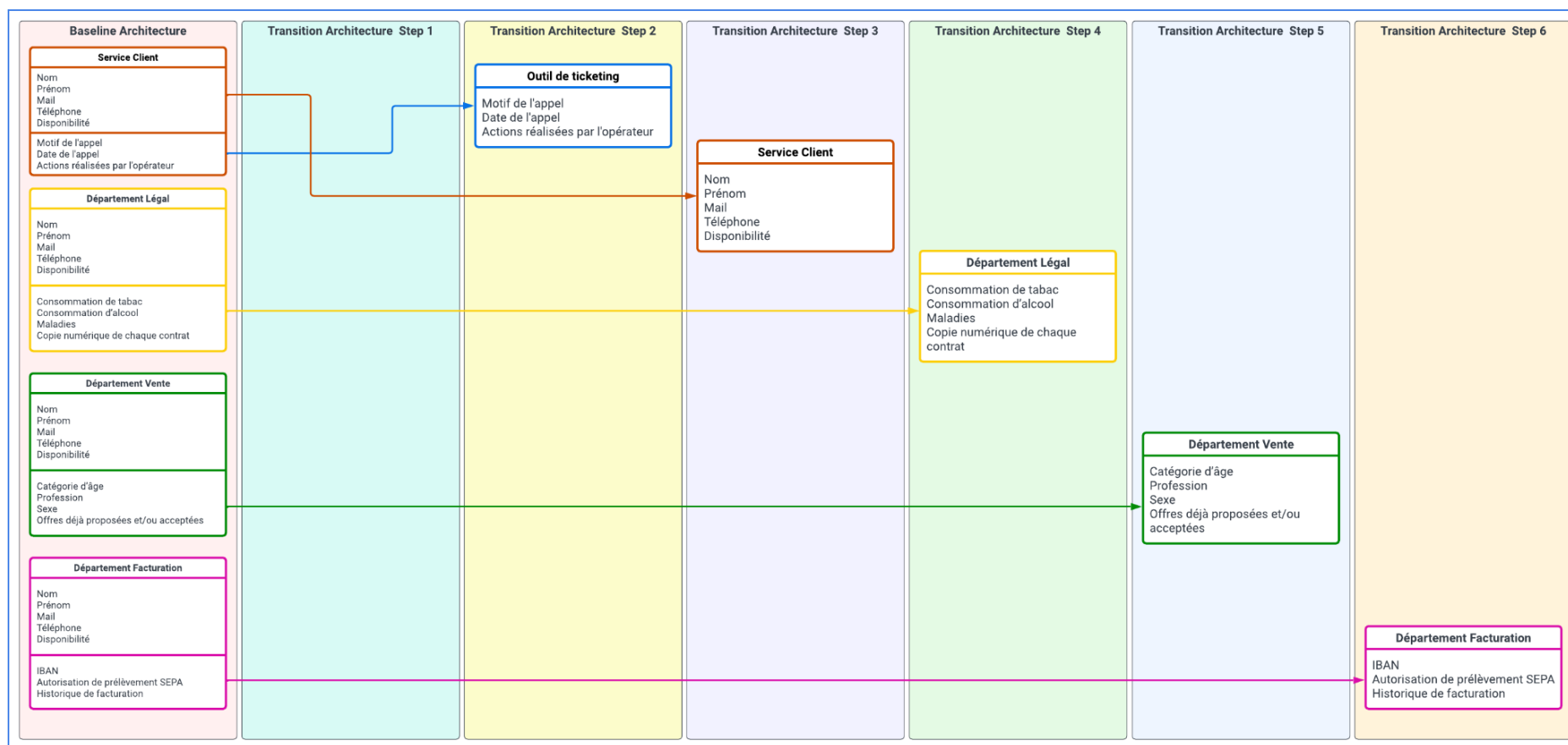
## Diagramme de transition technologique





# DOCUMENT DE DÉFINITION D'ARCHITECTURE

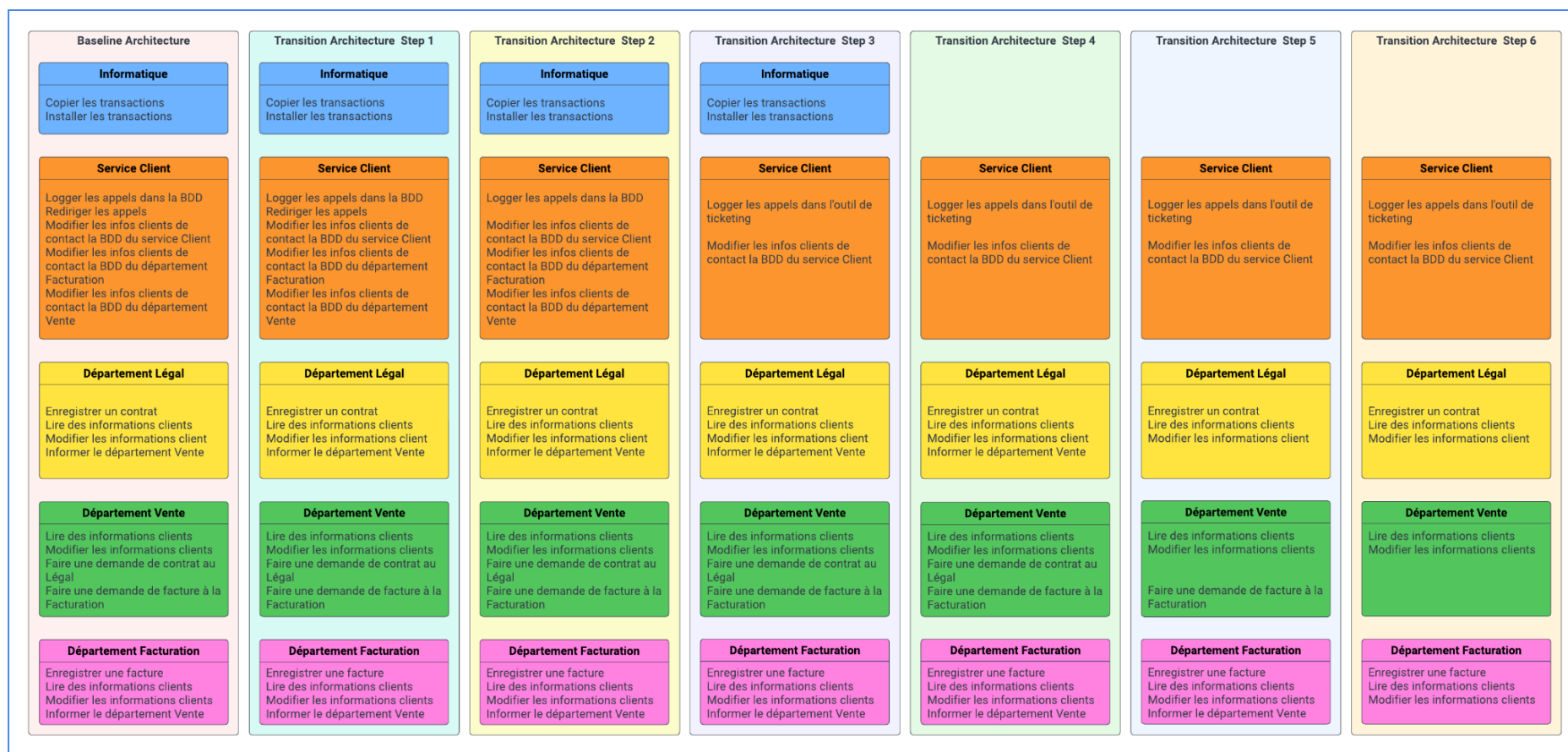
## Diagramme de transition des données





# DOCUMENT DE DÉFINITION D'ARCHITECTURE

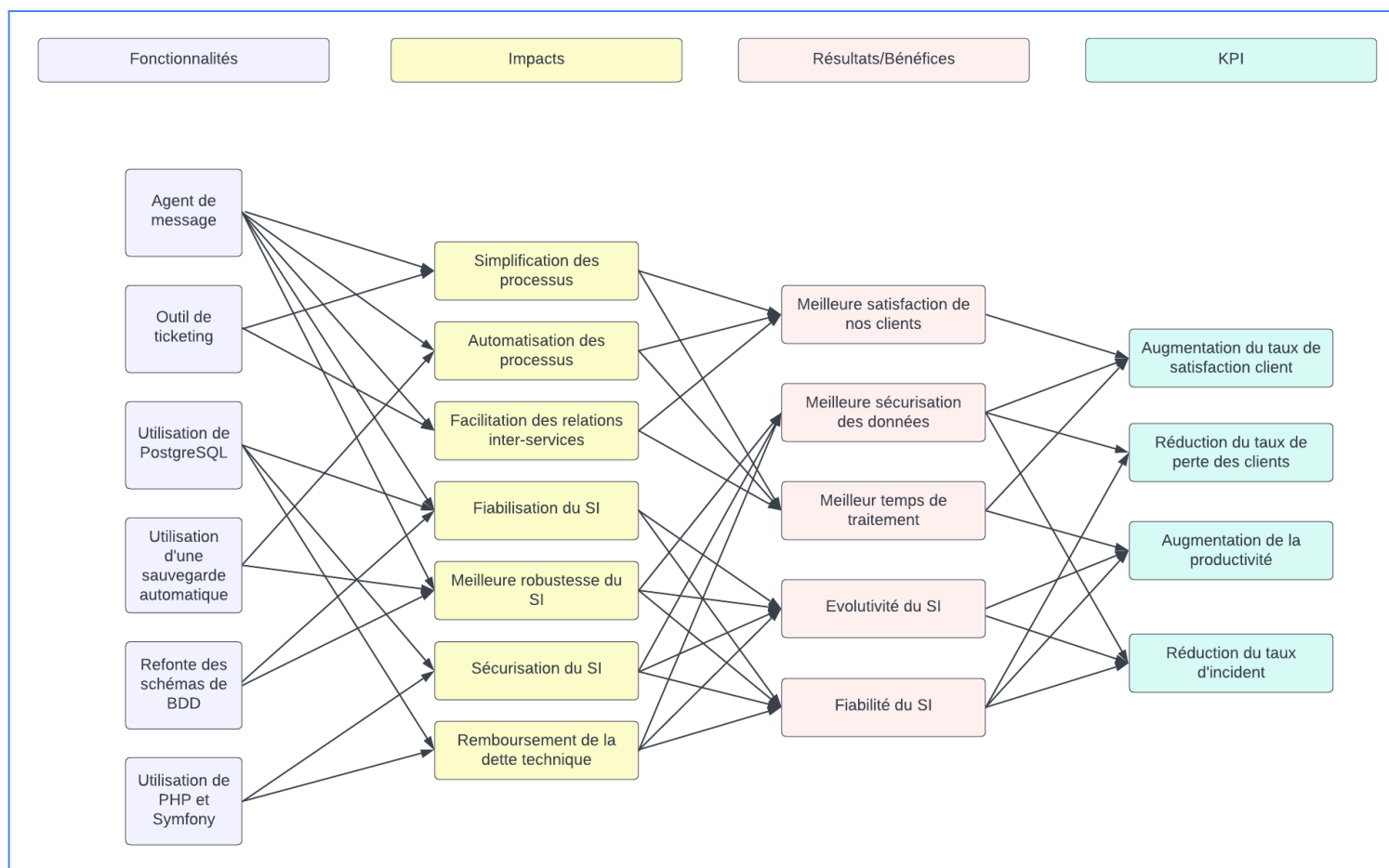
## Diagramme de transition métier





# DOCUMENT DE DÉFINITION D'ARCHITECTURE

## Diagramme des bénéfices





## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### *Catalogue des principes d'architecture*

Architecture Principles				
ID	Name	Description	Date	Category
PRN_ARC_01	Pas de redondance des données	Les données clients redondantes dans plusieurs BDD doivent être regroupées dans une seule BDD		
PRN_ARC_02	Pas de relations inter-services manuelles	Les processus nécessitant une relation entre les services/départements doivent s'appuyer sur un outil permettant d'automatiser les tâches de communication entre les services/départements		
PRN_ARC_03	Continuité de service	L'évolution de l'architecture doit se faire par des transitions permettant la continuité de service		
PRN_ARC_04				
PRN_ARC_05				





## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### *Catalogue des principes métiers*

Business Principles				
ID	Name	Description	Date	Category
PRN_BU_01	Séparation des responsabilités	Les processus nécessitant l'utilisation des applications ou des données des autres services/départements doivent être automatisées		
PRN_BU_02	Exécution des processus métiers	L'exécution des processus métiers ne doit pas être empêchée par la nouvelle architecture		
PRN_BU_03				
PRN_BU_04				
PRN_BU_05				



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### *Catalogue des principes de données*

Data / Information Principles				
ID	Name	Description	Date	Category
PRN_IN_01	Séparation des responsabilités	Les informations clients doivent être réparties dans les tables en fonction des services:départements qui en ont la responsabilité		
PRN_IN_02	Sécurité des données	Les données dont le département Légal est responsable ne doivent pas être exposées sur le réseau		
PRN_IN_03	Sauvegarde des données	Les données doivent pouvoir être sauvegardées sans intervention humaine		
PRN_IN_04				
PRN_IN_05				



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### *Catalogue des principes applicatifs*

Application Principles				
ID	Name	Description	Date	Category
PRN_AP_01	Séparation des responsabilités	Les applications métiers doivent être utilisées uniquement par le service/département concerné par le métier		
PRN_AP_02				
PRN_AP_03				
PRN_AP_04				
PRN_AP_05				



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### *Catalogue des principes technologiques*

Technology Principles				
ID	Name	Description	Date	Category
PRN_TL_01	Automatisation	L'architecture doit se baser un agent de message afin d'automatiser les tâches nécessitant une relation "manuelle" entre les services/départements		
PRN_TL_02	Pile technologique	La pile technologique doit être réduite afin de réduire les coûts de maintenance et de fiabiliser et sécuriser le SI		
PRN_TL_03	Pile technologique	La modification des applications legacy doit se faire en utilisant des technologies permettant de rembourser la dette technique		
PRN_TL_04	Langages de programmation	Les langages de programmation utilisés dans la pile technologique doivent être maintenus et faire l'objet de mises à jour régulières		
PRN_TL_05	SGBDR	Les SGBDR utilisés dans la pile technologique doivent pouvoir être utilisés de manière hybride (cloud et local), doivent être maintenus et faire l'objet de mises à jour régulières et doivent contenir une fonctionnalité de sauvegarde automatique paramétrable		



## DOCUMENT DE DÉFINITION D'ARCHITECTURE

### *Catalogue des principes de sécurité*

Security Principles				
ID	Name	Description	Date	Category
PRN_SE_01	Langages de programmation	Les langages de programmation utilisés dans la pile technologique ne doivent pas comporter de failles de sécurité		
PRN_SE_02	Données	Les données sensibles ne doivent pas être exposées sur le réseau		
PRN_SE_03				
PRN_SE_04				
PRN_SE_05				