

# **Cadre de l'architecture sur mesure**

## **Application de Communication Multiplateforme**



## Auteur(s) et contributeur(s)

Nom & Coordonnées	Qualité & Rôle	Société
Gérald ATTARD	Architecte logiciel	LAE Les Assureurs Engagés

## Historique des modifications et des révisions

N° version	Date	Description et circonstance de la modification	Auteur
1.0	13/08/2022	Création du document	Gérald ATTARD

## Validation

N° version	Nom & Qualité	Date & Signature	Commentaires & Réserves
1.0	BOND Jeannette CEO de LAE		

## Tableau des abréviations

Abr.	Sémantique
IT	Information Technology (trad. <i>technologie de l'information</i> )
LAE	Les Assureurs Engagés
OS	Operating System (trad. <i>système d'exploitation</i> )
ROI	Return On Investissement (trad. <i>retour sur investissement</i> )
SaaS	Software-as-a-Service (trad. logiciel en tant que service)
SI	Système d'information
SLO	Service Level Objectives (trad. <i>objectifs de niveau de service</i> )
SQL	Structured Query Language (trad. <i>langage de requête structuré</i> )
TCO	Total Cost of Ownership (trad. <i>coût total de possession</i> )
US	User's Story (trad. <i>histoire utilisateur</i> )
VM	Virtual Machine (trad. Machine virtuelle)
VMM	Virtual Machine Monitoring (trad. <i>moniteur de machine virtuelle</i> )

## Table des matières

I. Méthodes d'architecture sur mesure.....	4
I.A. Le <i>réhébergement</i> .....	5
I.B. La <i>refactorisation</i> .....	5
I.C. La <i>re-plateforme</i> .....	5
I.D. Le retrait et le remplacement.....	5
II. Contenu d'architecture sur mesure.....	6
II.A. Livrables de Documentation.....	6
II.B. Livrables de Production.....	9
III. Outils de migration.....	12
III.A. Serveur dédié.....	13
III.B. Serveur virtualisé.....	14
III.C. Hyperviseur.....	15
IV. Interfaces avec les modèles de gouvernance et autres cadres.....	17
IV.A. Planification d'entreprise.....	17
IV.A.1. Plan de migration.....	17
IV.A.2. Déploiement de la migration.....	17
IV.A.3. Services gérés dans le cloud.....	17
IV.A.4. Modernisation des applications.....	17
IV.B. L'architecture d'entreprise.....	18
IV.B.1. Des défis techniques imprévus.....	19
IV.B.2. Des coûts imprévus.....	19
IV.B.3. Des temps d'arrêt imprévus.....	19
IV.B.4. Des problèmes culturels ou difficultés de gestion du changement.....	19
IV.C. Programme.....	20
IV.D. Développement de système / Ingénierie.....	20
IV.D.1. Phase 1 : Identification et évaluation des candidatures.....	21
IV.D.2. Phase 2 : évaluation du coût total de possession (TCO).....	21
IV.D.3. Phase 3 : évaluer le risque global et la durée du projet.....	22
V. Éligibilité à la migration.....	23



## I. Méthodes d'architecture sur mesure

Tel qu'il est décrit au sein du *Document de Définition d'Architecture*, cette migration sera réalisée de façon progressive et ne présentera qu'un minimum d'interruption de services pour les collaborateurs de LAE, et à fortiori, pour les clients.

La nouvelle architecture répondra aux objectifs, définis par LAE, suivants :

- réduire les temps de traitement des données, dont leurs accès ;
- sécuriser aussi bien les données que les moyens avec lesquels elles sont traitées ;
- garantir l'intégrité des données ;
- faire preuve de robustesse en cas d'incident.

Cette migration, c'est à dire le passage de l'ancien SI, décentralisée et désynchronisée, vers un SI centralisé assurant l'intégrité des données détenues, se déroulera selon deux thématiques différentes :

- la migration applicative ;
- la migration d'hébergement.

En ce qui concerne spécifiquement la migration d'hébergement, ce questionnement sera abordé plus loin dans ce document.

En ce qui concerne la migration applicative au sein de LAE, elle correspondra au processus de déplacement de toutes les solutions logicielles, utilisées indépendamment dans chaque service, vers un environnement informatique centralisé.

Ce processus se déroulera à déplaçant chacun des centres de données vers l'environnement d'un fournisseur de cloud, qu'il soit privé ou public ; ce choix stratégique étant laissé aux parties prenantes décisionnelles.

Étant donné que les applications utilisées actuellement sont conçues pour s'exécuter sur des OS particuliers, dans des architectures réseau spécifiques, le déplacement d'une application vers un nouvel environnement peut poser un certain nombre de défis.

Il est généralement plus facile de migrer des applications à partir d'architectures virtualisées ou basées sur des services, que de migrer celles qui s'exécutent sur du matériel *bare metal* sur site local. En ce qui concerne LAE, l'environnement décentralisé actuel va constituer un véritable défi et nécessitera une coordination d'ensemble précise et structurée.

La détermination de cette stratégie globale de migration d'applications impliquera de prendre en compte les dépendances et les exigences techniques de chaque application, ainsi que les contraintes de sécurité, de conformité et de coût de LAE.

Différentes applications pourront alors emprunter des chemins différents vers une solution cloud, même au sein d'un seul environnement technologique cible.

Les parties prenantes décisionnelles de LAE devront alors effectuer des choix entre plusieurs modèles de migration d'applications, tels que :

- le *réhébergement* ;
- la *refactorisation* ou restructuration ;
- la *re-plateforme* ;
- le retrait et remplacement.

Le quatre options ci-dessus vont être définies dans les paragraphes qui suivent.

## **I.A. Le réhébergement**

Le *réhébergement*, également appelée *lift-and-shift*, est une stratégie courante dans laquelle l'entreprise déplacera l'application d'un serveur sur site vers une machine virtuelle, dans le cloud, sans apporter de modifications importantes.

Le *réhébergement* d'applications est généralement plus rapide que les autres stratégies de migration et peut réduire considérablement les coûts de migration.

L'inconvénient de cette option réside justement dans l'absence de modification des applications considérées. En effet, ces applications ne bénéficieront pas des capacités informatiques natives du cloud, et les coûts à long terme de leur exécution, dans le cloud, pourraient finir par être plus élevés.

## **I.B. La refactorisation**

La *refactorisation*, ou restructuration, consiste à apporter des modifications assez importantes à l'application afin qu'elle puisse évoluer ou mieux fonctionner dans un environnement cloud.

Cela peut impliquer de recoder des parties importantes de l'application afin qu'elle puisse mieux tirer parti des fonctionnalités natives du cloud, telles que la restructuration d'une application monolithique en un ensemble de microservices, ou la modernisation d'un magasin de données de SQL à NoSQL.

## **I.C. La re-plateforme**

La *re-plateforme* est une solution de compromis entre le *lift-and-shift* et la *refactorisation*.

En effet, la re-plateforme d'une application consiste à lui apporter des modifications mineures afin qu'elle puisse bénéficier au mieux de l'architecture cloud.

Les exemples peuvent inclure la mise à niveau de l'application pour qu'elle fonctionne avec une base de données gérée nativement dans le cloud, la modification des OS ou du *middleware* avec lesquels elle fonctionnera, ou la conteneurisation de l'application elle-même.

## **I.D. Le retrait et le remplacement**

Dans certains cas, il est tout simplement plus cohérent de mettre l'application purement et simplement hors service, le temps de réaliser la migration.

Cela peut être dû au fait que sa valeur est limitée, que ses fonctionnalités sont dupliquées ailleurs dans l'environnement ou qu'il est plus rentable de la remplacer par une nouvelle offre, telle qu'une plate-forme SaaS, qui consistera par la suite à migrer l'application.



## II. Artefact et livrables

### II.A. Dette technique

Dès la phase de réflexion relative à l'inadéquation du SI actuel de LAE, il a été identifié la présence d'une dette technique. Il sera à considérer, tout au long de cette étude, que cette dette technique n'est pas forcément une mauvaise chose ou un mauvais point sur lequel porter une attention dédaigneuse...

Dans un premier temps, la dette technique est toujours comparée à une dette financière. Néanmoins, à la différence d'une dette financière, une dette technique n'a pas à être payée, financièrement parlant. Dans certaines situations ou contextes, rembourser cette dette technique n'a, tout simplement, aucun sens : certaines parties de codes ou de logiciels n'étant jamais lues ou utilisées... Par conséquent, la dette technique doit aussi prendre en compte la probabilité événementielle : quelle est la possibilité et/ou la fréquence d'une future modification d'un morceau de code sale ?

Dans le monde financier, la dette n'est également pas une mauvaise chose. En effet, si le règlement de la dette, plus les intérêts, est inférieur au ROI alors cette dette sera profitable. Par exemple, s'endetter pour payer une maison est considéré comme *responsable* si le souscrivait sait comment rembourser son prêt.

Ce principe est également vrai dans l'industrie du logiciel. Par exemple, si la qualité interne est sacrifiée pour être le premier sur le marché, cette action pourra être considérée comme gagnante si l'avantage obtenu, grâce à cette décision, est plus élevée que le coût de retard sur le marché avec une meilleure qualité interne. Il faut donc, ici aussi, peser le Pour et le Contre d'une telle décision.

En complément, peu importe le domaine, il y aura cependant toujours un risque relatif à la difficulté d'estimer en avance ces bénéfices, puisqu'il y aura forcément un degré d'incertitude.

En outre, il convient d'acter le fait que la dette technique présente actuellement au sein de LAE ne peut être ignorée. Peu importe les fonctions des collaborateurs, techniques ou pas, tous les membres de LAE devront avoir un intérêt à gérer la dette technique, aussi intelligemment que possible, pour obtenir le meilleur équilibre entre succès à court terme et à -long terme.

Au sein de cette étude, nous considérerons deux approches qui peuvent être appliquées au contexte de LAE :

- le *backlog* technique,
- l'inclusion du coût de la dette technique dans les estimations.

Ces deux approches nécessiteront la définition de deux autres processus critiques :

- les tâches tampons de *refactorisation*,
- les versions de nettoyage.

A ces artefacts et méthodes, cette migration tiendra également compte de trois stratégies :

- le paiement de la dette,
- la conversion de la dette,
- le paiement des intérêts.

Ces trois notions seront à appliquer tout au long de la migration.

L'approche choisie pour appréhender la dette technique sera AVANT TOUTE CHOSE une décision du business.

## II.B. Livrables

En outre, durant tout le déroulé de ce projet de migration plusieurs livrables seront produits pour différentes parties prenantes intéressées, autant en termes de décisionnel qu'en termes d'utilisation métier.

Il convient alors de définir les deux types de livrable à considérer :

- **les livrables de type Documentation** à fournir aux parties prenantes décisionnelles pour préciser l'avancée de la migration ;
- **les livrables de type Production** à fournir :
  - dans un premier temps, aux parties prenantes fonctionnelles pour qu'elles s'assurent de l'aspect métier (test fonctionnel) ;
  - dans un deuxième temps, aux différents collaborateurs de LAE pour les informer de la mise en œuvre et de l'utilisation de la nouvelle solution applicative.

## II.C. Livrables de Documentation

Les livrables relatifs à la documentation d'avancée de la migration seront des catalogues, des matrices et des diagrammes que l'équipe d'architecture devra produire et valider avec chaque groupe de parties prenantes, pour fournir un modèle d'architecture efficace.

Il est important d'accorder une attention particulière aux intérêts des parties prenantes en définissant des catalogues, des matrices et des diagrammes spécifiques qui soient pertinents pour le modèle d'architecture particulier de LAE.

Ces rédactions permettront de communiquer l'architecture afin qu'elle soit comprise par toutes les parties prenantes, tout en leur permettant de vérifier que l'initiative d'architecture d'entreprise réponde à leurs préoccupations.

Le tableau ci-dessous répertorie les livrables documentaires à fournir à chaque partie prenante.

Nom Prénom	Fonction	Catalogue, Matrices et schémas à fournir
Antoine DURAND	Responsable informatique	<ul style="list-style-type: none"> <li>• Diagramme de l'empreinte commerciale</li> <li>• Diagramme But/Objectif/Service commercial</li> <li>• Diagramme de décomposition de l'organisation</li> <li>• Catalogue des capacités métier</li> <li>• Matrice capacité/organisation</li> <li>• Carte des capacités de l'entreprise</li> <li>• Matrice Stratégie/Capacité</li> <li>• Matrice capacité/organisation</li> <li>• Diagramme du modèle d'affaires</li> <li>• Catalogue de flux de valeur</li> <li>• Catalogue des étapes de la chaîne de valeur</li> <li>• Matrice de flux de valeur/capacité</li> <li>• Carte de la chaîne de valeur</li> </ul>
Jeannette BOND	CEO	<ul style="list-style-type: none"> <li>• Catalogue des exigences</li> <li>• Diagramme du contexte du projet</li> <li>• Diagramme des avantages</li> <li>• Diagramme de l'empreinte commerciale</li> <li>• Diagramme de communication d'application</li> <li>• Carte de l'organisation</li> <li>• Catalogue des capacités métier</li> <li>• Matrice capacité/organisation</li> <li>• Carte des capacités de l'entreprise</li> <li>• Matrice Stratégie/Capacité</li> <li>• Matrice capacité/organisation</li> <li>• Diagramme du modèle d'affaires</li> <li>• Catalogue de flux de valeur</li> <li>• Catalogue des étapes de la chaîne de valeur</li> <li>• Matrice de flux de valeur/capacité</li> <li>• Carte de la chaîne de valeur</li> </ul>

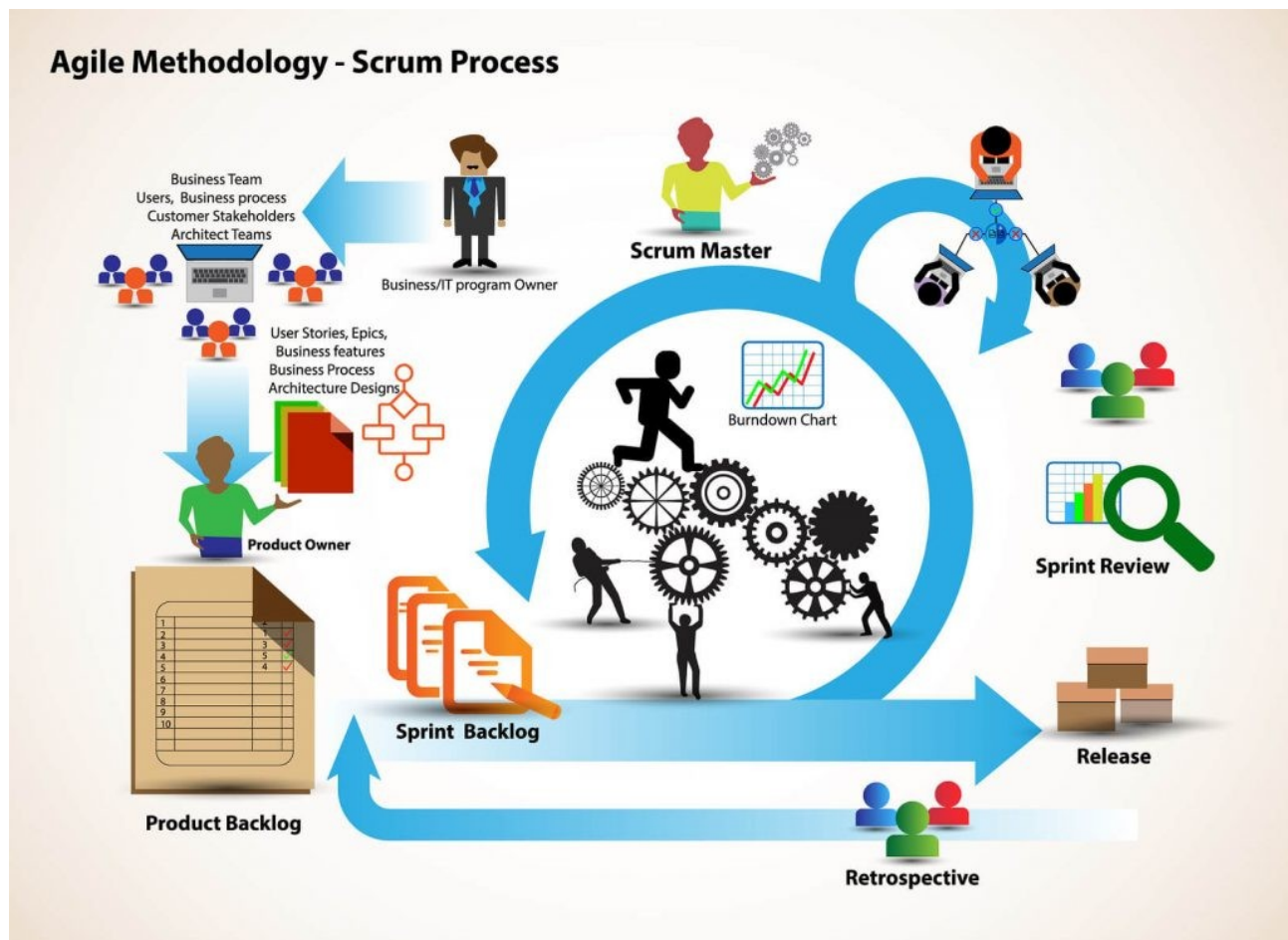


## II.D. Livrables de Production

Ce projet de migration sera g  r      partir de la m  thode **SCRUM**, telle que d  taill  e dans le *SCRUM guide*, de Juillet 2013, par *Ken Schwaber* et *Jeff Sutherland*.

Cette m  thode it  rative et incr  mentale, chronologiquement d  taill  e ci-dessous, est caract  ris  e par 3 notions fondamentales :

- Les sprints (it  rations) et leurs   v  nements (c  r  monies) ;
- Les 3 r  les cl  s (*Product Owner*, *Scrum Master*, et *devTeam* -   quipe de d  veloppeurs) ;
- Les art  facts (*Product Backlog*, *Sprint Backlog*, *incr  ment*, *Definition Of Done*, *Definition Of Ready*, *User Story*...).



La constitution de l'  quipe AGILE (les 3 r  les mentionn  es supra) sera trait  e comme une t  che ind  pendante, en tant qu'  pic US dans le *Product Backlog*. Cette   quipe sera constitu  e de :

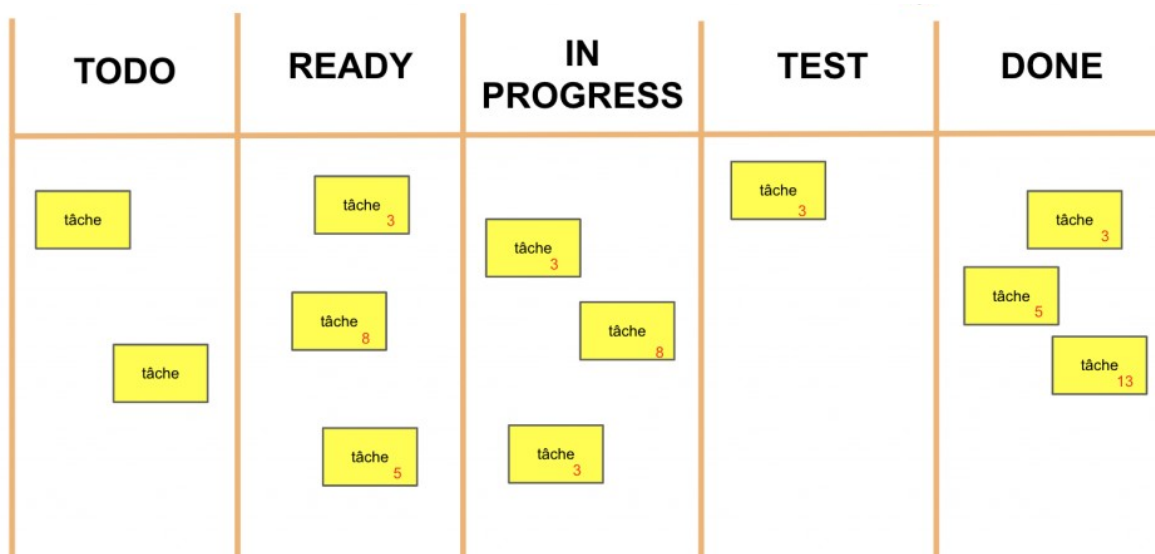
- 1 *Product Owner* ;
- 1 *Scrum Master* ;
- 8 d  veloppeurs constituant la *devTeam*.

Toutes les tâches, aussi bien fonctionnelles que techniques, seront centralisées au sein de tableaux de bord nommés :

- *Product Backlog*, pour les tâches fonctionnelles de haut niveau (*epic US*) ;
- *Sprint Backlog*, pour leurs décompositions techniques (*US*).

Chacune des *epic US* mentionnées supra sera décomposée en différentes *US techniques* qui seront traitées durant différents *Sprints* (jalon temporel) définis.

Ces *Sprints* seront planifiés au sein d'un *Sprint Backlog*, organisé et validé, en début de jalon temporel, par l'ensemble de l'équipe Scrum, tel que représenté dans l'exemple suivant :



Les post-it jaunes représentent des US ayant le formalisme suivant :

« *En tant que...* ;  
*je veux que...* ;  
*afin de... ..* »

Lors des cérémonies de *Sprint Planning* et de *Product Backlog Refinement*, l'équipe de développement, guidée par le *Scrum Master*, réalisera des sessions de *poker planning* afin d'estimer la charge Quantitative et Qualitative, nommée **Point d'Effort**, de chaque US proposée ; cette charge est spécifiée sur chaque US de l'exemple ci-dessus par un chiffre rouge.

Ces Points d'Effort, relatifs à une US, inclueront différentes notions que la *devTeam* prendra en compte pour faire l'estimation de celle-ci, à savoir :

- l'effort à faire pour développer la demande ;
- la complexité que peut comporter la demande ;
- les risques anticipés lors du développement de l'US ;
- les éventuelles inconnues existant au moment de l'estimation ;
- les dépendances potentielles avec des éléments extérieurs.

Enfin, chaque US devra avoir répondu au **DOR** (*Definition Of Ready*) pour être prise en compte dans le *Sprint Backlog* lors de la cérémonie de *Sprint Planning*.

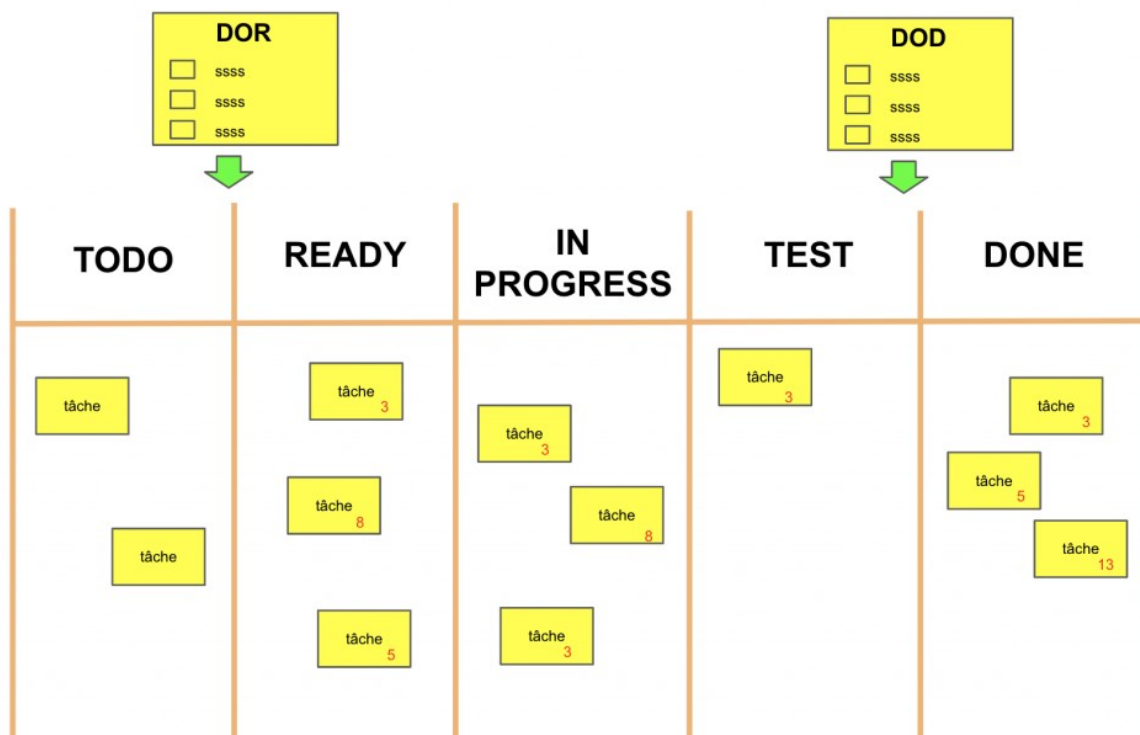
La méthode Scrum étant itérative et incrémentale, **l'incrément de chaque sprint sera considéré comme un livrable de production.**

Ainsi, **chaque livrable de production sera présenté pour validation à l'ensemble des parties prenantes** lors de la cérémonie de *Sprint Review*.

Une US sera considérée comme '**livrable**', si et seulement si, elle remplira les critères de la **DOD** (*Definition Of Done*).

Vous trouverez ci-dessous un exemple de *Sprint Backlog* avec la séquence d'application du **DOR** et du **DOD**.

Le schéma ci-dessous représente un *Sprint Backlog* d'un Sprint géré par le DevTeam.



En outre, lors de la cérémonie de *Product Backlog Refinement*, le *Product Owner* pourra proposer des **fonctionnalités** qui auront été **validées EN AMONT par les parties prenantes** principales, pour être prises en compte dans le(s) sprint(s) ultérieur(s).



### III. Outils de migration

La virtualisation est un élément fondamental de nombreuses stratégies de migration vers le cloud.

En ce sens, les machines virtuelles peuvent facilement être exécutées dans de nouveaux environnements matériels physiques. Il est même possible de déplacer une application en direct (s'exécutant sur une machine virtuelle) entre des machines hôtes physiques sans perturber l'expérience de l'utilisateur final.

La flexibilité et la polyvalence des environnements informatiques virtualisés simplifient considérablement le processus de migration des applications.

Pour réaliser ces processus, il existe plusieurs types d'hyperviseurs et d'options de migration.

Aussi, plusieurs solutions de réplication et de migration sont actuellement disponibles et permettent à leurs clients de migrer des machines virtuelles entre des serveurs *bare metal*, des serveurs virtuels dans le cloud et même des hyperviseurs :

- **migration d'applications virtualisées** : il est possible de transférer des machines virtuelles s'exécutant sur des instances sur site vers un serveur virtualisé s'exécutant dans un cloud privé sans interrompre les opérations, provoquer des temps d'arrêt ou nécessiter une reconfiguration des applications.
- **migration d'applications sur site** : il existe des COTS proposant des boîtes à outils de migration d'applications, qui sont des solutions logicielles personnalisables et extensibles. Ces dernières analysent les environnements informatiques pour identifier les interdépendances des applications. Ils fournissent des rapports d'analyse de type tableau de bord pour mettre en évidence les applications susceptibles de rencontrer des problèmes lors de la migration.

Suite à ces migrations, il est alors nécessaire de s'assurer qu'aucune donnée ou fonctionnalité n'est perdue pendant le processus de migration de l'application.

Il faudra alors effectuer des tests pendant la migration pour vérifier que :

- toutes les données sont présentes,
- leur intégrité a été maintenue,
- elles se trouvent maintenant dans le bon emplacement de stockage.

Il est également essentiel d'effectuer des tests de suivi une fois la migration terminée afin d'évaluer les performances des applications et de s'assurer que les contrôles de sécurité restent en place.

En outre, afin de donner une base de vocabulaire commun à la lecture de cette étude, les notions qui suivantes seront définies au sein des paragraphes à venir :

- serveur dédié ou *bare metal* ;
- serveur virtualisé ;
- hyperviseur.

### III.A. Serveur dédié

Les serveurs dédiés ou *bare metal* sont une forme de services cloud dans lesquels l'utilisateur loue une machine physique, auprès d'un fournisseur, qui n'est partagée avec aucun autre locataire.

Contrairement au *cloud computing* traditionnel, qui repose sur des machines virtuelles, les serveurs dédiés ne sont pas livrés avec un hyperviseur préinstallé et donnent à l'utilisateur un contrôle total sur son infrastructure de serveur.

Avec un serveur dédié, parce que les utilisateurs obtiennent un contrôle total sur la machine physique, ils ont la possibilité de choisir leur propre OS.

Ainsi, cette infrastructure dédiée permet d'éviter les potentiels problèmes de "voisin bruyant" d'une infrastructure partagée et d'ajuster finement le matériel et les logiciels pour des besoins spécifiques, souvent gourmands en données et charges de travail.



Ce modèle s'apparente aux solutions cloud de type IaaS qui est un modèle où l'entreprise dispose, sur abonnement payant, d'une infrastructure informatique (serveurs, stockage, sauvegarde, réseau) qui se trouve physiquement chez le fournisseur ; ce dernier étant alors responsable de la sécurité de l'infrastructure.

Cette solution peut représenter un moyen de réaliser des économies, principalement en transformant des investissements en contrats de location.

## III.B. Serveur virtualisé

Les ordinateurs serveurs, c'est à dire les machines qui hébergent des fichiers et des applications sur les réseaux informatiques, se doivent d'être puissants.

Certains ont des unités centrales de traitement (CPU) à processeurs multiples qui permettent à ces serveurs d'exécuter facilement des tâches complexes.

Les administrateurs de réseaux informatiques dédient généralement chaque serveur à une application ou une tâche spécifique. Beaucoup de ces tâches ne sont pas compatibles avec d'autres ; chacune nécessitant une machine dédiée.

Une application par serveur facilite également le suivi des problèmes au fur et à mesure qu'ils surviennent. C'est un moyen simple de rationaliser un réseau informatique d'un point de vue technique.

Cette approche pose cependant quelques problèmes.

D'abord, elle ne tire pas partie de la puissance de traitement des ordinateurs serveurs modernes. La plupart des serveurs n'utilisent qu'une petite fraction de leur capacité de traitement globale.

De plus, à mesure qu'un réseau informatique devient plus grand et plus complexe, les serveurs commencent à prendre beaucoup d'espace physique. Un centre de données peut rapidement devenir surpeuplé, avec des racks de serveurs consommant beaucoup d'énergie et générant de la chaleur.

La virtualisation des serveurs tente de résoudre ces deux problèmes d'un seul coup :

- en utilisant un logiciel spécialement conçu à cet effet, un administrateur peut convertir un serveur physique en plusieurs machines virtuelles. Chaque serveur virtuel agit comme un dispositif physique unique, capable d'exécuter son propre OS. En théorie, il serait alors possible de créer suffisamment de serveurs virtuels pour utiliser toute la puissance de traitement d'une machine, bien qu'en pratique ce ne soit pas toujours la meilleure idée.

La virtualisation n'est pas un concept nouveau. Les informaticiens créent des machines virtuelles sur des superordinateurs depuis des décennies. Cependant, il y a seulement quelques années que la virtualisation est devenue possible pour les serveurs. Dans le monde de l'IT, la virtualisation des serveurs est un sujet d'actualité. C'est une technologie encore jeune et plusieurs entreprises proposent différentes approches.

### III.C. Hyperviseur

Avant que les hyperviseurs ne soient démocratisés et n'atteignent le grand public, la plupart des ordinateurs physiques ne pouvaient exécuter qu'un seul OS à la fois. Cet état de fait les a rendu stables car le matériel informatique n'avait qu'à gérer les demandes de ce seul OS.

Néanmoins, l'inconvénient de cette approche était qu'elle gaspillait des ressources car le système d'exploitation ne pouvait pas toujours utiliser toute la puissance de l'ordinateur ; un hyperviseur résout ce problème.

En effet, il s'agit d'une petite couche logicielle qui permet à plusieurs systèmes d'exploitation de fonctionner côte à côte, en partageant les mêmes ressources informatiques physiques.

Ces OS se présentent sous la forme de VM, c'est-à-dire de fichiers qui imitent un environnement matériel informatique complet dans un logiciel.

L'hyperviseur, connu aussi sous le nom de VMM, gère ces machines virtuelles lorsqu'elles fonctionnent côte à côte. Il sépare logiquement les machines virtuelles les unes des autres, en attribuant à chacune sa propre tranche de la puissance de calcul, de la mémoire et du stockage sous-jacents.

Ce procédé empêche les VM d'interférer les unes avec les autres ; donc si, par exemple, un OS subit un crash ou une compromission de sécurité, les autres systèmes d'exploitation partageant la même machine physique n'en sont pas du tout impactés et survivent.

Il existe différentes catégories d'hyperviseurs et différentes marques d'hyperviseurs dans chaque catégorie. Le marché a mûri pour faire des hyperviseurs un produit de base dans l'espace de l'entreprise.

Aussi, afin de faire un choix cohérent avec le contexte de LAE, les parties prenantes devront considérer des facteurs de différenciation qui devraient guider leur choix, tels que :

- **les performances** : il s'agit ici des performances de l'hyperviseur dans un environnement de production. Idéalement, les hyperviseurs *bare metal* devraient prendre en charge des performances d'OS invité proches des vitesses natives.
- **l'écosystème** : une bonne documentation et un support technique adéquat seront indispensables pour mettre en œuvre et gérer des hyperviseurs sur plusieurs serveurs physiques à grande échelle. L'accès à une communauté saine de développeurs tiers, pouvant prendre en charge l'hyperviseur avec leurs propres agents et plug-ins, offre également des possibilités fonctionnelles, telles que l'analyse de la capacité de sauvegarde, de restauration et de la gestion du basculement.
- **les outils de gestion** : l'exécution de machines virtuelles n'est pas la seule notion qu'il faudra gérer lors de l'utilisation d'un hyperviseur. Il faudra également provisionner les machines virtuelles, les entretenir, les auditer et nettoyer celles qui ne sont plus utilisées pour empêcher la "prolifération des machines virtuelles". Il faudra alors s'assurer que le fournisseur ou la communauté tierce prend en charge l'architecture de l'hyperviseur avec des outils de gestion complets.
- **la migration en direct** : cette fonctionnalité permet de déplacer des machines virtuelles entre des hyperviseurs sur différentes machines physiques sans les arrêter, ce qui peut être utile à la fois pour le basculement et l'équilibrage de la charge de travail.
- **le coût** : lors du choix de l'hyperviseur, il faudra considérer la structure des coûts et les frais impliqués dans l'appropriation de licences pour la technologie de l'hyperviseur. Il ne faudra donc pas penser qu'au coût de l'hyperviseur lui-même. En effet, le logiciel de gestion qui le rend évolutif pour prendre en charge un environnement d'entreprise peut souvent être coûteux. En outre, l'examen de la structure de licence du fournisseur peut changer selon que LAE choisira de déployer sa solution dans le cloud ou localement.





## **IV. Interfaces avec les modèles de gouvernance et autres cadres**

### **IV.A. Planification d'entreprise**

De nombreux services seront disponibles pour aider LAE à élaborer une stratégie, planifier et exécuter une migration vers le cloud réussie.

#### **IV.A.1. Plan de migration**

Dans le cadre d'une offre de service complète, le fournisseur de cloud pourra aider à :

- clarifier aussi bien la stratégie et que les objectifs de migration ;
- collecter des informations sur les applications et leur(s) environnement(s) ;
- identifier les besoins des collaborateurs et leur(s) exigence(s) commerciale(s) ;
- proposer un plan d'action détaillé pour la migration en elle-même.

#### **IV.A.2. Déploiement de la migration**

En optant pour une solution de déploiement géré, le fournisseur de cloud pourra aider LAE à :

- élaborer une stratégie ;
- à planifier cette migration ;
- gérer la migration elle-même ainsi que les tests et dépannages associés.

Cette offre de service clé en main comprend une assistance complète de bout en bout.

#### **IV.A.3. Services gérés dans le cloud**

Une offre de services cloud gérés comprend la surveillance et la maintenance de l'environnement informatique basé sur le cloud.

Le fournisseur de services cloud assumera alors la responsabilité de plusieurs fonctions, dont la gestion de la sécurité du cloud et l'achat potentiel d'offres en tant que service auprès d'autres fournisseurs, au nom de LAE.

La migration d'applications peut être incluse dans une offre de services *packagés* ou ajoutée à la carte.

#### **IV.A.4. Modernisation des applications**

Les services de modernisation des applications incluent des offres de développement personnalisées qui peuvent préparer les applications héritées pour une utilisation dans le cloud.

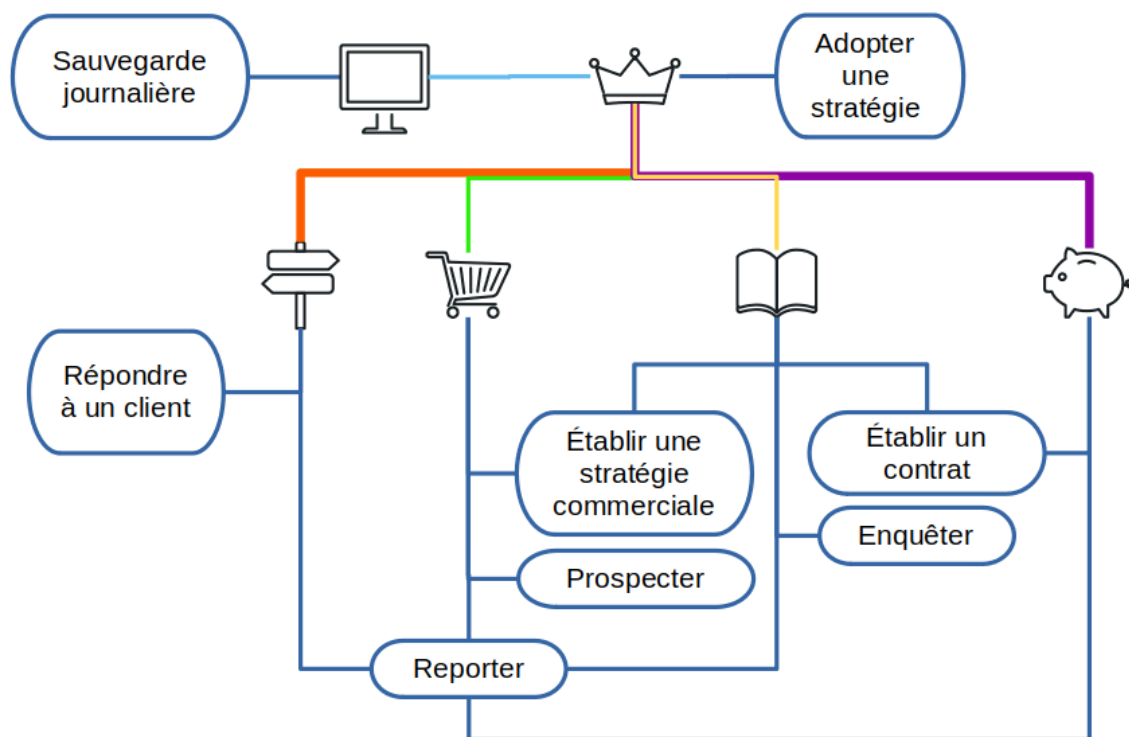
Ce principe consiste à modifier les applications source afin qu'elles s'exécutent dans des conteneurs ou des environnements virtualisés.

## IV.B. L'architecture d'entreprise

Pour développer la stratégie de migration d'applications la mieux adaptée à l'environnement informatique et aux besoins commerciaux uniques de LAE, les parties prenantes décisionnelles sont invitées à prendre en considération le *Document de Définition d'Architecture* qui définit les notions suivantes :

- les contraintes organisationnelles, financières, techniques et temporelles associées à cette migration ;
- les principes architecturaux à mettre en place au sein de la nouvelle solution architecturale ;
- le contenu du portefeuille d'applications de LAE ;
- les spécificités des modèles d'architecture constituant la future solution, ainsi que leurs justifications ;
- le mappage entre les modèles préconisés et diverses normes existantes ;
- l'impact et les états transitionnels relatifs à la progressivité de l'opération.

Outre ces éléments, le *Document de Définition d'Architecture* met également en avant la structure organico-fonctionnelle, telle que schématisée ci-dessous :



Ainsi, les parties prenantes devront non seulement prendre en considération les éléments factuels énumérés ci-dessus et, également, les stratégies de migration vers un cloud qui hébergera finalement la future solution.

Ce choix impactera la représentation des futures infrastructures de stockage, de calcul et de réseau de LAE.

Pour que cette migration vers le cloud réussisse, les parties prenantes de LAE devront également statuer, de façon claire, sur les principaux moteurs commerciaux qui motivent leur entreprise et aligner leur(s) stratégie(s) sur ces moteurs. Elles devront être conscientes de la raison pour laquelle cette migration vers le cloud a été opérée et de ce qu'elles espèrent réaliser avec, suite à la transition.

A l'aide d'un planning de migration clair et précis, les parties prenantes de LAE n'auront pas à craindre les perturbations d'activité engendrées par les opérations de migration d'applications, qui pourraient entraîner des coûts imprévus.

Aussi, les risques les plus courants sont relatifs aux notions suivantes et développés plus bas :

- des défis techniques imprévus ;
- des coûts imprévus ;
- des temps d'arrêt imprévus ;
- des problèmes culturels ou des difficultés de gestion du changement.

#### **IV.B.1. Des défis techniques imprévus**

Les défis techniques à relever par les équipes techniques pourront être de plusieurs types. Aussi, les parties prenantes de LAE pourraient avoir des réticences vis à vis de ceux-ci et, surtout, de leur impact sur le planning de déploiement.

Par exemple, une application pourrait avoir tellement de dépendances que la *refactorisation* ou la refonte de la plate-forme pourrait être beaucoup plus complexe et chronophage que cela n'aurait été prévu lors de la réflexion prévisionnelle initiale.

#### **IV.B.2. Des coûts imprévus**

Sans une planification adéquate, LAE pourrait engager des dépenses qu'elle n'avait pas budgétées, telles que les nouveaux frais de licence ou les coûts de formation associés à la mise à niveau des employés sur les nouveaux outils.

Aussi, en admettant qu'un partenariat avec un prestataire de cloud ne soit acté, les parties prenantes devront s'assurer que de tels débordements sont prévus contractuellement et pris en charge.

Par exemple, il serait possible de prévoir cette prise en charge par une assurance complémentaire qui pourrait être une compensation en nature ou en service de la part du prestataire.

### **IV.B.3. Des temps d'arrêt imprévus**

Relativement à certaines solutions architecturales, des modifications majeures devront être apportées à une application.

Ces altérations de l'application initiale pourraient entraîner des conflits ou des problèmes entraînant des temps d'arrêt imprévus, tant pour l'application que pour les systèmes connectés ou dépendants.

Des solutions parallèles devront alors être prévues pour minimiser ces interruptions de temps de service. D'ailleurs, ces derniers devront être clairement et spécifiquement définis en tant que SLO (objectif de niveau de services) afférents à ce projet.

### **IV.B.4. Des problèmes culturels ou difficultés de gestion du changement**

Il a été définie dans le *Document de Définition d'Architecture* que chaque service de LAE utilise ses propres solutions applicatives différemment et indépendamment les uns des autres.

Ces différences peuvent engendrer des frictions qui ralentissent un projet de migration.

Entreprendre une évaluation minutieuse et détaillée des risques et des avantages associés au *réhébergement*, à la *réarchitecture*/au changement de plate-forme ou au retrait de chaque application du portefeuille aidera à atténuer les risques globaux associés à la migration de l'ensemble des applications.

En particulier, il s'avérera important de comparer les coûts au niveau de chaque service avec le coût total pour LAE, afin d'évaluer le TCO de tout matériel dont LAE aurait besoin pour maintenir les applications sur site.

Ce TCO sera alors à comparer avec le coût d'hébergement par un fournisseur de solution cloud sur la même période, auquel il faudra également tenir compte du compte d'installation et de migration de l'ancien SI vers le nouveau.

## IV.C. Programme

La justification de cette migration a été abordée au sein du *Document de Définition d'Architecture*.

Associée à cette justification, les parties prenantes devront également considérer la modification éventuelle des cas d'utilisation pour cette migration d'applications.

En effet, pour l'heure, LAE cherche à déplacer des applications vers le cloud parce qu'elle souhaite :

- rendre cohérentes les données hébergées par son SI ;
- bénéficier de la flexibilité et de l'évolutivité du cloud ;
- rendre la structure de coûts prévisible à l'utilisation.

Cependant, de par sa politique d'Amélioration Continue relative à la Satisfaction de ses clients, LAE recherche également des environnements propices à l'innovation.

Qu'il s'agisse d'accéder aux processeurs hautes performances nécessaires pour alimenter des algorithmes d'apprentissage en profondeur, ou des applications conteneurisées qui permettent au responsable informatique de LAE d'améliorer rapidement les expériences numériques des clients en mettant en œuvre des changements à grande vitesse, les technologies cloud permettront à l'entreprise d'expérimenter, de tester de nouvelles idées , et d'"échouer plus vite".

Dans de nombreux cas, les technologies compatibles avec le cloud, comme la conteneurisation, permettront d'offrir aux utilisateurs finaux de meilleures expériences que les machines locales remplacées.

## IV.D. Développement de système / Ingénierie

D'une manière générale, le processus de planification de la migration des applications peut être divisé en trois étapes :

- **phase 1** : identification et évaluation des candidatures ;
- **phase 2** : évaluation du coût total de possession (TCO) ;
- **phase 3** : évaluer le risque global et la durée du projet.

Dans chacune d'entre elles, il sera indispensable d'estimer les coûts de toutes les options potentielles, y compris le choix de conserver certaines charges de travail sur site.

#### **IV.D.1. Phase 1 : Identification et évaluation des candidatures**

Dans cette phase de découverte initiale, et tel qu'il en a été question au sein de *Document de Définition d'Architecture*, LAE devra commencer par s'assurer de disposer d'un catalogue complet de toutes les applications de son portefeuille.

Une fois ce travail de *contrôle* réalisé, toutes les applications en vigueur seront classées et triées selon leur degré de criticité et leur valeur stratégique pour LAE.

Il sera alors nécessaire de définir précisément l'attendu de chacune de ses solutions après leur migration vers le cloud.

Pour chacune des applications en question, les collaborateurs de LAE devront s'efforcer de comprendre la valeur de chaque application en fonction des caractéristiques suivantes :

- impact sur l'entreprise ;
- capacité à répondre aux besoins commerciaux cruciaux ;
- actualité et importance des données ;
- taille, complexité et gérabilité ;
- coût de maintenance et de développement ;
- augmentation de la valeur de la migration vers le cloud.

Une fois ces analyses effectuées, LAE devra alors réaliser une évaluation de l'affinité cloud pour chaque application envisagée pour cette migration. Au cours de ce processus, il sera alors nécessaire de déterminer quelles applications sont prêtes à fonctionner telles quelles et lesquelles nécessiteront des modifications importantes avant de pouvoir être prêtes pour le cloud.

Lors de cette étape, il est conseillé d'utiliser des outils de découverte des dépendances des applications pour aider à déterminer la faisabilité de la migration d'une charge de travail particulière en dehors de son environnement actuel.

#### **IV.D.2. Phase 2 : évaluation du coût total de possession (TCO)**

Déterminer le coût total d'un projet de migration vers le cloud peut s'avérer être une opération complexe.

Dans un premier temps, il faudra lister les scénarii hypothétiques qui pourraient se dérouler lors de la migration, aussi en bien qu'en mal, puis comparer ces scénarii pour conserver les applications et l'infrastructure sur site avec ceux associés à leur migration vers le cloud.

Durant cette étape LAE devra calculer les coûts d'achat, d'exploitation et de maintenance du matériel à sur site dans l'un ou l'autre scénario, ainsi que les coûts des logiciels de licence.

Dans un deuxième, les collaborateurs de LAE devront comparer la facture mensuelle obtenue auprès du fournisseur de cloud, dans l'un ou l'autre scénario, avec les coûts de la migration elle-même comprenant :

- les coûts de test de la nouvelle infrastructure
- les coûts de formation des employés relative à l'utilisation des logiciels mis à jour ;
- les coûts de maintenance des applications héritées qui restent sur site.

### IV.D.3. Phase 3 : évaluer le risque global et la durée du projet

Dans cette phase finale de la planification de la migration, les collaborateurs de LAE devront établir un calendrier pour le projet et identifier les risques ou les obstacles particulièrement susceptibles de rencontrer.

Ces risques devront considérer la migration d'applications héritées vers le cloud. En effet, de manière générale, plus l'application est ancienne, plus il est difficile (et donc potentiellement moins intéressant) de migrer vers le cloud.

Les logiciels obsolètes sont problématiques à bien des égards :

- leur maintenance est coûteuse ;
- ils peuvent poser des problèmes de sécurité s'ils ne sont plus corrigés ;
- ils ont tendance à mal fonctionner dans les environnements informatiques modernes.

Les collaborateurs de LAE devront être particulièrement minutieux dans leur évaluation des applications héritées avant de décider de les migrer.

Il sera donc nécessaire de porter une attention toute particulière à cette phase critique.

Aussi, pour donner des indications quant à sa réalisation, le *Document de Définition d'Architecture* met en évidence, au sein de §IX *Evaluation de l'impact*, trois critères principaux :

- une preuve d'efficacité ;
- un outil de pilotage ;
- un dispositif de co-construction au service du partenariat.

## V.Éligibilité à la migration

Lors de l'évaluation de la viabilité et de la priorité de chaque application en tant que candidat à la migration, LAE devra se poser l'ensemble des questions suivantes :

- **Complexité**
  - Où cette application a-t-elle été développée ?
  - S'il est interne, le développeur travaille-t-il toujours chez LAE ?
  - La documentation de l'application est-elle facilement disponible ?
  - Quel est l'âge de l'application ?
  - Depuis combien de temps est-elle utilisée ?
  - Combien d'autres applications ou flux de travail au sein de LAE dépendent de celui-ci d'une manière ou d'une autre ?
- **Criticité**
  - Combien de collaborateurs dépendent de cette application au quotidien et sur une base hebdomadaire ?
  - Combien de temps d'arrêt pourraient-ils tolérer avant que les opérations commerciales ne soient interrompues ?
  - L'application est-elle utilisée en production, en développement, en test ou les trois ?
  - L'application est-elle gérée par une équipe informatique interne ou un fournisseur externe ?
  - Existe-t-il d'autres applications avec des exigences de disponibilité/d'indisponibilité qui doivent être synchronisées avec celles de cette application ?
- **Conformité**
  - À quelles exigences réglementaires votre organisation doit-elle se conformer ?
- **Disponibilité**
  - À quelles normes de disponibilité cette application doit-elle se conformer ?
  - Est-il soumis à un accord de niveau de service (SLA) stipulant une disponibilité de 99,99 %, par exemple ?



