**DOCUMENT DE DÉFINITION D’ARCHITECTURE**

**- LES ASSUREURS ENGAGÉS -**

****

**David EVAN**

**06/01/2022**

**Version 1.0**

**Document de définition d’architecture – Les Assureurs Engagés**

**Historique des révisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Numéro de version** | **Auteur** | **Description** | **Date de modification** |
| 1.0 | EVAN David  *(Architecte logiciel)* | Livraison initiale | XX XX 2021 |
|  |  |  |  |
|  |  |  |  |

Tableau 1 : Historique des révisions

**Objectif du document**

Ce document de définition de l'architecture rappel le contexte, les objectifs et les contraintes de ce chantier d’architecture. Il fournit aussi une vue d’ensemble des différents états de l’architecture (de référence, cible et de transition).

Quatre axes d’architecture y seront abordés :

* L’architecture métier (business layer)
* L’architecture des données (data layer)
* L’architecture logicielle (application layer)
* L’architecture en termes d’infrastructure et de technologie (physical & technologies layer)

Certaines notions de ce document supposent une connaissance préalable des précédents livrables réalisés dans le cadre de la phase préliminaire de chantier d’architecture à savoir :

* L’organigramme de l’entreprise
* L’audit technique
* Le cahier des charges
* Le dossier de définition des processus.

**Note :**

La majeure partie des modèles présentés sont issus d’un projet « Archi » auquel il est possible d’accéder à partir du lien ci-après :

**Table des matières**

[CONTEXTE & PORTÉE 5](#_Toc92431542)

[L’entreprise 5](#_Toc92431543)

[Contexte du changement d’architecture 5](#_Toc92431544)

[Cadre et portée des modifications 6](#_Toc92431545)

[OBJECTIFS & CONTRAINTES 7](#_Toc92431546)

[Objectifs 7](#_Toc92431547)

[Contraintes 7](#_Toc92431548)

[ARCHITECTURE DE RÉFÉRENCE 8](#_Toc92431549)

[Organisation de l’entreprise 8](#_Toc92431550)

[Cartographie macroscopique de l’entreprise 8](#_Toc92431551)

[Architecture métier (business layer) 9](#_Toc92431552)

[Département service client 9](#_Toc92431553)

[Processus : Répondre à un client 9](#_Toc92431554)

[Département légal 10](#_Toc92431555)

[Processus : Enquêter 10](#_Toc92431556)

[Processus : Établir un contrat 11](#_Toc92431557)

[Département facturation 12](#_Toc92431558)

[Processus : Établir une facture 12](#_Toc92431559)

[Département vente 13](#_Toc92431560)

[Processus : Définir une stratégie commerciale 13](#_Toc92431561)

[Processus : Prospecter des clients 14](#_Toc92431562)

[Processus : Vendre un produit d’assurance 15](#_Toc92431563)

[Service Informatique 16](#_Toc92431564)

[Processus : Sauvegarder quotidiennement les données 16](#_Toc92431565)

[Évaluation des processus business 17](#_Toc92431566)

[Architecture des données (data layer) 18](#_Toc92431567)

[Cartographie des données métiers et des accès 18](#_Toc92431568)

[Évaluation de l’architecture des données 19](#_Toc92431569)

[Accès et mises à jour des données 19](#_Toc92431570)

[Sécurité des données 19](#_Toc92431571)

[Résilience des données 19](#_Toc92431572)

[Architecture applicative (application layer) 20](#_Toc92431573)

[Cartographie des applicatifs de l’entreprise 20](#_Toc92431574)

[Évaluation des applicatifs 21](#_Toc92431575)

[Architecture technique (technical layer) 22](#_Toc92431576)

[Cartographie de la plateforme technique 22](#_Toc92431577)

[Évaluation de la plateforme technique 22](#_Toc92431578)

[APPROCHE ARCHITECTURALE 23](#_Toc92431579)

[Synthèse de l’évaluation de l’architecture de référence 23](#_Toc92431580)

[Points fort de l’architecture de référence 23](#_Toc92431581)

[Points faibles de l’architecture de référence 23](#_Toc92431582)

[Approche architecturale retenue 24](#_Toc92431583)

[Prototype de l’architecture cible 24](#_Toc92431584)

[Justification de l’approche architecturale 25](#_Toc92431585)

[Choix des technologies (PHP/Symfony – Vue.JS) 25](#_Toc92431586)

[Logiciels accessibles sous forme de Web-App (SaaS) 25](#_Toc92431587)

[Sécurité des flux avec OAuth / OIDC 25](#_Toc92431588)

[Fédération des API autours d’une API Gateway 26](#_Toc92431589)

[Outil de ticketing externe 26](#_Toc92431590)

[Exigences de l’architecture cible 27](#_Toc92431591)

[ARCHITECTURE CIBLE 29](#_Toc92431592)

[Architecture métier (business layer) 29](#_Toc92431593)

[Département service client 29](#_Toc92431594)

[Processus : Répondre à un client 29](#_Toc92431595)

[Service Informatique 30](#_Toc92431596)

[Processus : Contrôler trimestriellement la sauvegarde des données et tester la restauration 30](#_Toc92431597)

[Architecture des données (data layer) 31](#_Toc92431598)

[Cartographie des données métiers dans l’entreprise 31](#_Toc92431599)

[Sécurité des données et contrôle d’accès 32](#_Toc92431600)

[Architecture applicative (application layer) 33](#_Toc92431601)

[Cartographie des applicatifs 33](#_Toc92431602)

[Catalogue des composants applicatifs 35](#_Toc92431603)

[Architecture technique (technical layer) 36](#_Toc92431604)

[Cartographie de la plateforme technique 36](#_Toc92431605)

[Coût d’exploitation de la plateforme technique 37](#_Toc92431606)

[ANALYSE DES ÉCARTS 38](#_Toc92431607)

[Écarts d’architecture 38](#_Toc92431608)

[Business 38](#_Toc92431609)

[Data 38](#_Toc92431610)

[Application 38](#_Toc92431611)

[Technique 38](#_Toc92431612)

[Risques de l’architecture 39](#_Toc92431613)

[ARCHITECTURE DE TRANSITION 40](#_Toc92431614)

[ANNEXES 41](#_Toc92431615)

[Modèle de classification des risques 41](#_Toc92431616)

[APPROBATIONS 42](#_Toc92431617)

[TABLES DES RÉFÉRENCES 43](#_Toc92431618)

[Figures 43](#_Toc92431619)

[Tableaux 43](#_Toc92431620)

# CONTEXTE & PORTÉE

## L’entreprise

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.

L’entreprise se décompose en quatre services que sont :

* **Le service client**, qui est en charge de réceptionner les appels, modifier ou accéder aux informations client et rediriger des appels vers les autres services.
* **Le service vente**, qui définit les plans d’action commerciale (définition de cibles, de produits et d'événements publicitaires). Ce service réalise aussi l’envoi de devis adaptés aux besoins des clients et aux risques pris par LAE.
* **Le service contrat**, qui établit des contrats pour les besoins du service vente, et enquête sur les comportements des futurs clients.
* **Le service facturation**, qui établit une facture après la signature d’un contrat.

L’entreprise dispose aussi d’un responsable informatique chargé des opérations de maintenance et de sauvegarde. Le CEO, placé au centre de l’organigramme est responsable de la définition de la stratégie de l’entreprise en s’appuyant sur le reporting régulier réalisé par les « leaders » (chefs) de chaque service.

## Contexte du changement d’architecture

L’entreprise a bâti son SI en faisant face aux besoins immédiats et sans s’appuyer sur des normes ou une gouvernance d’architecture commune.

Ceci a conduit à l’élaboration d’unsystème en patchwork dans lequel cohabitent des technologies anciennes (parexemple COBOL) et modernes (PHP, javascript). Aujourd’hui, chaque servicedispose de son propre SI avec une interface et une base de données dédiées,implémentées dans des langages et avec des technologies différentes.

Les utilisateurs et le responsable informatique remontent de nombreuses considérations parmi :

* Des difficultés à assurer une cohérence des données entre plusieurs service,
* Une absence de SI centralisé entrainant des opérations laborieuses pour la transmission d’information entre service,
* Un manque de robustesse du système,
* Des vulnérabilités identifiées dû à l’utilisation de technologie obsolètes.
* Une maintenance rendu difficile par la fragmentation technologique.
* Une absence de contrôle d’accès et de capacités des utilisateurs pouvant conduire à des pertes de données.

Ces constats ont conduit l’entreprise à souhaiter un changement en profondeur dans l’architecture du système d’information afin de permettre de retrouver de la fiabilité, de la rapidité, de la robustesse et de la sécurité dans son SI tout en diminuant la complexité de la maintenance.

## Cadre et portée des modifications

Le cadre des modifications n’inclut pas l’analyse et la représentation des processus associés à la stratégie de l’entreprise (ex : Reporting des leads de service au CEO).

La portée des modifications intègre l’ensemble des couches du système d’information (applicatif, données, technologies et plateforme technique).

Les processus associés au responsable informatique seront modifiés pour s’adapter aux capacités offertes par la nouvelle architecture.

Le service rendu n’étant pas remis en cause dans les constats réalisés, les actions des utilisateurs dans les processus métiers opérationnels ne seront pas ou peu modifiés. (Les interfaces logicielles seront modifiées mais ne remettront pas en cause l’organisation de l’entreprise).

# OBJECTIFS & CONTRAINTES

## Objectifs

Les objectifs formalisés à atteindre sont listés dans le catalogue ci-après :

|  |  |  |
| --- | --- | --- |
| **Id.** | **Objectif** | **Description** |
| **O1** | Vélocité | Diminuer les temps de traitement des opérations (de collaboration ou technique). |
| **O2** | Fiabilité des données | Améliorer la fiabilité des données et le partage dans l’entreprise. Une donnée modifiée à un instant T doit être disponible pour l’intégralité des opérateurs immédiatement. |
| **O3** | Robustesse du système | Le système doit être robuste et les opérations de sauvegarde / restauration doit être réalisable rapidement et avec le minimum d’intervention humaine. |
| **O4** | Sécurité | Aucune faille de sécurité ne doit être présente dans le système. Le système doit intégrer des mécanismes de contrôle d’identification et d’autorisation. |
| **O5** | Maintenance simplifiée | La maintenance du SI doit être simple. La diversité des technologies employées doit être limités aux seuls besoins réels. |

Tableau 2 : Catalogue des objectifs du chantier d’architecture

## Contraintes

Les contraintes associées à ce chantier d’architecture sont listées dans le catalogue ci-après :

|  |  |  |
| --- | --- | --- |
| **Id.** | **Catégorie** | **Description** |
| **CP1** | Technique | Le service client impose l’utilisation d’un outil de ticketing externe. Le choix est laissé libre sous réserve qu’il soit justifié d’un point de vue coût et dette technique. |
| **CP2** | Ressources humaines | LAE met à disposition un ingénieur généraliste en informatique à temps plein (38,5 h/semaine). Toute ressource humaine supplémentaire devra être compris dans la limite du budget. |
| **CP3** | 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. |
| **CP4** | Délais | Le système devra être opérationnel d’ici sept mois à compter de la validation de la solution. Le non-respect des délais entraînera des indemnités compensatrices. |
| **CP5** | Impacts sur les services | L’intégration devra être réalisé de façon continue, de sorte à ne pas perturber le travail des collaborateurs. Le non-respect de ces conditions entraînera des indemnités compensatrices. |

Tableau 3 : Catalogue des contraintes du chantier d’architecture

# ARCHITECTURE DE RÉFÉRENCE

L’architecture de référence (*baseline architecture*) constitue l’état actuelle de l’architecture de l’entreprise LAE. Les modélisations présentées dans cette section sont issues des différents documents précédemment rédigés et remis sous une forme standardisé à l’aide de la grammaire [Archimate](https://pubs.opengroup.org/architecture/archimate3-doc/toc.html).

## Organisation de l’entreprise

### Cartographie macroscopique de l’entreprise

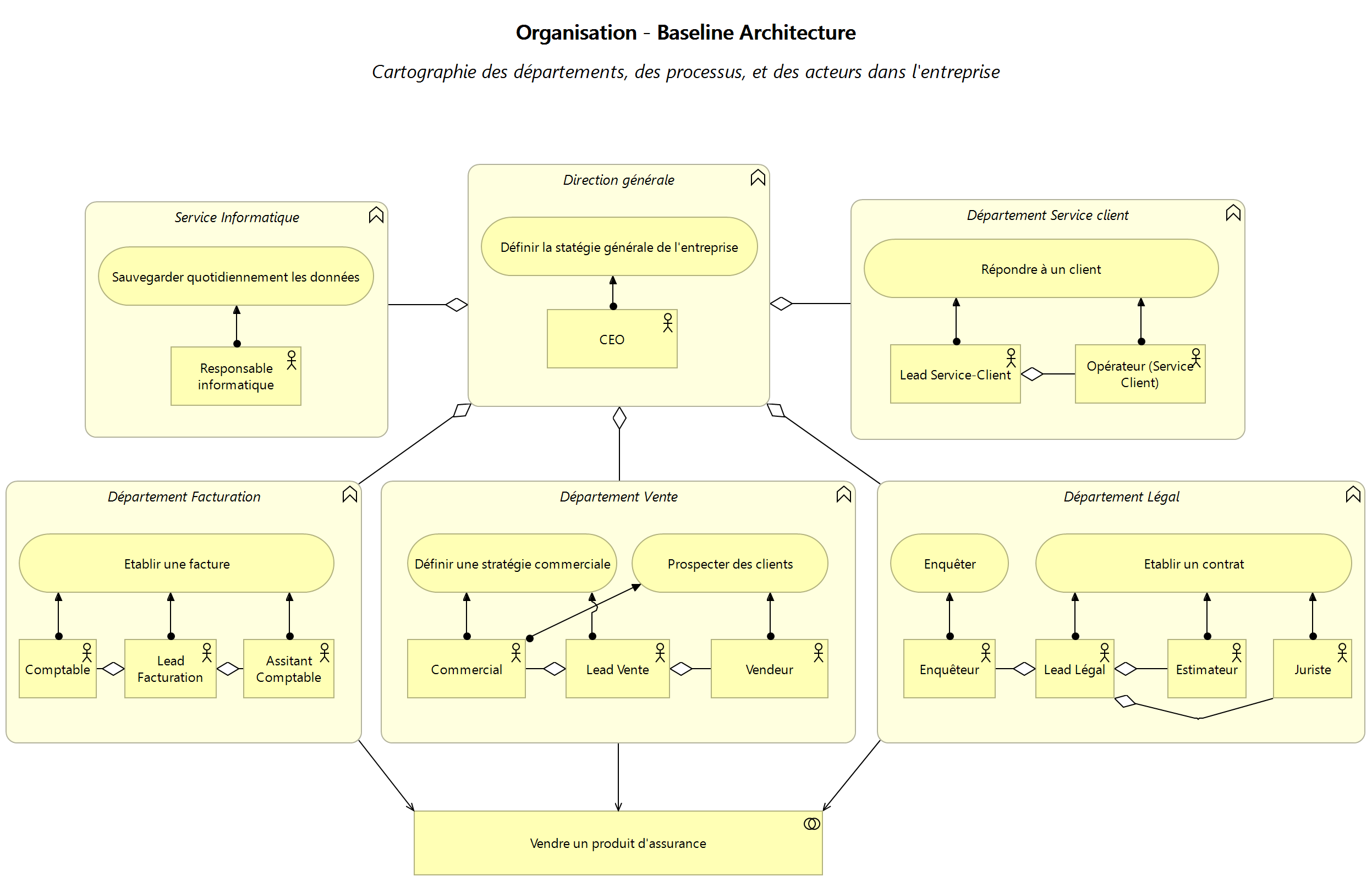


Figure 1 : Baseline Architecture - Cartographie des départements, des processus et des acteurs.

La cartographie macroscopique de l’entreprise, ci-avant présentée, montre l’organisation générale de l’entreprise, les différents départements, les processus et leurs acteurs dans l’entreprise.

## Architecture métier (business layer)

*NB : Les activités de conception d’architecture haut niveau ne nécessitant pas une analyse approfondie des processus de l’entreprise, les diagrammes d’activités BPMN ne seront pas présentés dans le présent document.*

### Département service client

#### Processus : Répondre à un client

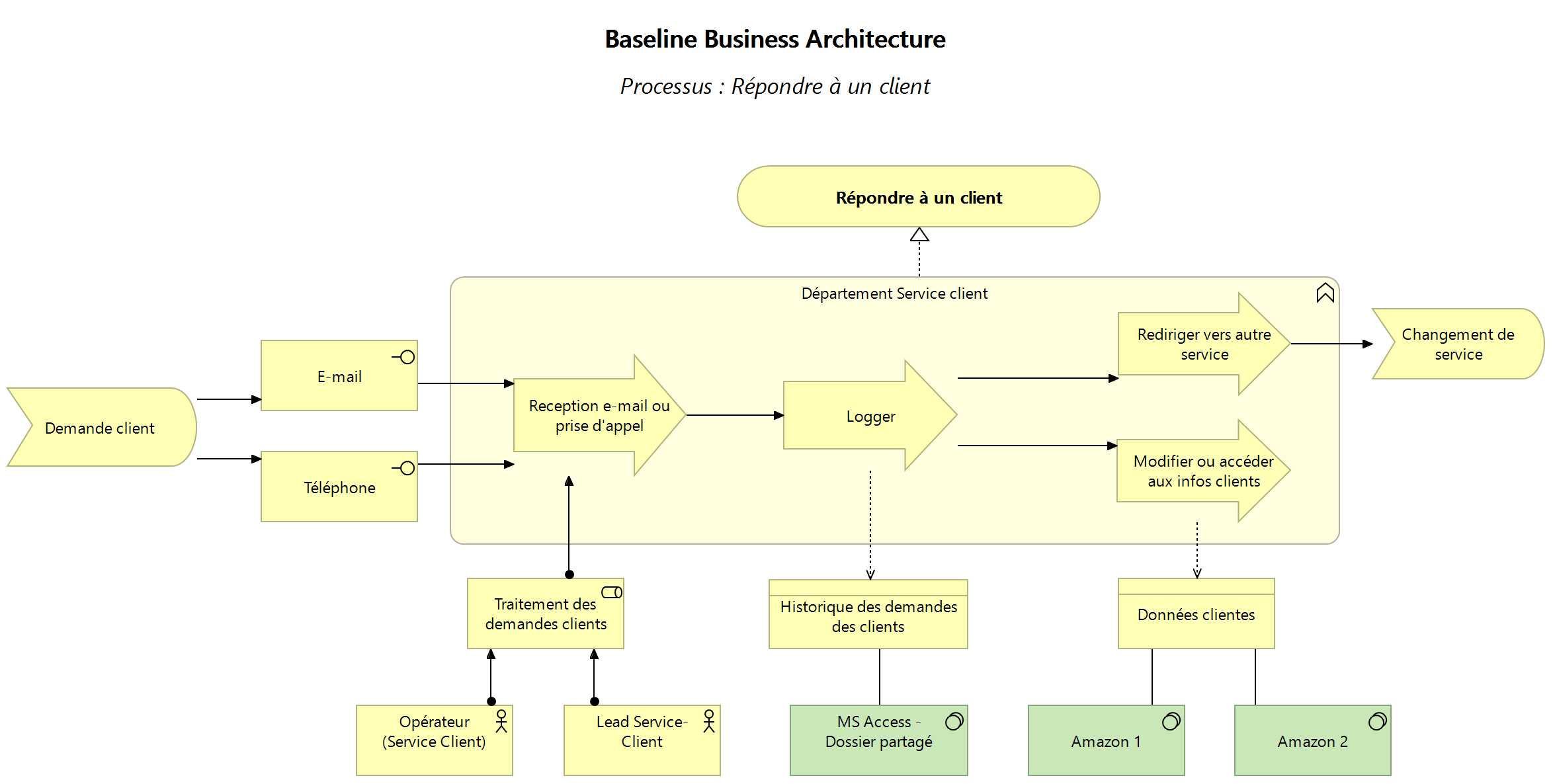


Figure 2 : Business Baseline Architecture - Répondre à un client

Les opérateurs du service client, par l’intermédiaire du processus *« Répondre à un client* » disposent de la capacité à traiter les demandes clients provenant des canaux e-mails ou téléphone. Les demandes clients sont historisés et, en fonction de la demande, l’opérateur transfert le client à un autre service ou accède aux / modifie les informations associées.

### Département légal

#### Processus : Enquêter

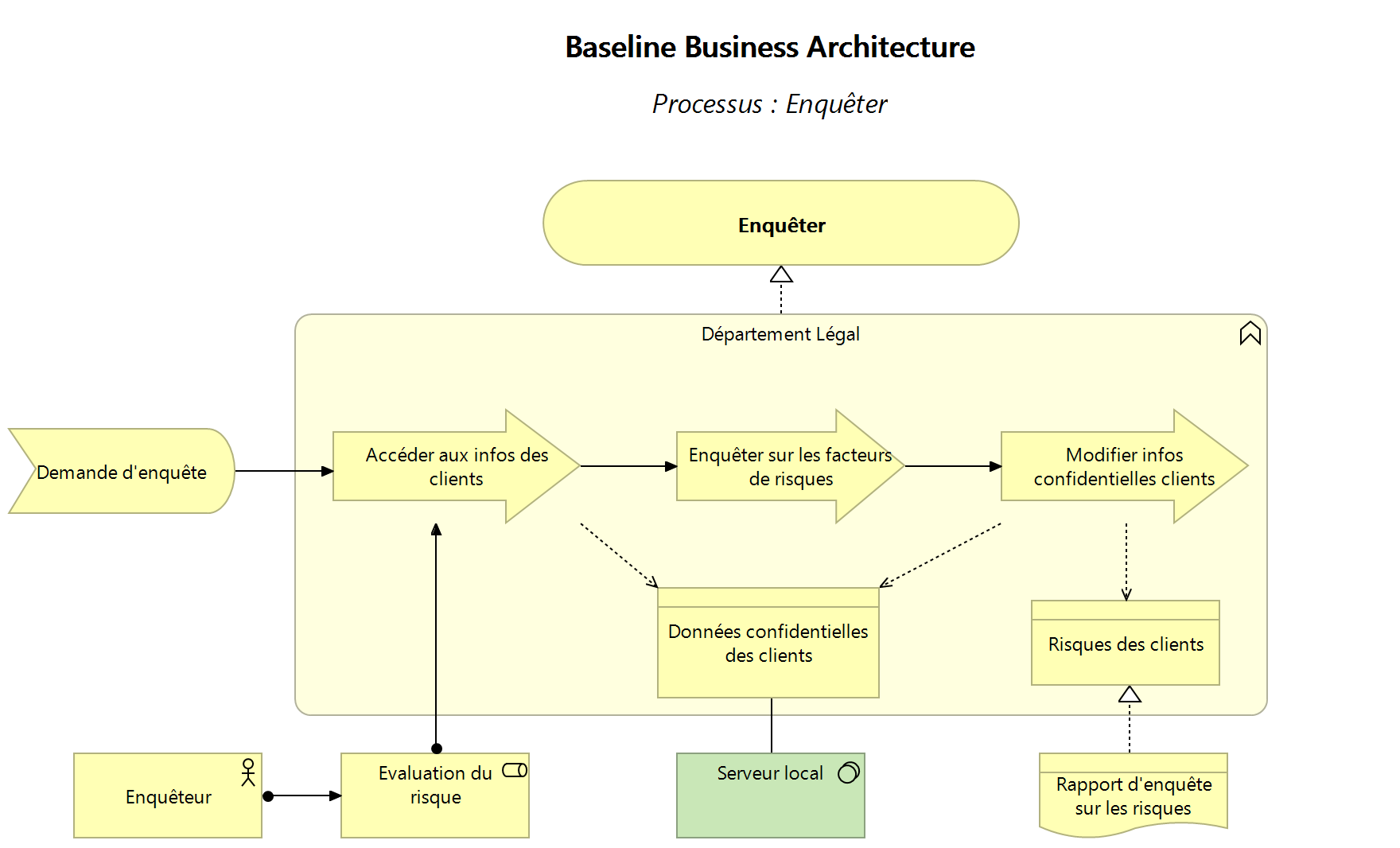


Figure 3 : Business Baseline Architecture - Enquêter

Les enquêteurs du département légal sont chargés des missions d’évaluation du risque qui consistent à déterminer les risques encourus par un client. Dans le cadre de leur mission, ils disposent de la capacité à accéder aux données confidentielles des clients, stockés sur un serveur local dédié au service, et produisent des rapports d’enquête d’évaluation des risques.

Ces rapports de risques sont partagés au sein du service uniquement.

#### Processus : Établir un contrat

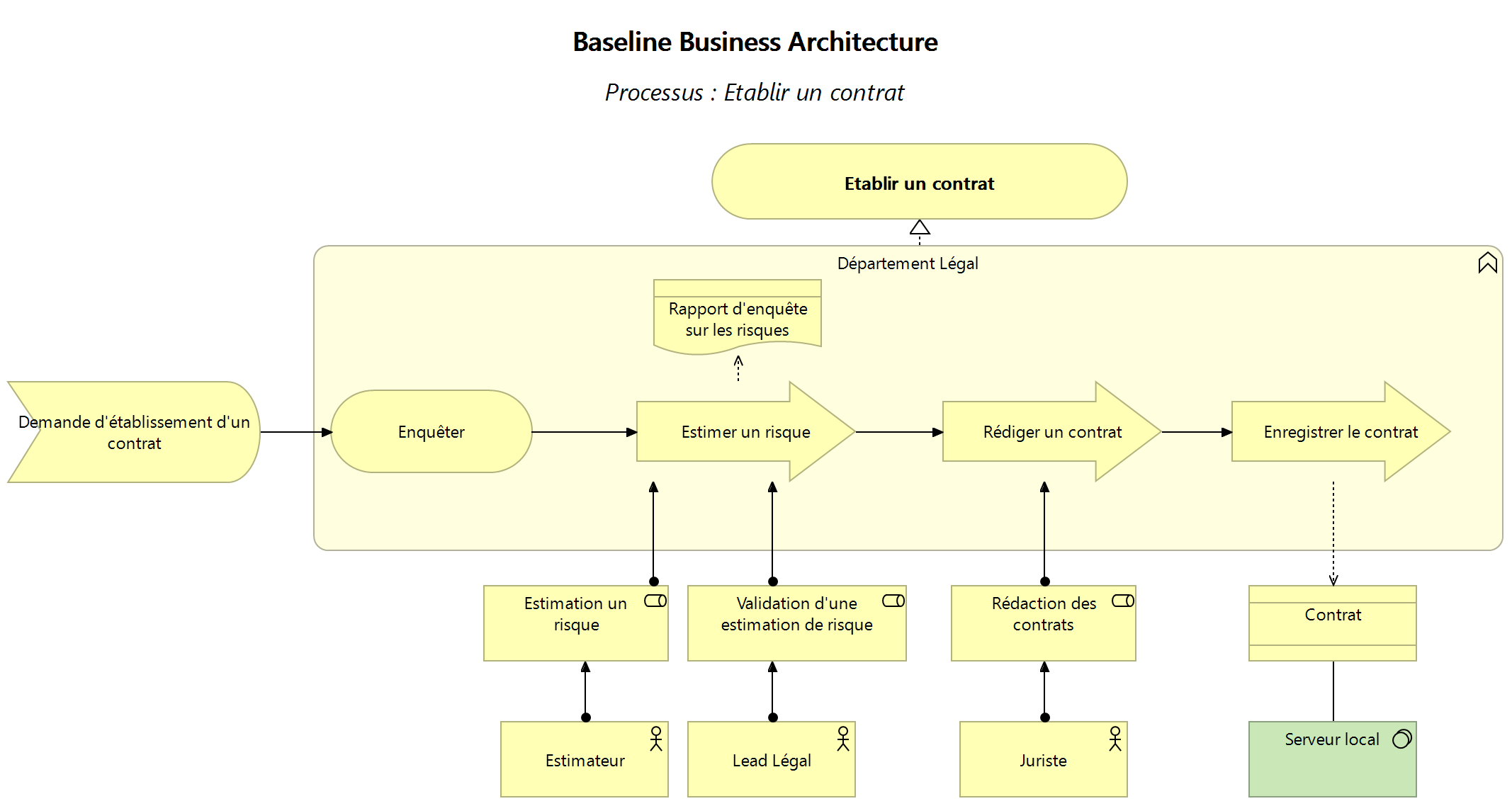


Figure 4 : Business Baseline Architecture - Établir un contrat

L’établissement d’un contrat est réalisé par le département légal au travers du processus « établir un contrat ». Ce processus fait intervenir plusieurs acteurs :

* Les estimateurs, chargés de l’estimation du coût du risque.
* Le responsable de service (« Lead légal »), chargé de la validation des estimations réalisées,
* Les juristes, chargé de la rédaction et de l’enregistrement des contrats.

Les contrats sont stockés sur le serveur local du département à usage exclusif du département.

### Département facturation

#### Processus : Établir une facture

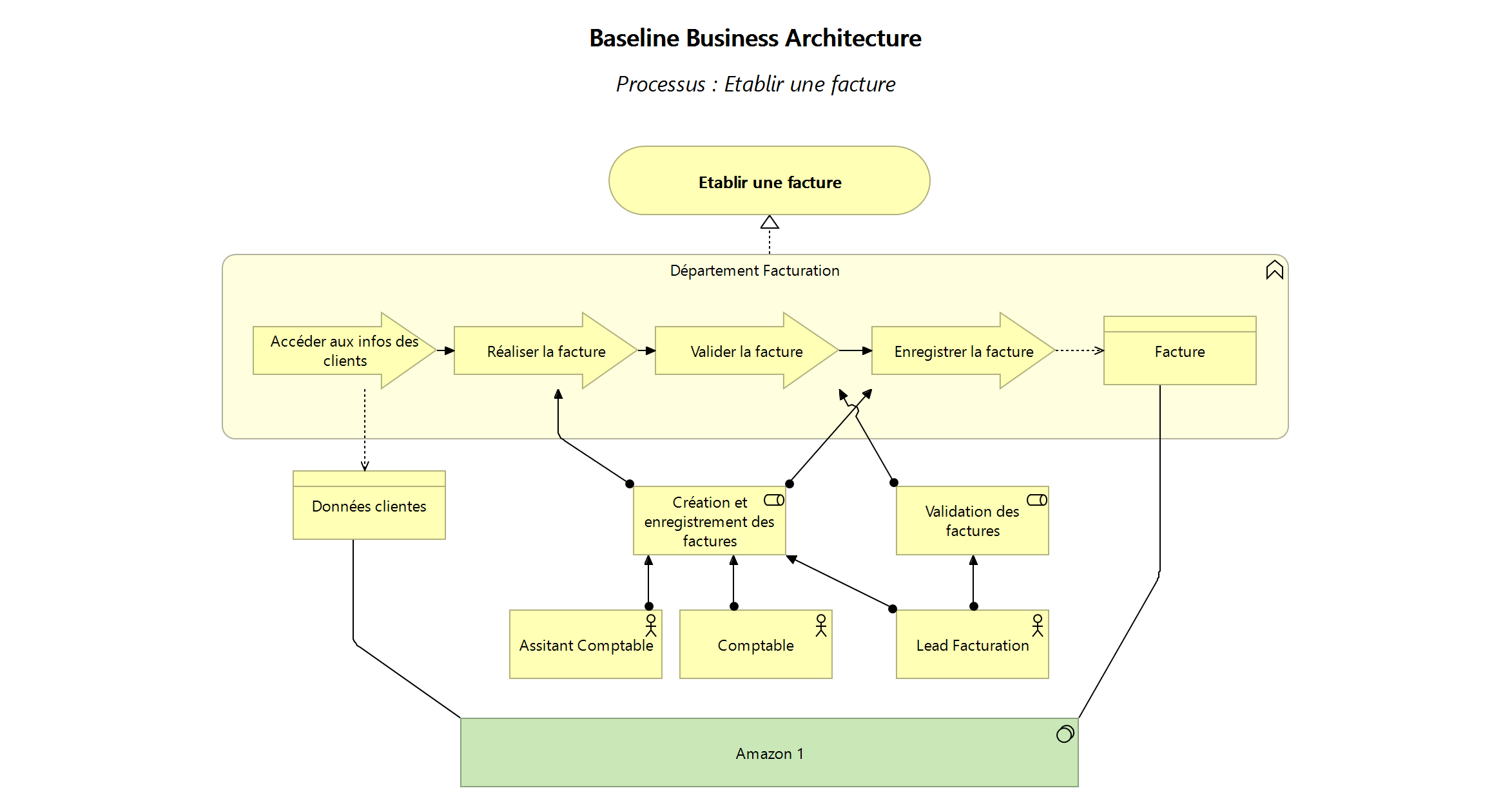


Figure 5 : Business Baseline Architecture - Établir une facture

Le département facturation est chargé de création des factures clients. Les comptables et assistants comptable crées et enregistrent les factures validées qui sont validés par le responsable de service (« Lead facturation »).

Le responsable de service peut être amené à participer aux activités de création des factures en cas de surcharge de travail.

### Département vente

#### Processus : Définir une stratégie commerciale

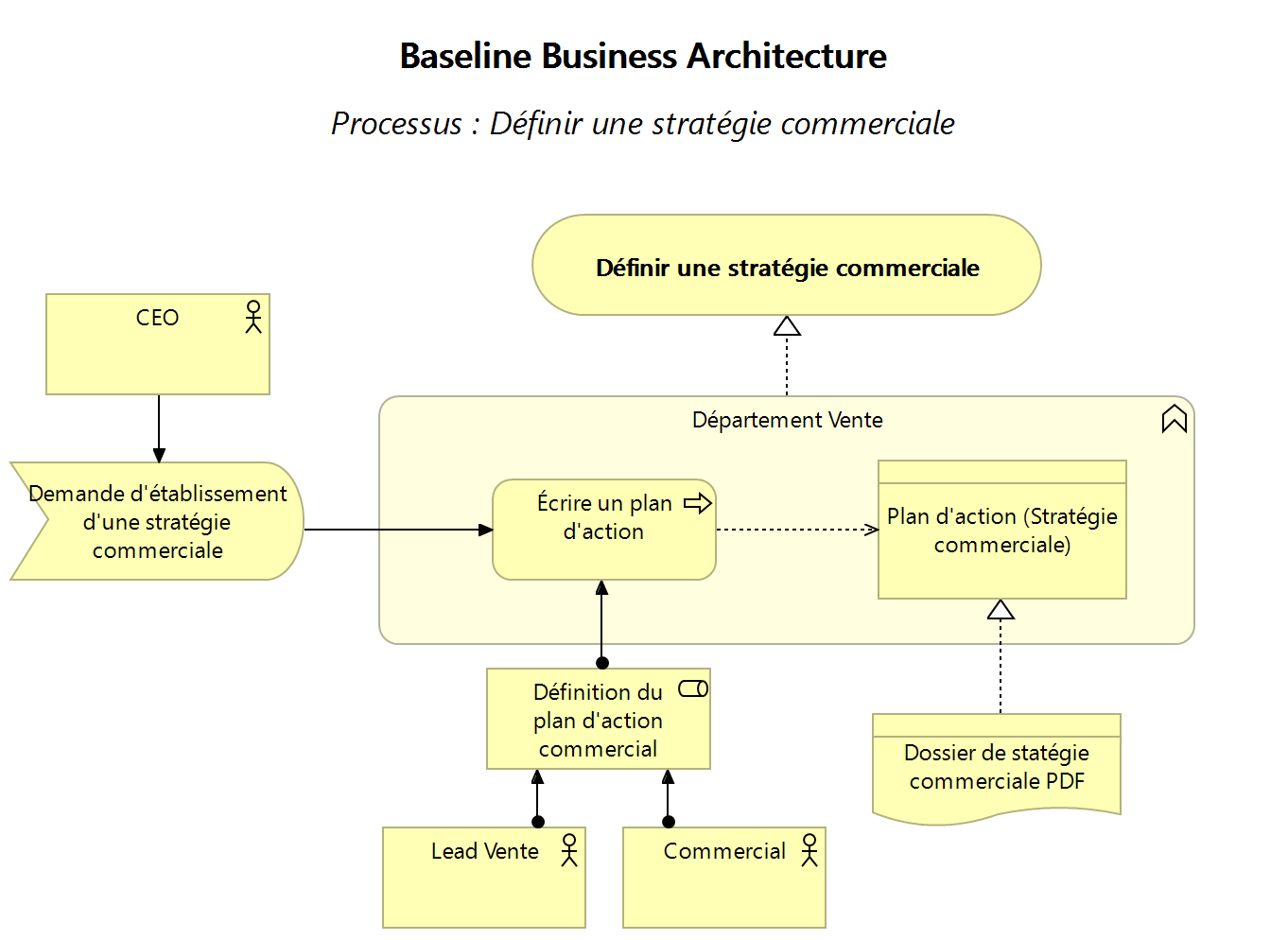


Figure 6 : Business Baseline Architecture - Définir une stratégie commerciale

Le processus de définition de la stratégie commerciale est initié par le CEO. Les commerciaux et le responsable de service sont chargés de la définition du plan d’action commercial qui fournira la méthodologie et les métriques à atteindre pour acquérir des clients.

La stratégie commerciale est livrée sous forme de dossier stratégique.

#### Processus : Prospecter des clients

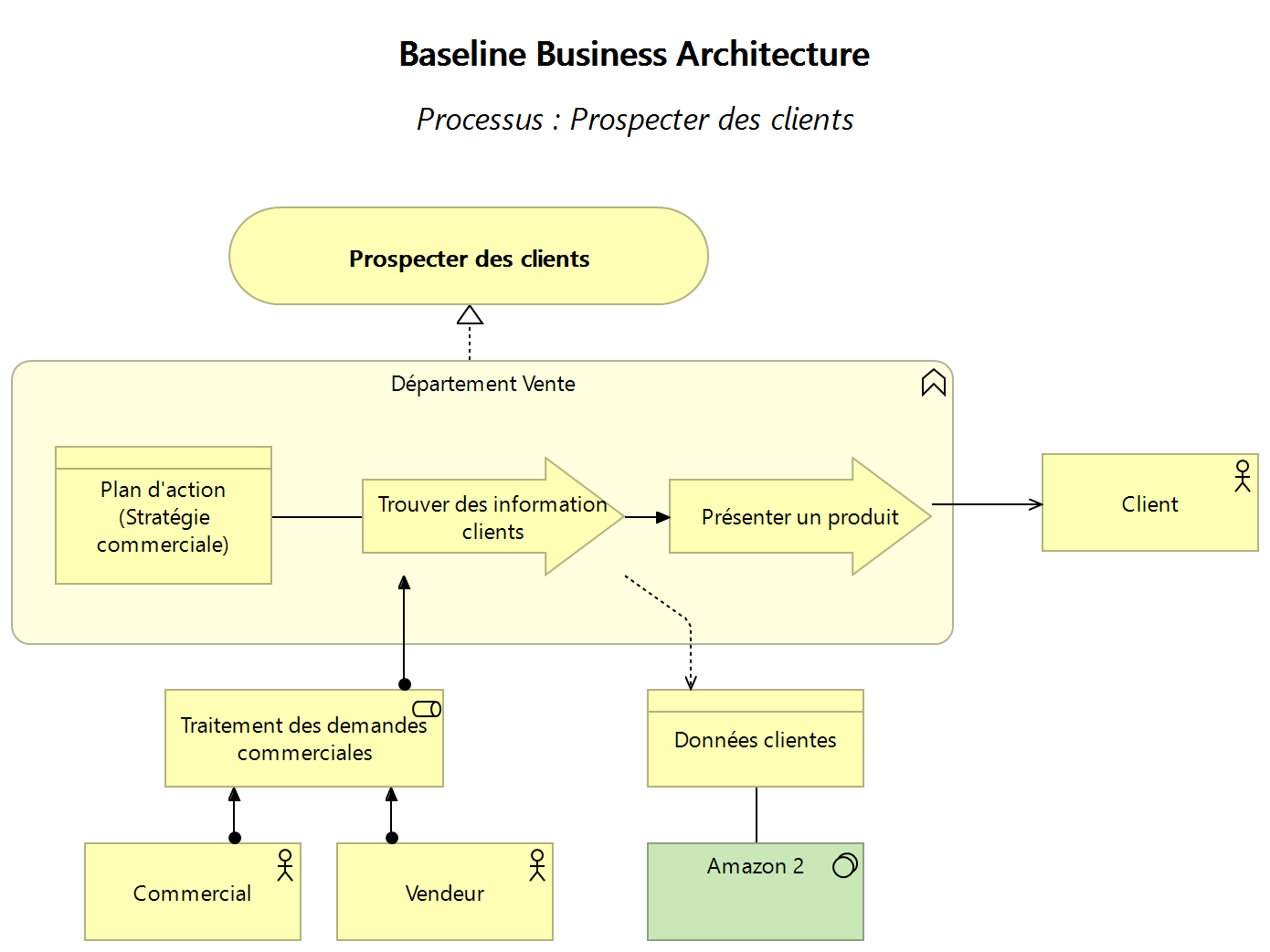


Figure 7 : Business Baseline Architecture - Prospecter des clients

La prospection des clients, dictée par le plan d’action de la stratégie commerciale vise à permettre à l’entreprise d’acquérir des nouveaux clients. Ce processus fait intervenir les commerciaux et les vendeurs au travers du traitement des demandes commerciales.

#### Processus : Vendre un produit d’assurance

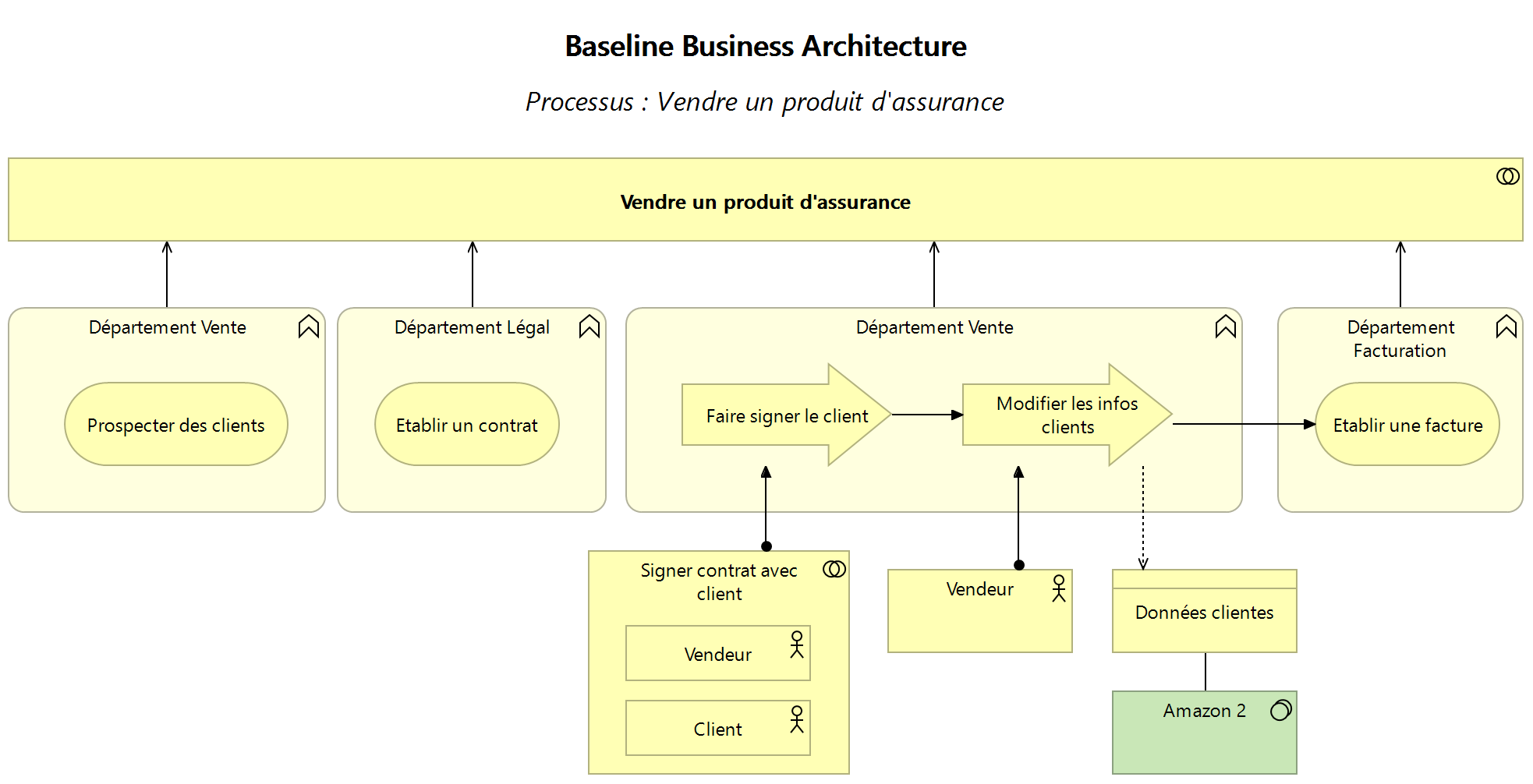


Figure 8 : Business Baseline Architecture - Vendre un produit d'assurance

La vente d’un produit d’assurance fait intervenir de nombreux acteurs, département et processus de l’entreprise collaborant pour fournir cette capacité.

Initié par le département vente et le processus de prospection commerciale, la vente d’un produit d’assurance nécessite l’établissement d’un contrat (département légal - processus décrit précédemment), la signature du contrat réalisés par les vendeurs, et l’établissement de la facture réalisée par le département facturation.

### Service Informatique

#### Processus : Sauvegarder quotidiennement les données

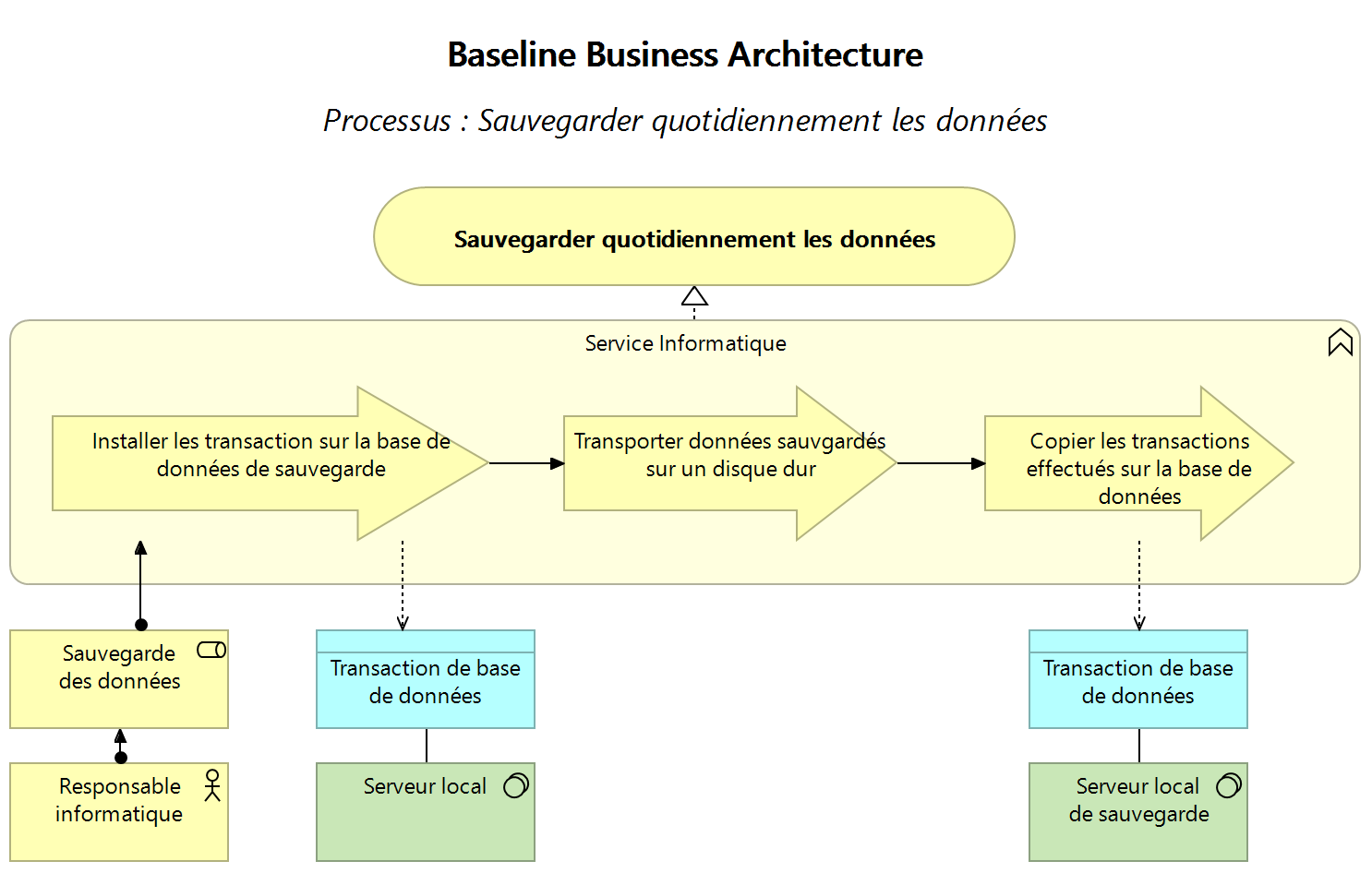


Figure 9 : Business Baseline Architecture - Sauvegarder quotidiennement les données

Le service informatique, composé du seul responsable informatique, est chargé du processus de sauvegarde quotidien des données du serveur local (serveur du département légal).

Ce processus nécessite une manipulation physique des données, dont les transactions journalières sont copiées sur un disque dur qui sera ensuite transporté jusqu’au serveur local de sauvegarde ou les transactions seront lancées.

### Évaluation des processus business

Le catalogue ci-après présente un résumé de l’évaluation des processus business. Notons que l’analyse de l’entreprise n’a pas relevé de problématique lié à l’organisation du travail ou à la réalisation des objectifs de service. Les processus présentent surtout des problématiques liés aux interactions interservices et aux interactions avec les données.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id.** | **Département** | **Processus business** | **Évaluation** | **Doit être révisé** |
| **P1** | Service client | Répondre à un client | Ce processus est complexe dû à la fragmentation des données dans plusieurs sources (MS Access, Amazon 1 & 2). La réplication dans plusieurs des données entraine des incohérences et alourdis les traitements.  Notons que le processus de transfert vers un autre service se fait à l’aide de la boite e-mail, ce qui entraine de nombreuses difficultés pour la collaboration et le partage d’information. | Oui |
| **P2** | Légal | Enquêter | D’un point de vue business ce processus ne présente aucun problème. Une révision de la source des données (serveur local) est cependant nécessaire. | Non |
| **P3** | Légal | Établir un contrat | D’un point de vue business, ce processus ne présente aucun problème. Une révision de la source des données (serveur local) est cependant nécessaire. | Non |
| **P4** | Facturation | Établir une facture | Ce processus ne présente aucun problème. | Non |
| **P5** | Vente | Définir une stratégie commerciale | Ce processus ne présente aucun problème. | Non |
| **P6** | Vente | Prospecter des clients | Ce processus ne présente aucun problème. | Non |
| **P7** | Vente | Vendre un produit d’assurance | Ce processus ne présente aucun problème. | Non |
| **P8** | Service Informatique | Sauvegarder quotidiennement | Ce processus doit être révisé entièrement. La sauvegarde manuelle chaque jour impose un RPO [[1]](#footnote-1)de 24h qu’il convient de diminuer. Par ailleurs, ce processus reposant entièrement sur la disponibilité d’un collaborateur, ses absences prévues ou imprévues peut entrainer des dégradations notables du RPO. | Oui |

Tableau 4 : Business Baseline Architecture - Évaluation des processus d'entreprise

## Architecture des données (data layer)

### Cartographie des données métiers et des accès

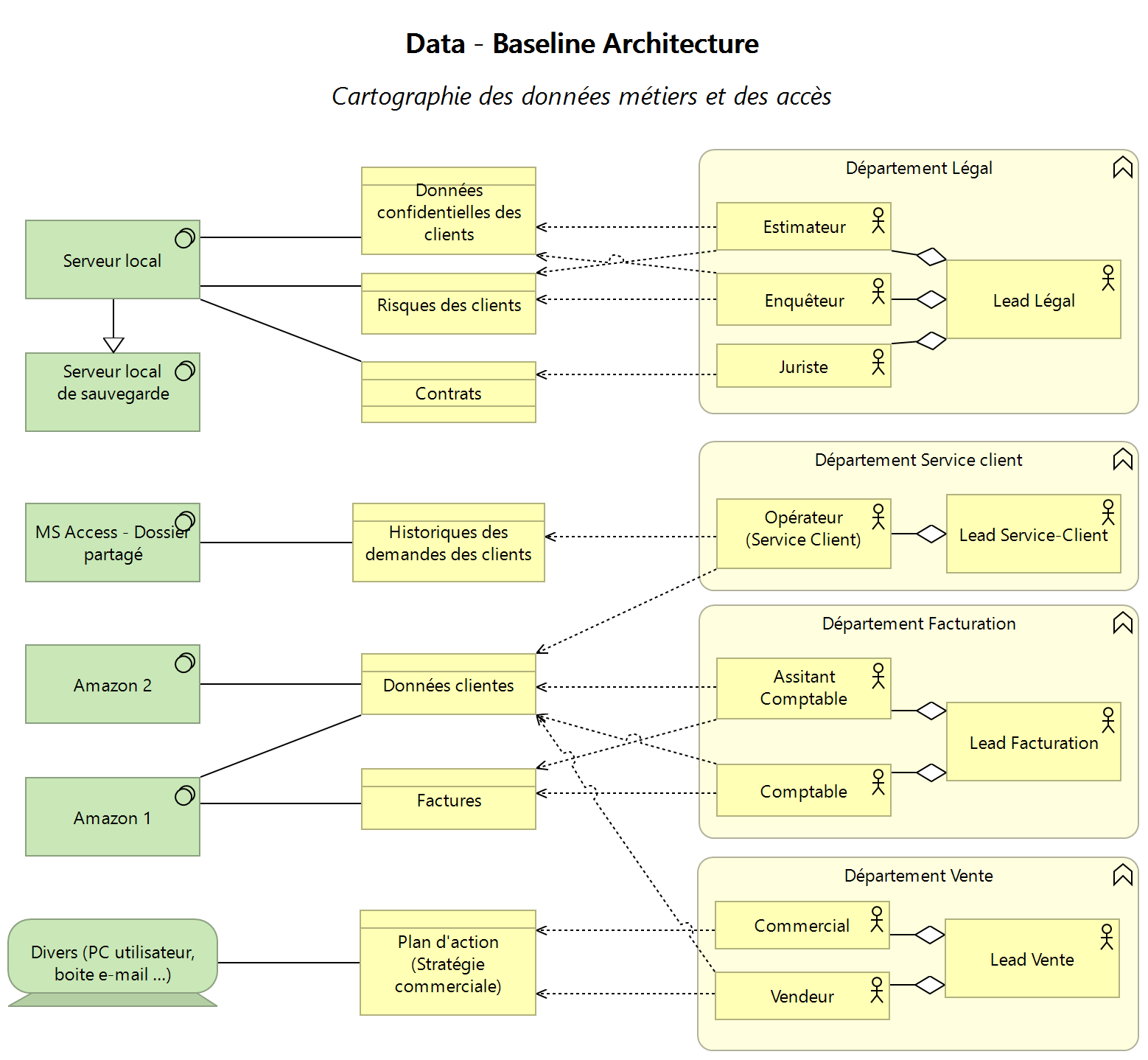


Figure 10 : Data Baseline Architecture - Cartographie des données

La cartographie des données métiers et de leurs accès permet de visualiser quels acteurs accède à qu’elles données. Les responsables de département (« leads ») sont supposés avoir accès aux informations de subordonnés. Notons que le rôle Responsable SI et CEO n’ont pas été représenté car supposés disposer d’un accès complet à l’ensemble des données.

Leurs supports de stockage est représenté à gauche du diagramme.

### Évaluation de l’architecture des données

#### Accès et mises à jour des données

Les données de l’entreprises sont actuellement réparties entre de nombreux support de stockage (MS Access, Amazon 1, Amazon 2, Serveur Local, et divers autres supports locaux) et pour certaines d’elles redondantes sur plusieurs stockages sans synchronisation apparente.

Cette situation, entraine à la fois une :

* Une forte fragmentation des données,
* Une absence de cohérence des données dans le cas de modification manuelles si les modifications ne sont pas appliquées à plusieurs reprises,
* Des complexités d’accès aux données à jours (latence pouvant aller jusqu’à 30minutes).

#### Sécurité des données

La sécurité des données n’est pas clairement établie dans cette architecture.

Un système de contrôle des droits d’accès, de modification ou de suppression sur les différentes données ne semblent pas implémentés[[2]](#footnote-2).

Aussi, il est possible de constater que la sécurité des *données confidentielles des clients* est assurée par le simple contrôle physique au serveur local et qu’aucune autre forme de sécurité (chiffrement, contrôle d’autorisation logiciel) n’a été identifié.

#### Résilience des données

La résilience des données n’est pas suffisante dans cette architecture.

Le processus de sauvegarde de la base de données MS Access n’est pas clairement défini. Par ailleurs la synchronisation de 30 minutes peut entrainer une perte de données importante (historique d’un appel).

Le RPO sur les données confidentielles est de 24h, délai bien trop long pour assurer une résilience efficace des données.

En dernier lieu, il est possible d’affirmer que l’absence de système centralisé conduit à l’utilisation d’outils non conventionnels (ex : boite mail, téléphone, notes) pour le partage des informations conduisant naturellement à la perte d’information et entrainant des délais importants dans la mise à jour des données.

## Architecture applicative (application layer)

### Cartographie des applicatifs de l’entreprise

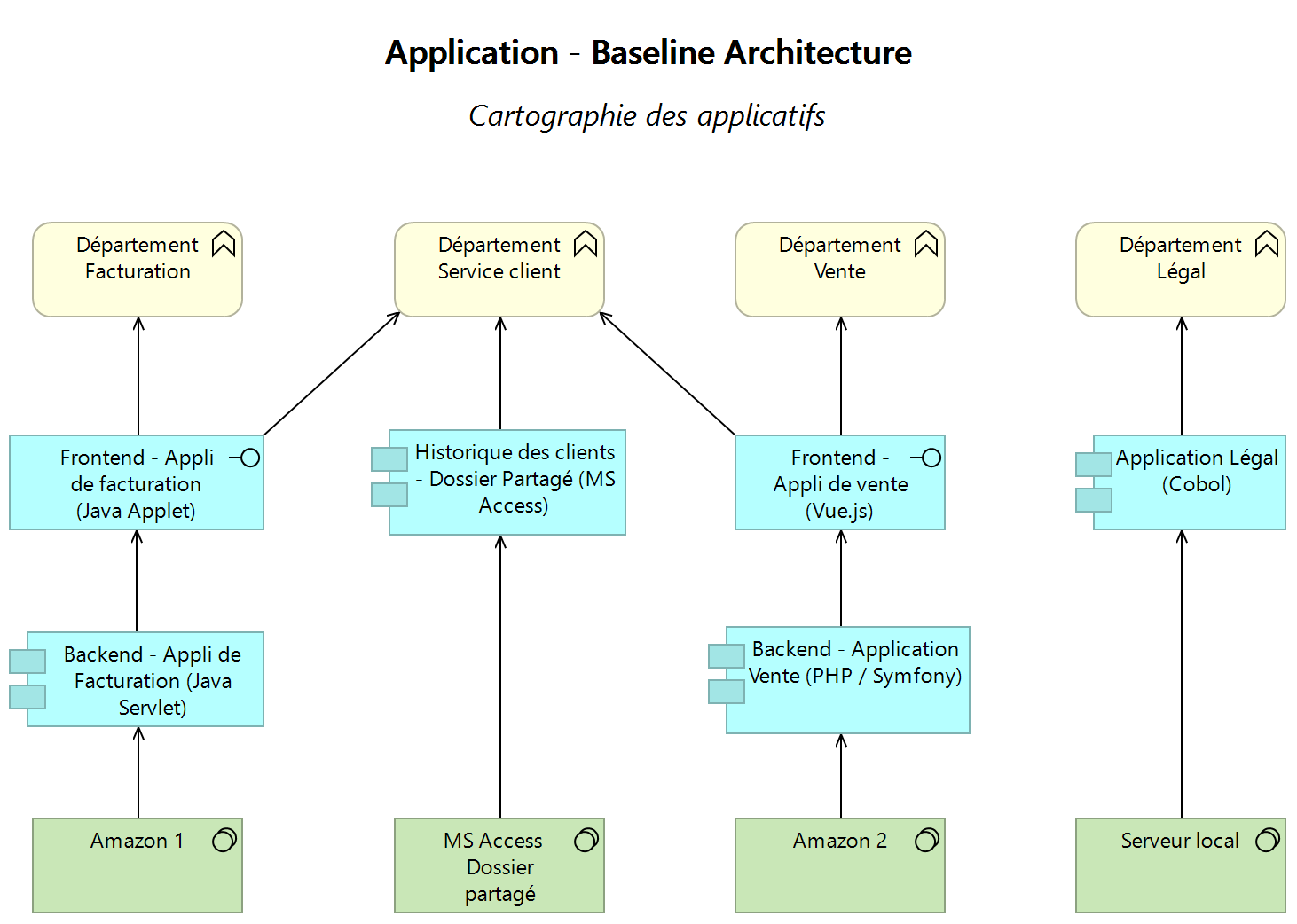


Figure 11 : Application Baseline Architecture - Cartographie des applicatifs

La cartographie des applicatifs de l’entreprise présente la liste des applications, les technologies employées, les infrastructures les supportant et les départements qui les utilisent.

L’audit technique réalisé en complément en de ce dossier d’architecture fourni des détails supplémentaire sur les capacités et le fonctionnement de chaque composant logiciel.

### Évaluation des applicatifs

Le catalogue ci-après présente l’évaluation des applicatifs de l’entreprise. Pour la plupart, ces applicatifs ne présentent pas de problématique de fond sur le service rendu, mais principalement sur la manière dont le service est rendu (trop lent, pas de collaboration possible …).

Les technologies étant pour la majorité d’entre elles obsolètes, les modifications et les évolutions sont complexes et couteuses.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id.** | **Applicatif** | **Type / Technologie** | **Évaluation** | **Doit être révisé** |
| **A1** | Frontend – Appli de vente | Client web / Vue.js 3 | Ces applicatifs sont construits sur des technologies modernes et ne font l’objet d’aucun incident technique ou de complexité de processus métier. | Non |
| **A2** | Backend – Appli de vente | Service web / PHP 7 (Symfony 4) | Non |
| **A3** | Frontend – Appli de facturation | Client web / Applet Java JSE7 | Ces applicatifs présentent des problèmes de compatibilité avec les navigateurs web moderne. Une évolution est nécessaire. | Oui |
| **A4** | Backend – Appli de facturation | Service web / Servlet Java JSE7 | Oui |
| **A5** | Application Légal | Client lourd / Cobol | Cet applicatif utilise des technologies obsolètes, rendant les évolutions et la maintenance complexe et couteuse. Une évolution est nécessaire. | Oui |
| **A6** | Historique des clients | Dossier partagé MS Access | Cet applicatif présente différents problèmes liés aux temps de synchronisation des données. Nous noterons par ailleurs que l’intégration de ces données dans les autres applicatifs semble complexe. | Oui |

Tableau 5 : Application Baseline Architecture - Évaluation des applicatifs de l'entreprise

## Architecture technique (technical layer)

### Cartographie de la plateforme technique

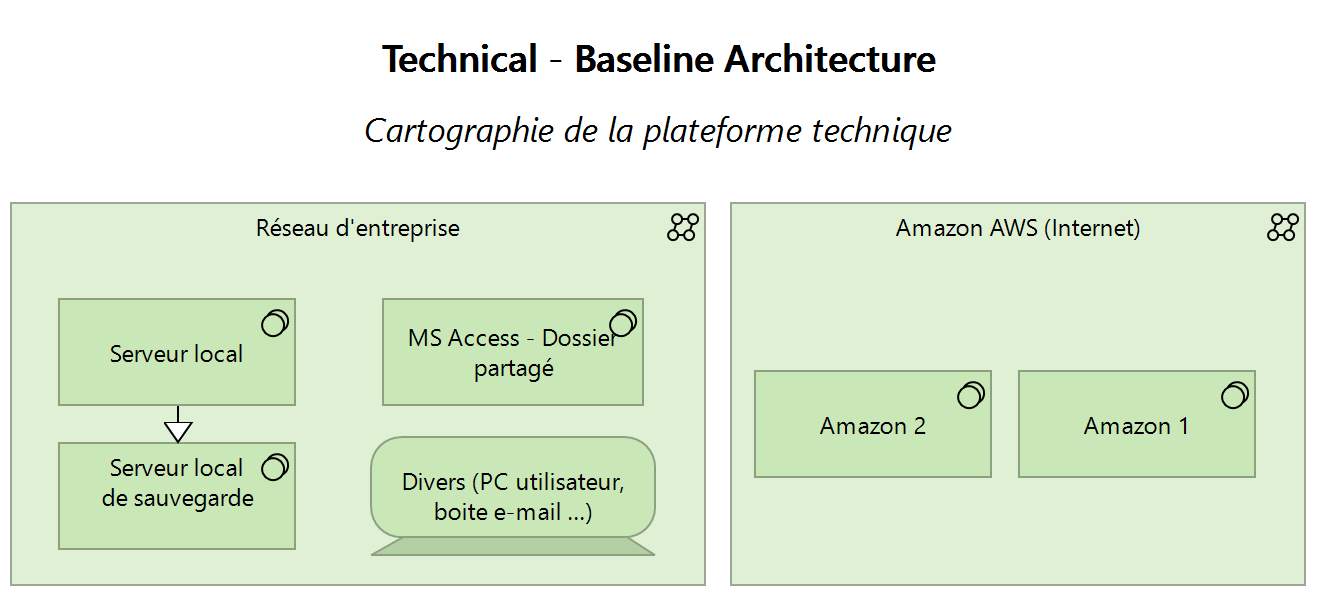


Figure 12 : Technical Baseline Architecture - Cartographie de la plateforme technique

La plateforme technique de l’entreprise est relativement simple et est composé de plusieurs serveurs hébergé sur la plateforme AWS (Amazon 1&2) et de machines locales (Serveur Local / Dossier MS Access partagé). Les sections précédentes du présent document et l’audit technique fournissent des détails supplémentaires sur les rôles et les capacités de chaque machine.

### Évaluation de la plateforme technique

Le catalogue ci-après présente l’évaluation de la plateforme technique. La plateforme technique utilise des composants cloud pour stocker une partie des données. Notons que les processus de sauvegardes des données entreposés sur Amazon 1 & 2 ne sont pas connus.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id.** | **Composant technique** | **Type** | **Évaluation** | **Doit être révisé** |
| **T1** | Serveur local | Machine physique | Ce composant présente des problématiques lié à sa robustesse et aux modalités de sauvegarde. Une évolution est nécessaire. | Oui |
| **T2** | MS Access – Dossier partagé | Répertoire réseau | Ce composant présente des difficultés de synchronisation lié à son mode d’exploitation. Une évolution est nécessaire. | Oui |
| **T3** | Amazon 1 & 2 | Machine virtuelle | Ces machines virtuelles ne posent aucun problème connu. Une précision des processus de sauvegarde et de la répartition des données est toutefois nécessaire. | Non |

Tableau 6 : Technical Baseline Architecture - Évaluation de la plateforme technique

# APPROCHE ARCHITECTURALE

## Synthèse de l’évaluation de l’architecture de référence

### Points fort de l’architecture de référence

Les évaluations de l’architecture de référence permettent d’identifier plusieurs points fort qu’il convient d’exploiter :

* **Les processus business fonctionnent correctement**. Les modifications de processus ne porteront que sur le service client, afin d’intégrer un nouvel outil de ticketing pour simplifier la collaboration, et sur les processus associé au responsable SI qui vont naturellement devoir évoluer avec la nouvelle architecture.
* **Une partie de l’infrastructure exploite déjà les capacités de la plateforme cloud AWS**. Il convient de conserver et de renforcer ces usages.
* **Le logiciel de vente est une web app utilisant un bloc de technologies moderne**. Il est possible de s’appuyer sur cet outil pour s’en servir comme base de code pour développer les nouvelles interfaces et nouveaux services associés.

### Points faibles de l’architecture de référence

Les points faibles identifiés de l’architecture de référence devront être corrigé dans la conception de la nouvelle architecture. La liste ci-dessous reprend les principaux points faibles identifiés :

* **Duplication et mise à jour complexe des données**. Les données sont dupliquées et ne disposent pas de processus de réplication automatique rendant les mises à jour complexes et diminuant la cohérence du référentiel de données.
* **Contrôle d’accès insuffisant aux informations.** L’accès aux ressources et aux données devraient faire l’objet d’un contrôle d’accès renforcé basé sur des rôles et profils utilisateurs.
* **Hétérogénéité et obsolescences des technologies utilisées**. Les technologies obsolètes doivent être supprimées et l’homogénéité des systèmes et des technologies recherchée.

## Approche architecturale retenue

### Prototype de l’architecture cible

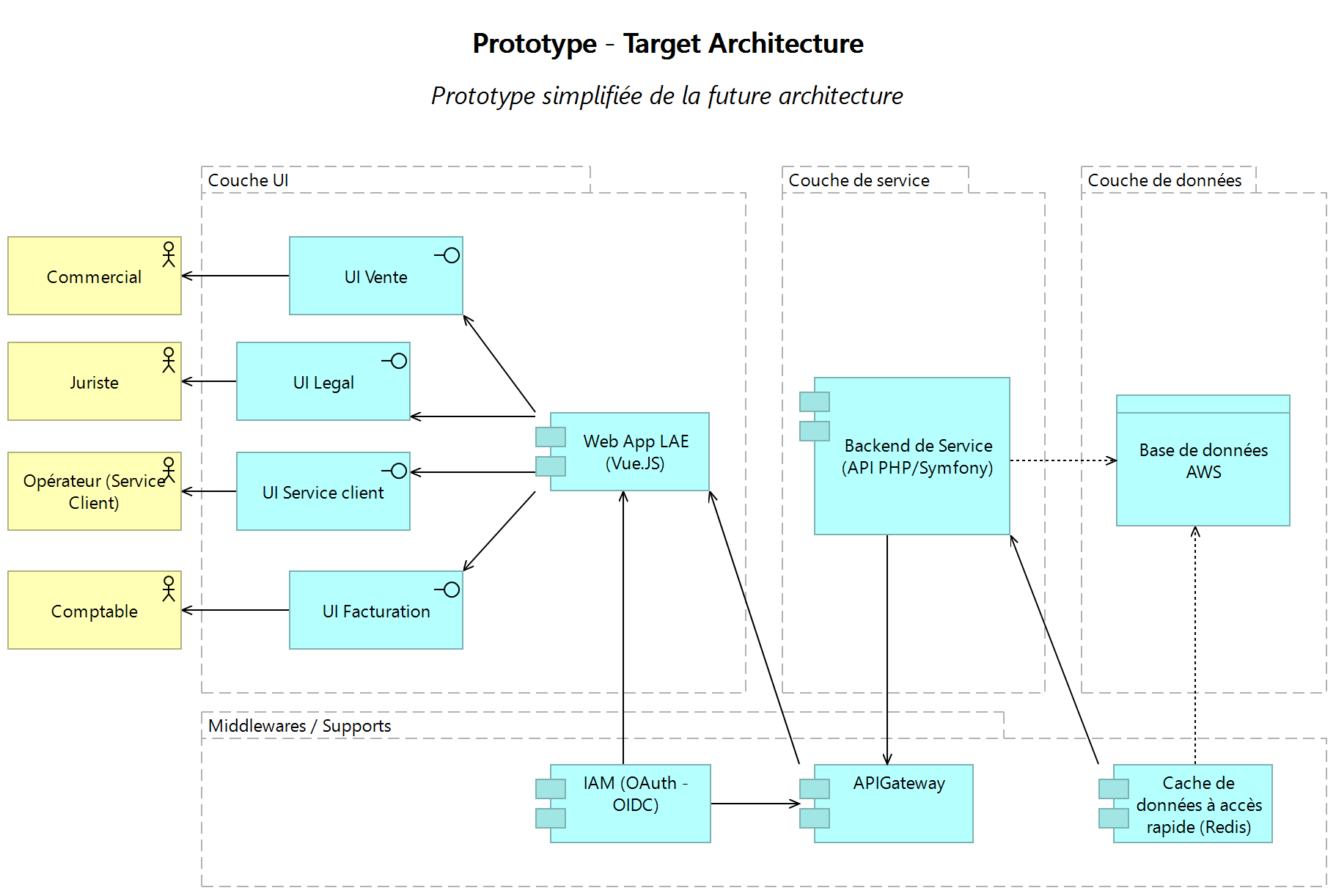


Figure 13 : Prototype simplifiée de la nouvelle architecture

En s’appuyant sur les conclusions de l’évaluation et en temps compte de la structure de l’entreprise (capacités, ressources, objectifs) et des contraintes associé à ce chantier d’architecture, le prototype présenté ci-avant (Figure 13) présente une vue simplifiée des principales caractéristiques de l’architecture applicative cible.

Cette approche, qui sera précisée dans les modélisations de l’architecture cible prévoit un découpage en 3 couches principales + des couches middleware / support.

Les couches présentes sont :

* **La couche UI** (*Presentation layer*) ; qui fournit les interfaces logicielles (sous forme de webapp Vue.js dans le présent cas).
* **La couche de service** (*Service layer*) ; qui assure les traitements métiers (business process). Cette couche doit être uniquement accessible au travers d’API Web REST – basées sur le protocole HTTPS.
* **La couche de données** (*Data layer*) ; qui contient les données exploitées par la couche de service. Notons que les interfaces n’ont jamais directement accès aux données sans passer par la couche de service.
* **La couche middleware / support** (*Middleware layer*) ; qui contient l’ensemble des composants transverses (authentification, autorisation, cache de données à accès rapide, API Gateway …).

Le design pattern d’architecture logiciel utilisé sera basé sur une approche SOA avec une forte modularité.

### Justification de l’approche architecturale

#### Choix des technologies (PHP/Symfony – Vue.JS)

Les technologies employés sont basées sur le couple PHP/Symfony + Vue.JS. Ce choix est justifié par la présence de ces technologies déjà existantes dans le SI qui serviront de base de code pour les nouveaux outils.  
Le Framework Symfony (open-source) embarque par ailleurs un écosystème complet (notamment un moteur de traitement asynchrone) et est activement maintenu par une importante communauté.

La libraire Vue.JS est utilisée pour développer des interfaces réactives facilement et dispose-t-elle aussi d’un excellent écosystème (Vuex, VueCLI …) facilitant les développements.

#### Logiciels accessibles sous forme de Web-App (SaaS)

Les interfaces sont accessibles au travers de web-app (construite avec Vue.JS) afin de simplifier les opérations de maintenance et de mise à jour des logiciels, faciliter le déploiement sur les postes opérateur (un simple navigateur web est nécessaire) et réduire les coûts de développement.

Les utilisateurs sont déjà habitués à utiliser ne nombreux logiciels SaaS.

#### Sécurité des flux avec OAuth / OIDC

Le Framework d’autorisation OAuth est largement implémenté dans la plupart des outils (notamment les API Gateway et les IAM) et fourni une base solide pour l’implémentation de la sécurité par autorisation dans les API et les logiciels.

L’extension OIDC (OpenID Connect), notamment lorsque le serveur d’autorisation prend en charge le SSO fourni une méthode simple et standardisé pour l’implémentation de l’authentification dans les applications web.

#### Fédération des API autours d’une API Gateway

Bien que ce type d’architecture ne l’impose pas, l’utilisation d’une API Gateway fait partie des bonnes pratiques dès lors que des API sont utilisées.

L’API Gateway fourni un ensemble de service (contrôle d’authentification et d’autorisation basé sur OAuth, load balancing, gestion des versions, url virtuelles) permettant de décharger la couche service de ces actions. L’évolutivité et la modulation sont par ailleurs simplifiés ce qui en fait un composant à haute valeur ajoutée.

## Outil de ticketing externe

Le cahier des charges impose l’utilisation d’un outil de ticketing externe. Cet outil de ticketing vise à rendre deux services principaux :

* Centraliser les demandes e-mails des clients avec création de ticket automatique.
* Faciliter la collaboration interservices via l’utilisation de ticket dans un outil centralisé.

L’outil choisi devra obligatoirement intégrer ces deux fonctionnalités. Le choix sera par ailleurs justifié en fonction du coût de la solution, de sa réputation, de son ergonomie et de sa capacité à s’intégrer à d’autres outils.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fonctionnalité / Produit** | **GROOVEHQ** | **ZENDESK** | **FRESHDESK** |
| Collaboration en interne | Oui | Oui | Oui |
| Support client (Création automatique de ticket lors de la réception d’un e-mail) | Oui | Oui | Oui |
| Ergonomie |  |  |  |
| Disponibilité en Français | Non | Oui | Oui |
| Intégration aux outils tiers | Oui | Oui | Oui |
| SSO (avec OIDC - JWT externe) | Oui | Oui | Oui |
| Coût (pour 20 licences utilisateurs) | ~620 EUR / Mois | ~980 EUR / Mois | ~980 EUR / Mois |
| RGPD Compliant | ? | Oui | Oui |

Tableau 7 : Table de comparaison des outils de ticketing du marché

## Exigences de l’architecture cible

Le catalogue ci-après fourni la liste des exigences d’architecture (haut niveau) pour la nouvelle architecture. Ce catalogue a été créé en s’appuyant sur les évaluations de l’architecture de référence et les bonnes pratiques générales d’architecture.

|  |  |  |  |
| --- | --- | --- | --- |
| **Catégorie** | **Id.** | **Type** | **Spécification** |
| **Architecture** | **AR1** | Gouvernance | Une gouvernance de l’architecture doit être défini au sein de l’entreprise. Les missions principales de la gouvernance porteront sur la conception et le contrôle de conformité de l’architecture SI et business. |
| **AR2** | Design Pattern | Une architecture basée sur une couche de service modulable est préconisée. L’architecture doit répondre aux objectifs de disponibilité, de vélocité, de sécurité et de maintenance simplifiée recherchés. |
| **AR3** | Stack techno. | La stack technologique PHP – Symfony – Vue.JS est préconisé pour le développement des applicatifs. |
| **AR4** | Mutli-environnement | L’ensemble du système devra exister à minima sur deux environnements : développement et production. |
| **AR5** | Building block | L’architecture devra être bâti autant que possible sur des briques de références faiblement couplées et à forte cohérence. |
| **AR6** | Technologies | Les logiciels et technologies open-source et disposant d’un écosystème complet sont préconisés. |
| **Sécurité** | **AR7** | Contrôle d’accès | Un contrôle d’accès (authentification) basé sur des standards de sécurité devra être utilisé. L’utilisation de la technologie OAuth2 / OIDC est préconisé pour l’authentification. |
| **AR8** | 2FA | Le système exige une authentification à deux facteurs (validation par délivrance de token) pour tout accès nécessitant une sécurité par login / mot de passe. |
| **AR9** | Transfert des données | Le transfert des données tant à l’extérieur qu’à l’intérieur du SI n’utilisera que des protocole sécurisés basé sur la technologie TLS (HTTPS, AMQPS, SSH tunneling ...) |
| **AR10** | Chiffrement | Les données sensibles doivent être stockées de manière non lisible sur le support de stockage (base de données, serveur de fichier ...). Le protocole AES-256 est préconisé. |
| **AR11** | Single-Sign On | Les technologie de type SSO doivent être mise en œuvre afin de renforcer la protection et améliorer le confort utilisateur. |
| **AR12** | Mise à jour des systèmes | Les systèmes doivent être maintenus à jour. Des procédures de contrôle et de mise à jour à intervalle régulier doivent être implémentés. |

Tableau 8 : Catalogue des exigences de l'architecture cible (1/2)

|  |  |  |  |
| --- | --- | --- | --- |
| **Catégorie** | **Id.** | **Type** | **Spécification** |
| **Disponibilité** | **AR13** | Plateforme cloud | L’utilisation d’une plateforme cloud de type PaaS (Platform as a Service). L’écosystème AWS (en partie déjà implémenté) permet de répondre aux besoins définis. |
| **AR14** | Redondance | L’ensemble des services et applicatifs doit être redondé au moins 2 fois. Des mécanismes de bascule simplifiés doivent être implémentés. |
| **AR15** | Sauvegarde des système et des données | Les systèmes et les données doivent être sauvegardée par des processus automatiques. Des procédures de reprises doivent être implémentée et testées régulièrement. |
| **AR16** | Non duplication des données | Les données ne doivent pas être dupliquée. La centralisation des donnés avec contrôle d’accès doit être recherchée. |
| **AR17** | RTO / RPO | Le RPO après incident doit être < 10min.  Le RTO après incident doit être < 1h. |
| **Qualité de service / Performance** | **AR18** | Cache applicatif | L’utilisation de technologie de cache applicatif sont préconisés (Local ou type Redis). Ces caches applicatifs ont pour triple objectif de : Réduire les temps d’accès aux données, réduire les opérations sur le SI, améliorer les performances des applications et services. |
| **AR19** | Réactivité des interfaces | Toutes les interfaces doivent répondre avec des temps de latences inférieures à 3 secondes dans des conditions normales d’exploitation. |
| **AR20** | Traitement différés | Les traitements lourds doivent être systématiquement traités en asynchrones (via des queues FIFO par exemple) et implémentés des mécanismes de tolérances aux erreurs (re-déclenchement automatique par exemple) |

Tableau 9 : Catalogue des exigences de l'architecture cible (2/2)

# ARCHITECTURE CIBLE

L’architecture cible de l’entreprise correspond à l’implémentation des nouvelles exigences et recommandations permettant d’atteindre les objectifs définis par l’entreprise. Cette section présente et décrit les modifications d’architecture à apporter selon les différents points de vue : Business, data, applicatif et technique.

*Pour des raisons de simplifications, les artefacts dont l’objet principal n’a pas été modifiés ne seront pas représentés dans cette section. Le référentiel d’architecture permet de naviguer plus facilement à travers les différentes modélisations.*

## Architecture métier (business layer)

Peu de processus métiers seront modifiés. Comme précisé dans l’évaluation de l’architecture business, seul les processus du service client et le processus du service informatique seront amené à évoluer. Notons toutefois que les outils et la plateforme technologique utilisés seront amenées à évoluer.

### Département service client

#### Processus : Répondre à un client

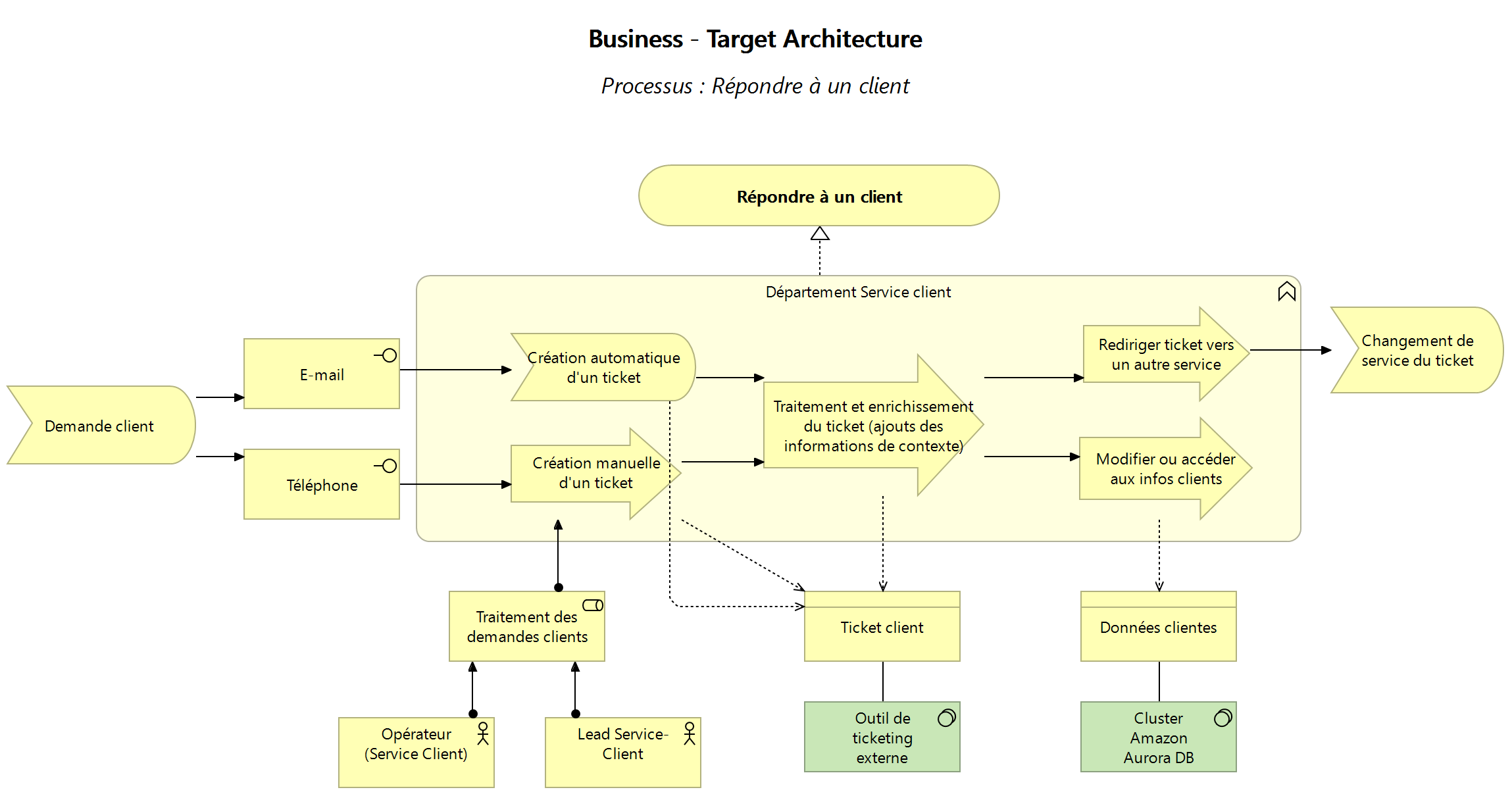


Figure 14 : Business Target Architecture - Répondre à un client

Le processus « Répondre à un client » va évoluer pour s’articuler autour du nouvel outil de ticketing.

L’outil de ticketing remplace l’ancien système d’historique client par dossier partagé au sein d’une base MS Access pour fournir une interface ergonomique permettant aux différents intervenants d’enrichir et d’affecter les tickets aux services concernés. Les tickets peuvent être créées par les opérateurs du service client, suite à un appel, ou généré automatiquement à la réception d’un e-mail d’un client.

Le système de ticketing permet désormais une collaboration en temps réel, centralise les demandes des clients au sein d’un même outil et facilite la collaboration et les processus inter-service.

### Service Informatique

#### Processus : Contrôler trimestriellement la sauvegarde des données et tester la restauration

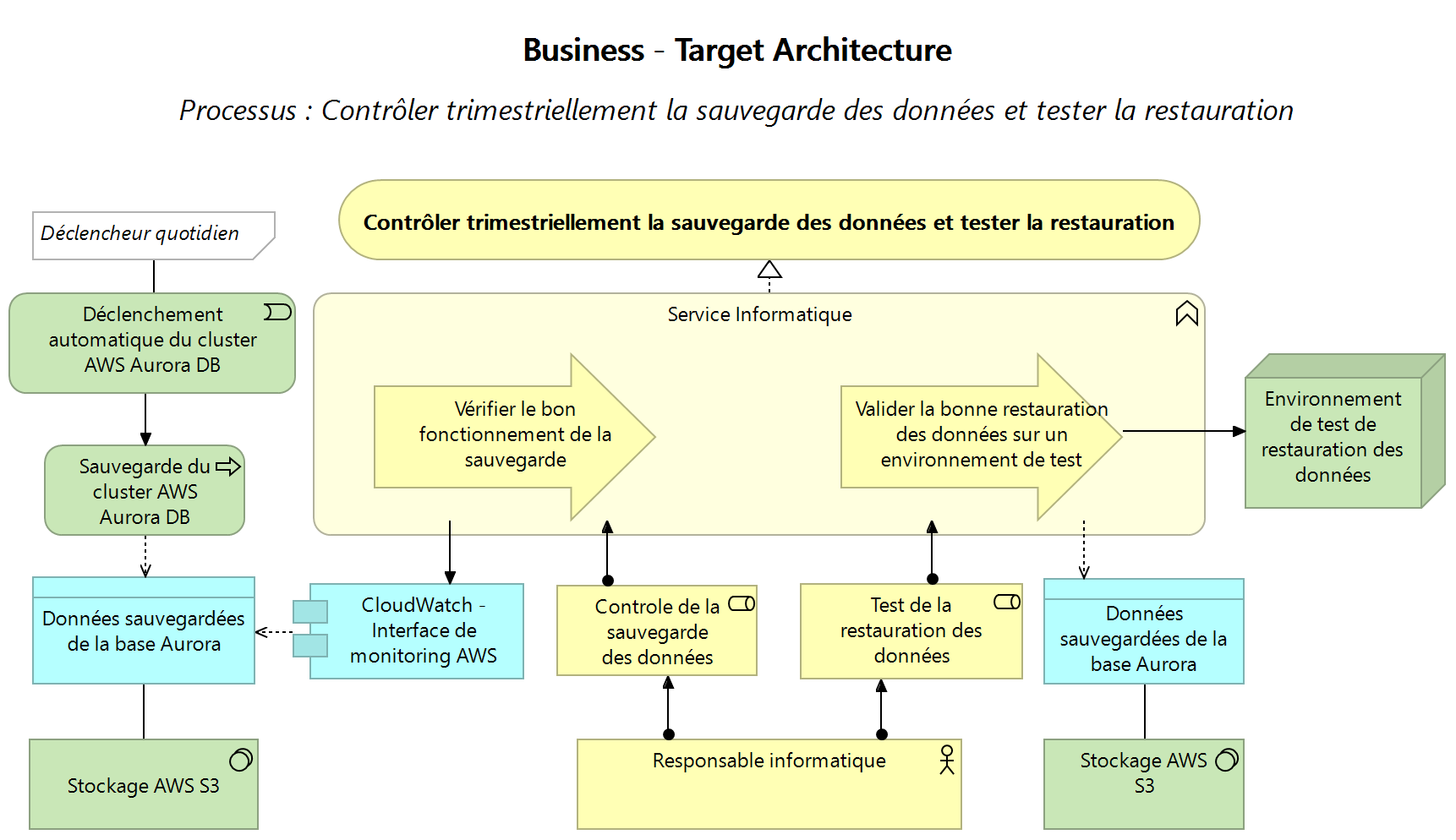


Figure 15 : Business Target Architecture - Controller la sauvegarde et la restauration des données

Le processus « Sauvegarder quotidiennement les données » et supprimé au profit d’un tout nouveau processus attribué au responsable informatique : Vérifier le bon fonctionnement des sauvegarde automatique et tester la restauration des données.

Ce processus vise à automatiser les actions de sauvegarde sur la plateforme AWS et à remplacer les tâches fastidieuse de sauvegarde par des contrôles et des tests à haute valeur ajoutée afin de s’assurer une parfaite exécution du PCA en cas d’incident majeur.

## Architecture des données (data layer)

### Cartographie des données métiers dans l’entreprise

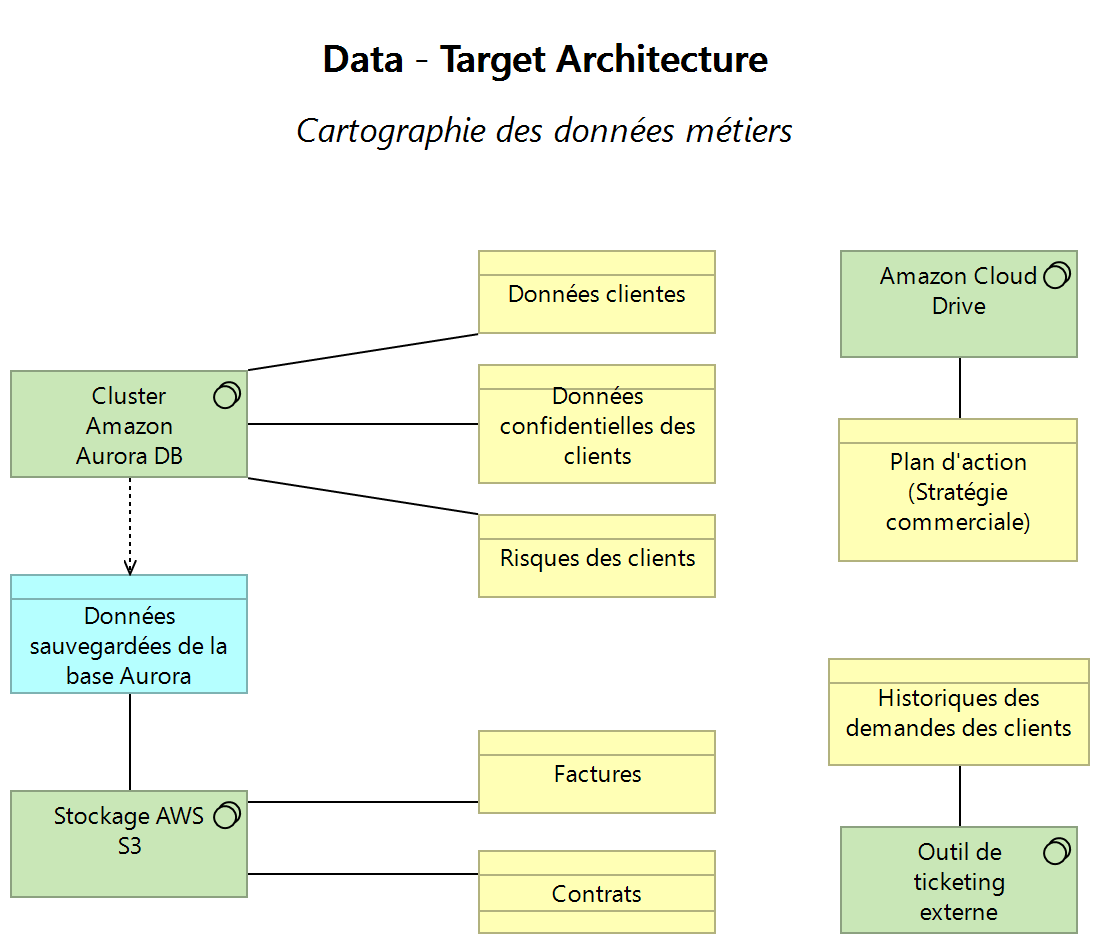


Figure 16 : Data Target Architecture - Cartographie des données métiers

Il est possible de visualiser sur la figure ci-avant les différentes données métiers de la nouvelle architecture. Le périmètre des données n’a pas évolué, toutefois les supports de stockage ont été modifiés. Il possible de constater que la plupart des données métiers, anciennement fragmentées dans de nombreuses bases, sont regroupés au sein d’une même base de données déployée sur un cluster Amazon Aurora[[3]](#footnote-3).

Les données de la base Aurora sont automatiquement sauvegardées sur une instance Amazon S3. Les données statique du type « Plan d’action » et autres données seront stockés sur un répertoire partagé Amazon Cloud Drive. L’outil de ticketing dispose bien entendu de sa propre base stockée chez l’éditeur.

### Sécurité des données et contrôle d’accès

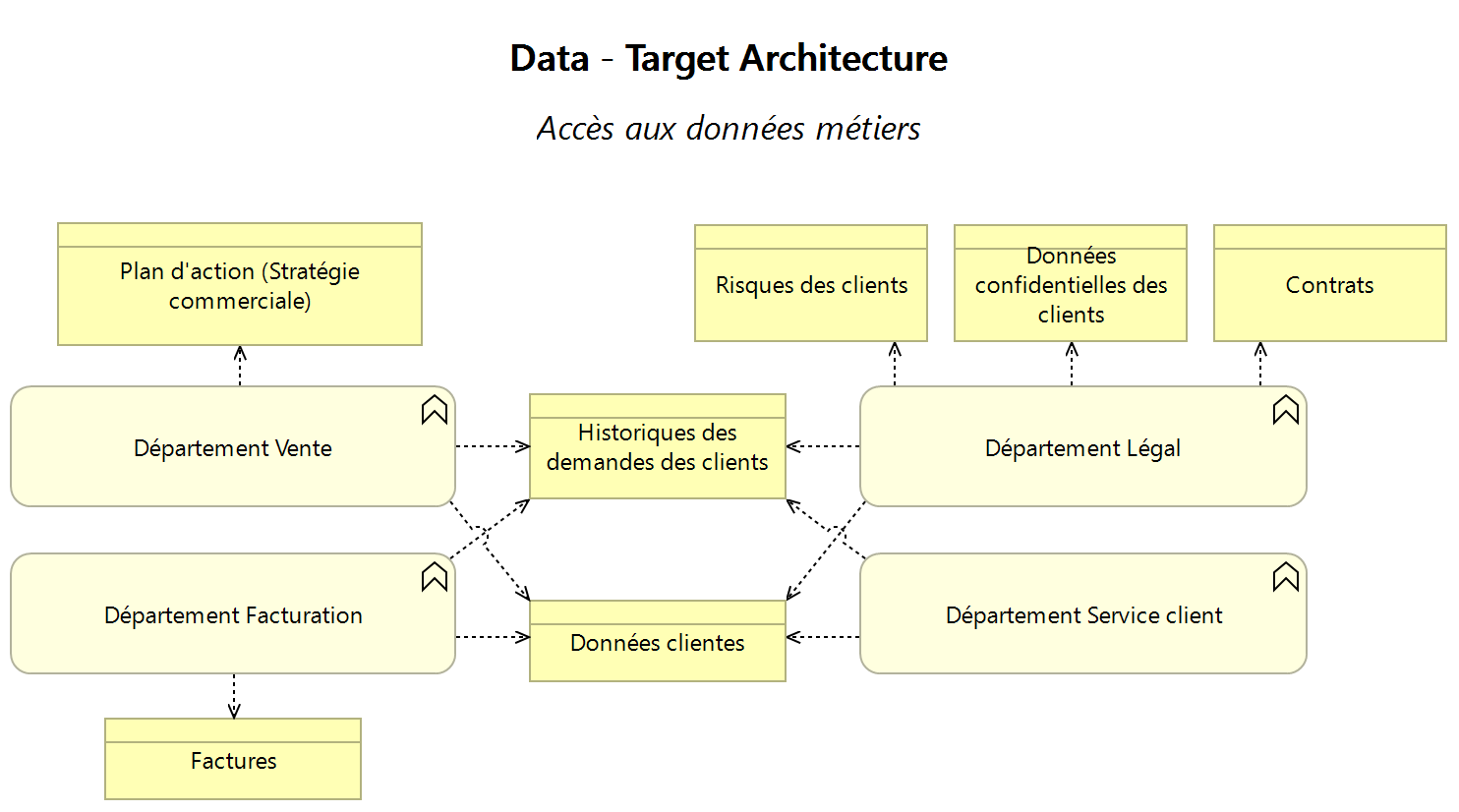


Figure 17 : Data Target Architecture - Accès aux données métiers

Le regroupement des données au sein d’une même base ne signifie pas un accès libre aux données de l’entreprise. L’accès aux différentes informations est réalisé à l’aide des interface de la nouvelle web app LAE. Le logiciel embarque un contrôle d’accès strict basé sur un système de rôle qui assure la sécurité des données, tant sur le périmètre, que sur les capacités (lecture / écriture).

Nous noterons toutefois que les données « Historique des demandes clients » est désormais accessible à tous, et ce avec objectif de simplifier la collaboration inter-service.

Un soin tout particulier sera apporté aux données sensibles ou critiques (notamment les coordonnées bancaires ou les informations de santé) en ajoutant un cryptage spécifique (voir exigences d’architecture) afin de se prémunir des risques en cas de fuite de données.

Notons par ailleurs que les exigences d’architecture imposent une authentification à deux facteurs (2FA) pour accéder au système d’information. Le vol d’information d’identification d’un collaborateur n’aura pas pour conséquence d’entrainer une fuite de données.

## Architecture applicative (application layer)

### Cartographie des applicatifs

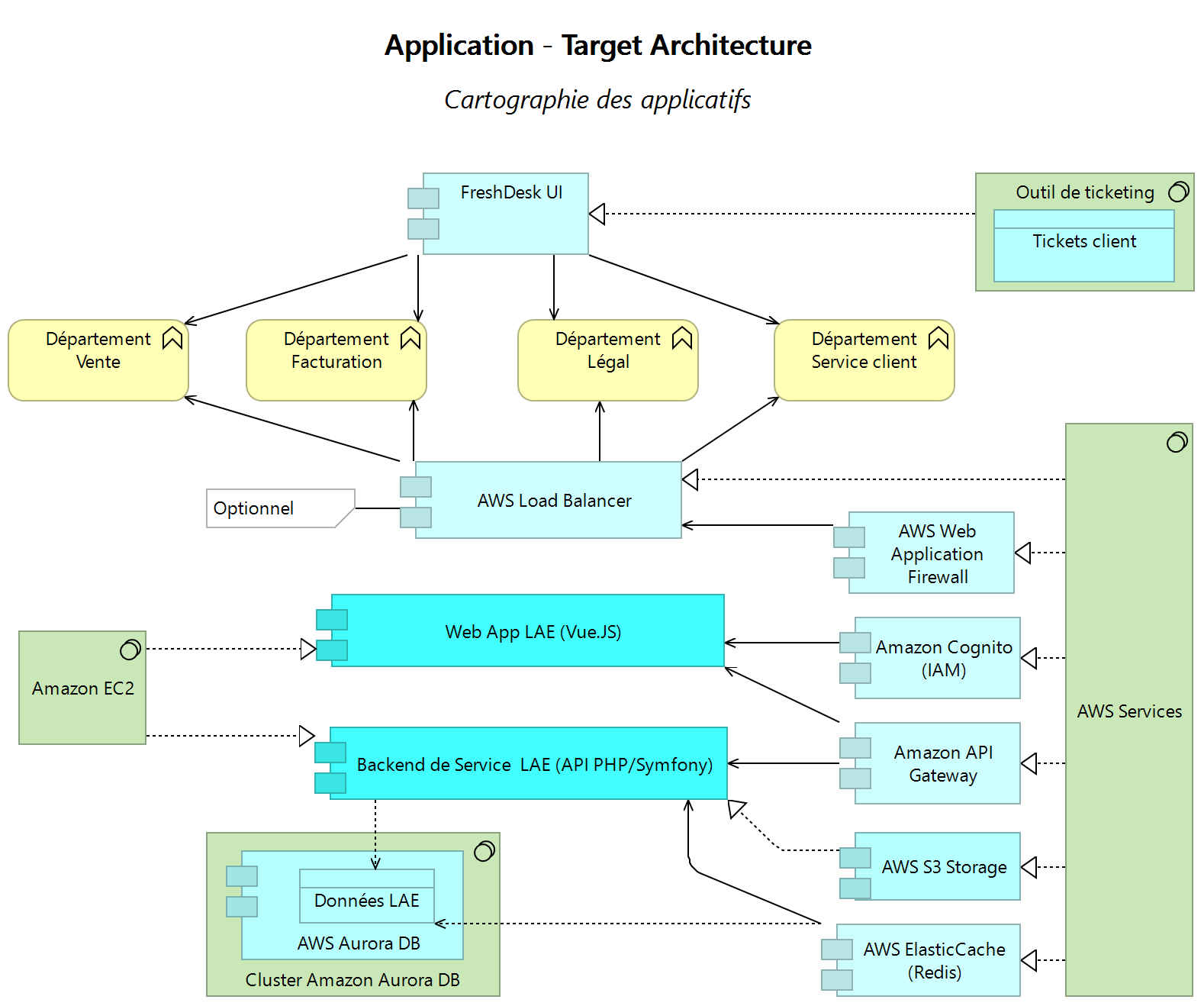


Figure 18 : Application Target Architecture - Cartographie des applicatifs

La figure ci-avant présente la vue de la nouvelle architecture applicative de l’entreprise. Le modèle affiché est directement issu du prototype présenté dans le présent document. Les figures en « bleu clair » représente des services « clef en main » sur lesquelles aucun développement interne n’est nécessaire. Les figures « bleu foncé » sont les applications sur lesquelles des développements logiciels sont à prévoir.

Nous noterons que l’ensemble des interfaces utilisateur sont regroupées dans la nouvelle Web App LAE (Les fonctionnalités différent en fonction des capacités de l’utilisateur), développée en Vue.JS. L’application de vente servira de base à la création WebApp.

La web App exploite, au travers de l’API Gateway, les services backend, exposés sous la forme d’API RESTFULL et développé en PHP/Symfony. La couche service exploite les données du cluster Aurora DB regroupant la plupart des données métiers de l’entreprise. Pour rappel, le backend de l’application de vente servira de base à la création de ce composant. Une montée en version du langage utilisé (PHP 7 vers PHP 8) est à prévoir au cours des développements.

La base de données principale et gérée par deux instances (répliquées à l’identique et auto-scalable) d’Amazon Aurora DB.

L’authentification, l’identification et les contrôles d’accès aux services et données sont assurés par le service Amazon Cognito agissant comme IAM et fournissant les informations d’identification et d’authentification en exploitant le Framework d’autorisation OAuth 2.0 et son extension OIDC.

La présence du service AWS S3 est rendu nécessaire pour assurer le stockage des données volumineuses (factures, contrats, justificatifs divers, CGV …).

Le service AWS ElasticCache vise à fournir au backend un cache de données à accès rapide visant à améliorer les performances globales du SI en fournissant une solution simple et centralisée.

Le Web Application Firewall est utilisé principalement pour renforcer la sécurité de la web application et fournir un premier niveau de détection des cyberattaques.

Nous noterons également la présence d’un load balancer optionnel. Le load balancer est utilisé pour réceptionner les requêtes et les transférer aux diverses instances de l’application web. Bien qu’il ne soit pas indispensable au bon fonctionnement de l’application et que la charge ne l’impose pas, la mise à disposition de plusieurs instance de la web-app (tout comme celle du back-end) permet de s’assurer d’une disponibilité de la plateforme en cas d’échec d’une instance de service et au cours des processus de déploiements de mise à jour. Il permet par ailleurs de faciliter les futures évolutions de cette plateforme (telle qu’une ouverture aux clients par exemple). Son utilisation paraît totalement justifiée eu égard au coût relativement faible d’un tel service.

En dernier lieu, nous pouvons visualiser la présence du nouvel outil de ticketing, accessible aux différents services opérationnels de l’entreprise, et exploitant bien entendu ses propres données. Bien que non mentionné dans ce schéma et hors du scope de ce chantier, l’outil de ticketing sélectionné fourni des capacités d’intégration permettant d’envisager une évolution pour l’intégrer directement dans la web app.

La plateforme technique (représentée en « vert » dans le schéma) sera détaillée dans la section correspondante.

### Catalogue des composants applicatifs

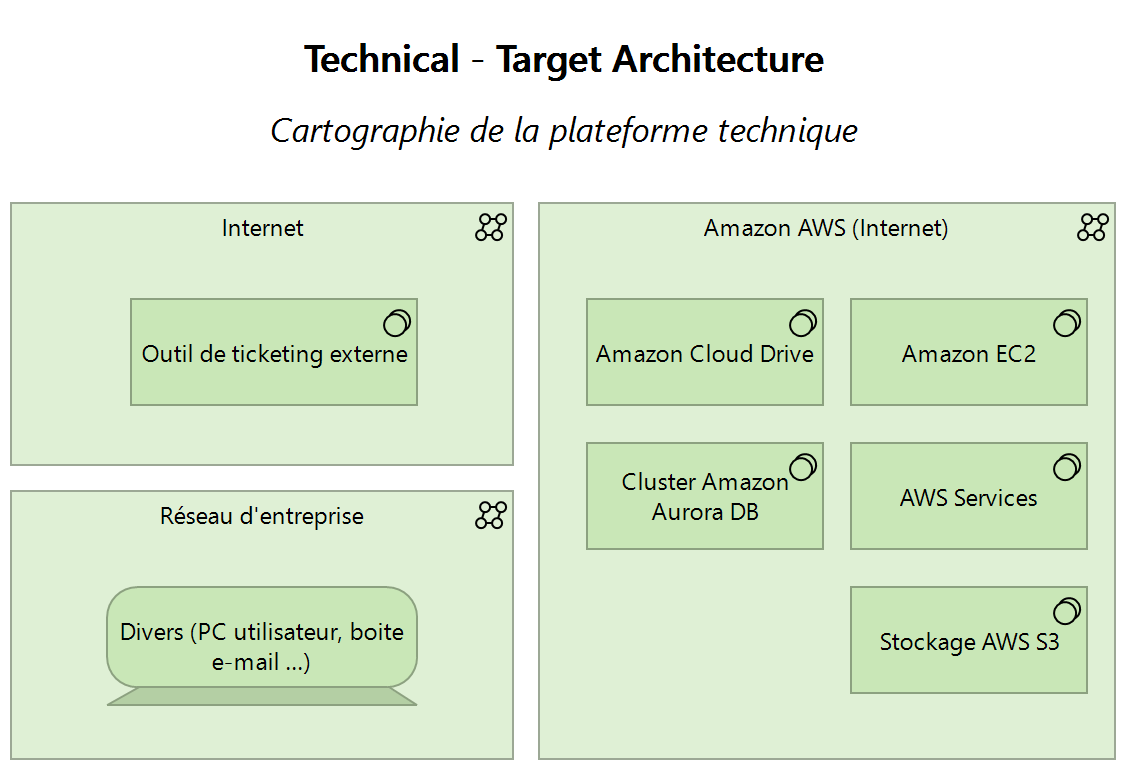
Le catalogue ci-après fourni une vue d’ensemble des applicatifs de la nouvelle architecture et des technologies mise en œuvre.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id.** | **Composant applicatif** | **Justification de présence** | **Description** | **Technologie** |
| **A1** | FreshDesk | Ticketing | Outil de ticketing externe (SaaS). | *Propriétaire* |
| **A2** | WebApp LAE | UI | WebApp en SPA assurant la présentation et les interactions avec les composants d’interface utilisateur. | Vue.JS |
| **A3** | Backend LAE | Couche de service | API Restfull exposant la couche de service utilisée par la WebApp. Contient la logique métier de l’entreprise. | PHP 8 / Symfony – API Restfull |
| **A4** | AWS Load Balancer | Réparation de charge  Robustesse | Load balancer permettant la répartition du trafics entre deux instances AWS EC2 fournissant la WebApp. | *Propriétaire* |
| **A5** | AWS Web Application Firewall | Sécurité | Composant de sécurité visant à réduire les tentatives de cyberattaques. | *Propriétaire* |
| **A6** | AWS Cognito | Sécurité  Contrôle d’accès | IAM (Identity and Access Manager) assurant l’authentification, l’autorisation, et gérant les droits d’accès et les capacités des utilisateurs. | OAuth 2 / OIDC |
| **A7** | AWS S3 Storage | Stockage | Système de stockage pour les fichiers volumineux. | *Propriétaire* |
| **A8** | AWS API Gateway | Sécurité Évolutivité  Robustesse | API Gateway assurant la fédération des API et le contrôle des droits d’accès (contrôle des JWT OAuth 2) | *Propriétaire* |
| **A9** | AWS ElasticCache | Performance | Fourni un système centralisé de cache à accès rapide basé sur le moteur Redis. | Redis |
| **A10** | AWS Aurora DB | Stockage des données | Base de données relationnelle autoscallable, répliquée sur deux instances et sauvegardée quotidiennement et automatiquement sur AWS S3. | PostreSQL |

Tableau 10 : Catalogue des applicatifs de l'architecture cible

## Architecture technique (technical layer)

### Cartographie de la plateforme technique



La nouvelle plateforme technique est résolument basée sur la plateforme cloud AWS. L’utilisation du fournisseur AWS est justifié à la fois par le panel de service proposé, sa tarification qui le positionne avantageusement par rapport à la concurrence (MS Azure, Google Cloud …) et son actuelle présence dans l’entreprise nous assurant que le responsable informatique dispose d’une maitrise de l’infrastructure.

L’ensemble des serveurs physique de l’entreprise ont été supprimés pour alléger les tâches d’administration technique lié à l’exploitation et la maintenance (y compris des locaux adaptés à l’exploitation de serveurs).

Le SI exploite par ailleurs un outil de ticketing SaaS.

### Coût d’exploitation de la plateforme technique

Une évaluation du coût globale de fonctionnement de la plateforme est fourni dans le tableau ci-dessous.

Ce coût d’exploitation est à mettre en relation avec la suppression des coûts d’exploitation actuel (information non disponible au moment de la rédaction du présent document), des avantages en termes de sécurité, rapide et robustesse de la plateforme ainsi que de l’allégement des tâches du responsable informatique.

*Les données issues de se tableaux proviennent des estimations de dépenses pour l’outil de ticketing réalisé lors du comparatif des solution ainsi que de l’outil « AWZ Pricing Calculator », majorés de 20%, et dont les coûts ont été ramenés en EUR puis arrondis.*

|  |  |  |
| --- | --- | --- |
| **Poste de dépense** | **Dépenses mensuelles** | **Dépenses annuelles** |
| FreshDesk – 20 licences | 81,50 € | 980 € |
| AWS EC2 - WebApp LAE (t4g.large – 2 instances) | 58 € | 696 € |
| AWS EC2 - Backend de service LAE (t4g.xlarge – 3 instances) | 132 € | 1 584 € |
| AWS Aurora DB (Serverless – 1 To – 3 Instances) | 288 € | 3 456 € |
| AWS Web Application Firewall (WAF) | 33 € | 396 € |
| AWS ElastiCache Redis | 47 € | 564 € |
| AWS S3 Storage (3To Storage – 1,5 To E/S) | 120 € | 1 440 € |
| AWS Cognito | 3,50 € | 42 € |
| AWS API Gateway | 5,50 € | 66 € |
| AWS Load Balancing | 32 € | 384 € |
| AWS Cloud Drive - 5To | 50 € | 600 € |
| **Total pour l’exploitation de la plateforme** | **850,50 €** | **10 206 €** |

Tableau 11 : Évaluation des coûts d'exploitation de la plateforme technique

*Source :*   
*-* [*https://calculator.aws/#/estimate?id=15e7b56bd61300135ca182695d4ff077952dc2b2*](https://calculator.aws/#/estimate?id=15e7b56bd61300135ca182695d4ff077952dc2b2) *-* [*https://freshdesk.com/fr/pricing/*](https://freshdesk.com/fr/pricing/)

# ANALYSE DES ÉCARTS

### Écarts d’architecture

La présentation de l’architecture cible permet de visualiser un certain nombre d’écarts avec l’architecture de référence.

#### Business

Bien que les processus métiers de l’entreprise soit peu amené à évoluer, nous notons que le nouvel outil de ticketing va faciliter la collaboration inter-service et simplifier la prise de contact avec les clients par e-mail (création automatique de ticket).

Les processus métier du responsable informatique vont être révisés en automatisant les tâches techniques fastidieuse et en le laissant s’approprier de nouveaux processus à haute valeur ajoutée.

#### Data

Les données de l’entreprise ne seront pas amenées à évoluer. Le nouvel outil de ticketing viendra en remplacement des historiques clients basés sur la base MS Access.

#### Application

Une refonte profonde des applications de l’entreprise est nécessaire. Les applications utilisant des technologies obsolètes (légal, facturation …) seront supprimées. L’application de vente, récente, servira de base pour la construction de la nouvelle Web App commune à l’entreprise et d’un backend de service.

De nombreux composants de services (authentification, sécurité, cache …) seront déployés « clef-en-main » en utilisant les outils fournis par la plateforme AWS. Les développements à réalisés seront concentrés uniquement sur les applications à valeur métier : la Web App et le back-end de services.

Notons que la nouvelle base de données Aurora DB, déployée en tant que service (serverless) viendra en remplacement des différentes sources de données existante.

#### Technique

La plateforme technique sera désormais entièrement déployée dans le cloud AWS (certains composants l’étaient déjà). Plus aucun serveur physique ne sera nécessaire dans les locaux de l’entreprise pour le fonctionnement du SI.

### Risques de l’architecture

Le catalogue ci-dessous référence les risques associé à la nouvelle architecture et les mesures d’atténuation envisagées. (Voir annexes pour la terminologie utilisée)

*Notons que les risques associés à la mise en œuvre chacun des projets du chantier d’architecture feront l’objet d’une évaluation séparée et d’un suivi tout particulier lors de la rédaction des différents cahier des charges.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id.** | **Risque** | **Effet** | **Fréq.** | **Impact** | **Mesure d’atténuation** |
| **R1** | Capacité de développement des nouveaux outils / Respect des budgets. | Marginal | Seldom | Low | Libérer des crédits supplémentaire. Notons toutefois que le budget semble suffisant. |
| **R2** | Défaillance du fournisseur Cloud | Critical | Unlikely | Low | Prévoir un back-up chez un fournisseur alternatif. |
| **R3** | Vulnérabilité de l’outil de ticketing | Critical | Seldom | Moderate | Ajout de l’authentification 2FA obligatoire. Archivage régulier des historiques |
| **R4** | Vulnérabilité de la plateforme aux cyber-attaques | Critical | Seldom | Moderate | Renforcer les mesures de détection. Limitation des accès aux seules adresses IP de l’entreprise. |
| **R5** | Augmentation des coûts d’exploitation de la plateforme | Marginal | Occasional | Moderate | Les technologies utilisées étant standard, il est possible de prévoir un back-up chez un autre fournisseur. |
| **R6** | Adaptation du personnel à la nouvelle Web App et aux nouveaux outils. | Marginal | Occasional | Moderate | Prise en compte des retours utilisateur durant le cycle de développement, formation interne aux nouveaux outils. |
| **R7** | Absence de Plan de Continuité d'Activité / Absence de Plan de Reprise d'Activité. | Catastrophic | Occasional | High | Rédaction des plans et des procédures de reprise. |

Figure 19 : Catalogue d'évaluation des risques

# ARCHITECTURE DE TRANSITION

# ANNEXES

## Modèle de classification des risques

Le modèle d’évaluation des risques utilisé dans ce document utilise la terminologie définie par les publication de l’OpenGroup sur TOGAF :

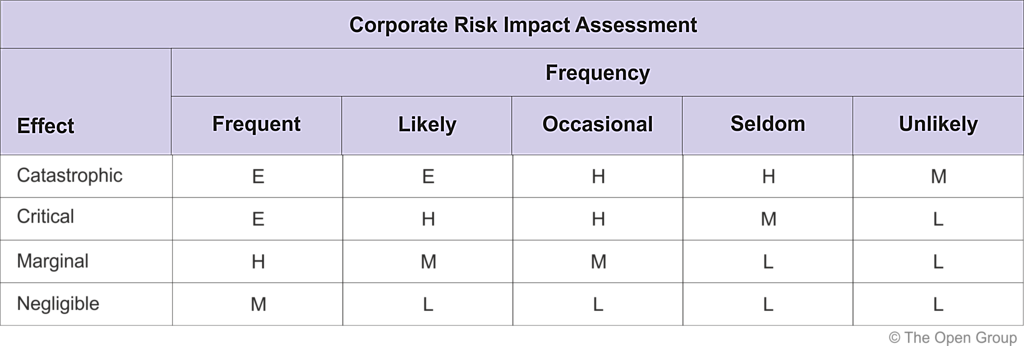


Figure 20 : Modèle TOGAF d'évaluation des risques

Source : <https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap27.html>

# APPROBATIONS

Le tableau ci-après liste toutes les parties prenantes ayant approuvé ce document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Poste** | **Signature** | **Date** |
| **David EVAN** | Architecte Logiciel | David Evan | 06/01/2022 |
| **Jeannette Bond** | CEO | *[En attente]* | *[En attente]* |
|  |  |  |  |

Tableau 12 : Approbation du document de définition de l'architecture

# TABLES DES RÉFÉRENCES

## Figures

[Figure 1 : Baseline Architecture - Cartographie des départements, des processus et des acteurs. 8](#_Toc92133507)

[Figure 2 : Business Baseline Architecture - Répondre à un client 9](#_Toc92133508)

[Figure 3 : Business Baseline Architecture - Enquêter 10](#_Toc92133509)

[Figure 4 : Business Baseline Architecture - Établir un contrat 11](#_Toc92133510)

[Figure 5 : Business Baseline Architecture - Établir une facture 12](#_Toc92133511)

[Figure 6 : Business Baseline Architecture - Définir une stratégie commerciale 13](#_Toc92133512)

[Figure 7 : Business Baseline Architecture - Prospecter des clients 14](#_Toc92133513)

[Figure 8 : Business Baseline Architecture - Vendre un produit d'assurance 15](#_Toc92133514)

[Figure 9 : Business Baseline Architecture - Sauvegarder quotidiennement les données 16](#_Toc92133515)

[Figure 10 : Data Baseline Architecture - Cartographie des données 18](#_Toc92133516)

[Figure 11 : Application Baseline Architecture - Cartographie des applicatifs 20](#_Toc92133517)

[Figure 12 : Technical Baseline Architecture - Cartographie de la plateforme technique 22](#_Toc92133518)

[Figure 13 : Prototype simplifiée de la nouvelle architecture 24](#_Toc92133519)

[Figure 14 : Business Target Architecture - Répondre à un client 29](#_Toc92133520)

[Figure 15 : Business Target Architecture - Controller la sauvegarde et la restauration des données 30](#_Toc92133521)

## Tableaux

[Tableau 1 : Historique des révisions 2](#_Toc92133567)

[Tableau 2 : Catalogue des objectifs du chantier d’architecture 7](#_Toc92133568)

[Tableau 3 : Catalogue des contraintes du chantier d’architecture 7](#_Toc92133569)

[Tableau 4 : Business Baseline Architecture - Évaluation des processus d'entreprise 17](#_Toc92133570)

[Tableau 5 : Application Baseline Architecture - Évaluation des applicatifs de l'entreprise 21](#_Toc92133571)

[Tableau 6 : Technical Baseline Architecture - Évaluation de la plateforme technique 22](#_Toc92133572)

[Tableau 7 : Table de comparaison des outils de ticketing du marché 26](#_Toc92133573)

[Tableau 8 : Catalogue des exigences de l'architecture cible (1/2) 27](#_Toc92133574)

[Tableau 9 : Catalogue des exigences de l'architecture cible (2/2) 28](#_Toc92133575)

[Tableau 10 : Approbation du document de définition de l'architecture 36](#_Toc92133576)

1. RPO – Recovery Point Objective. Perte de données maximale admissible pour le SI en cas à la suite d’un incident. [↑](#footnote-ref-1)
2. Des suppressions de données non autorisés impliquants des membres de la société sans droits d’accès à partir d’une simple présence physique ont été relevés. [↑](#footnote-ref-2)
3. Base de données relationnelle basée dans le cloud AWS. [↑](#footnote-ref-3)