

# Rapport d'audit général



Site de Boston

Site de Londres

Site de Francfort

Site de Hong Kong

Site de New Delhi

Site de Dubaï

Période du 11 avril 2022 au 13 mai 2022

## Auteur(s) et contributeur(s)

Nom & Coordonnées	Qualité & Rôle	Société
ATTARD Gérald	Architecte Logiciel, en qualité d'auditeur principal	A++ Logistics

## Validation

N° version	Nom & Qualité	Date & Signature	Commentaires & Réserves

## Tableau des abréviations

Abr.	Sémantique
AWS	Amazon Web Services (trad. <i>‘services web Amazon’</i> )
COTS	Commercial Off-The-Shell (trad. <i>‘produit sur étagère’</i> ou <i>‘produit informatique standard’</i> )
CSS	Cascading Style Sheet (trad. <i>‘feuille de style en cascade’</i> )
EC2	Amazon Elastic Compute Cloud
ERP	Entreprise Ressource Planning (trad. <i>‘progiciel de gestion intégré’</i> )
IaaS	Infrastructure-As-A-Service (trad. <i>‘infrastructure en tant que service’</i> )
HPC	High Performance Calcul (trad. <i>‘calcul à haute performance’</i> )
IHM	Interface Homme Machine
IDE	Integrated Development Environment (trad. <i>‘environnement de développement intégré’</i> )
SGBDR	Système de Gestion de Base de Données Relationnel
MVC	Modèle Vue Contrôleur
MVP	Modèle Vue Présentation
MVVM	Modèle Vue Vue-Modèle
PaaS	Platform-As-A-Service (trad. <i>‘plateforme en tant que service’</i> )
PTMS	Package Transportation Management System (trad. <i>‘système de gestion de colis’</i> )
RFID	Radio Frequency Identification (trad. <i>‘identification par radio-fréquence’</i> )
SAP	System, Applications and Products for data processing (trad. <i>‘système, applications et produits pour le traitement de données’</i> )
SaaS	Software-As-A-Service (trad. <i>‘logiciel en tant que service’</i> )
SGBDR	Système de Gestion de Base de Données Relationnel
SI	Système d’Information
SOA	Service-Oriented Architecture (trad. <i>‘architecture orientée services’</i> )
TMS	Transportation Management System (trad. <i>‘système des gestion de transport’</i> )
WMS	Warehouse Management System (trad. <i>‘système de gestion d’entrepôt’</i> )

Date d'émission du rapport : 11 Mai 2022

Société A++ Logitics,  
sites de :

- Boston,
- Londres,
- Francfort,
- New Delhi,
- Hong Kong,
- Dubaï.

Rapport établi à Boston, Massachusetts, États-Unis

Nous présentons les résultats de notre audit général informatique couvrant la période du 11 Avril 2022 au 13 Mai 2022. Le rapport comprend nos conclusions sur les points relatifs au remplacement de la suite logicielle logistique existante. La solution de remplacement choisie pour atteindre cet objectif devra être synonyme de gain de performance en homogénéisant les processus des traitements et de formalisation des données, entre les différents bureaux régionaux de la société. L'audit a été effectué conformément aux normes et lignes directrices de l'ISACA en matière d'audit et d'assurance des systèmes informatiques. Nous estimons que les éléments probants recueillis fournissent une base raisonnable à nos conclusions et observations concernant les objectifs de l'audit.

Gérald ATTARD  
Architecte Logiciel  
en qualité de responsable de l'audit

A handwritten signature in black ink, appearing to be 'G. Attard', written in a cursive style.

## Table des matières

Introduction.....	6
Portée de l’audit.....	7
Objectifs de l’audit.....	8
Méthodologie.....	9
Pré-audit / Planification de l'audit.....	9
Réalisation de l'audit.....	9
Constats de l’audit.....	10
Architecture du système.....	10
SITE DE BOSTON.....	10
SITE DE DUBAÏ.....	11
SITE DE FRANCFORT et LONDRES.....	12
SITE DE HONG KONG.....	13
SITE DE NEW DELHI.....	14
Synthèse des architectures.....	15
Hypothèses de performance / analyse.....	16
Évaluation de la capacité du système.....	17
Limites et Risques.....	19
Limites.....	19
Risques.....	20
Risques humains.....	20
Risques techniques.....	22
Matrice des risques.....	23
Cartographie des risques.....	24
Exigences fonctionnelles et techniques.....	25
Contraintes fonctionnelles.....	26
Solution proposée.....	27
Solution propriétaire développée sur mesure.....	27
.Net.....	28
Backend Node.js (JavaScript) – Frontend Angular (TypeScript).....	29
Backend Java – Frontend Javascript.....	30
Conclusion partielle.....	31
Solutions AWS.....	32
AWS EC2.....	33
AWS Cloud Control API.....	34
AWS Cloud Development Kit.....	35
AWS CodeDeploy.....	36
AWS Pricing.....	37
Conclusion partielle.....	39
Solution COTS : LogiStax Suite 20.....	40
Conclusion partielle.....	42
Conclusions et opinions de l’audit.....	43
Annexes.....	45
Tableau de sourcing.....	45



## Introduction

De par sa reconnaissance au niveau mondiale, la société A++ Logistics a su s'imposer depuis plusieurs années dans le domaine du transport de marchandises.

Cette société, de renom international, travaille avec de grandes enseignes commerciales afin d'assurer toutes les opérations logistiques de transport sécurisé et de livraison planifiée, faisant suite aux démarches commerciales relatives à un produit.

Ce contexte international est concrétisé par une couverture de zones géographiques dévolues à 6 bureaux régionaux, à savoir :

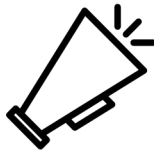
- le bureau de Boston, le siège social, couvrant la zone des Amériques ;
- le bureau de Dubaï couvrant la zone du Moyen-Orient ;
- le bureau de Francfort couvrant la zone d'Europe centrale et méridionale ;
- le bureau de Hong Kong couvrant la zone d'Asie orientale ;
- le bureau de Londres couvrant la zone d'Europe septentrionale et scandinave ;
- le bureau de New Delhi couvrant la zone d'Asie occidentale.

Ces bureaux régionaux utilisent des systèmes d'information hétérogènes, relatifs à 4 domaines principaux :

- un système de reporting ;
- un système de gestion des transports (TMS) ;
- un système de gestion des colis (PTMS) ;
- un système de gestion d'entrepôt (WMS).

Ainsi, l'objectif de cet audit sera d'établir une synthèse de l'étude de l'existant, fonctionnelle, technique et financière, afin de proposer une solution d'homogénéisation du système d'information.

Cette solution répondra alors au *Besoin d'évaluation de la faisabilité d'un projet d'installation d'un nouveau système pour gérer les opérations logistiques dans tous les bureaux du monde.*



## Portée de l'audit

Selon la méthodologie de conduite d'une mission d'audit interne de l'Institut Français de l'Audit et du Contrôle Interne (IFACI), nous avons effectué un audit interne de synthèse de l'existant au sein d'A++ Logistics, pour la période du 11 Avril 2022 au 13 Mai 2022.

La portée de notre audit consistait en une évaluation du système d'information actuel composé de 3 solutions logicielles complémentaires. Ces dernières, assurant les fonctions de reporting, de gestion des transports et des entrepôts, étaient le socle commun des différentes architectures logicielles relevées auprès des bureaux régionaux.

En outre, bien qu'un socle fonctionnel commun d'architecture logicielle ait pu être constaté au sein de chaque bureau, leur application logicielle associée diffère d'un bureau à l'autre.

Cet audit restituera les analyses fonctionnelles, techniques et financières propres à chaque bureau, dans l'optique de proposer une solution homogène répondant aux mêmes besoins fonctionnels, à moindre coût.

De plus, l'audit a été effectué conformément aux normes et lignes directrices de l'ISACA en matière d'audit et d'assurance des systèmes d'information. Il s'est également basé sur les normes de qualification et de fonctionnement du cadre de référence de l'audit interne dans l'Administration de l'État, issu du Comité d'Harmonisation de l'Audit Interne (CHAI). Ces normes exigent que l'audit soit planifié et réalisé de manière à obtenir des éléments probants suffisants, pertinents et valides pour fournir une base raisonnable aux conclusions et observations de l'audit (le cas échéant).

Ainsi, le champs d'action de cet audit s'est limité à la mise en exergue des différents contextes fonctionnels et techniques entre les bureaux régionaux, afin d'en factoriser un ou des processus communs. Puis en partant de ce(s) facteur(s) commun(s), la complétion des besoins fonctionnels devra être corrélée avec les besoins fonctionnels et les contraintes techniques et financières déjà en vigueur au sein de l'entreprise.



## Objectifs de l'audit

En prenant comme hypothèse de départ que le système d'information global doit remplir des fonctions de reporting, de gestion des transports et d'entrepôts, l'objectif de notre audit est de déterminer si ces 3 fonctionnalités sont, dans un premier temps, bien présentes au sein de chaque bureau régional.

Suite à la confirmation ou l'infirmerie de la présence de ces fonctions, une analyse technico-fonctionnelle a été réalisée pour identifier les solutions logicielles utilisées pour chacune d'elles.

Ainsi, ce document présentera une approche « systémique et méthodique » afin de :

- présenter les faits au travers d'une synthèse de l'existant ;
- proposer des solutions de remplacement du système d'information actuel, pertinentes et cohérentes, en toute objectivité ;
- aider à la prise de décision quant au choix de la solution.





## Pré-audit / Planification de l'audit

Pour déterminer la portée et les objectifs de l'audit, nous avons effectué des étapes préalables à l'audit, notamment en déterminant et en enregistrant ce qui constitue la mission, les opérations commerciales pertinentes et la technologie de soutien d'A++ Logistics. Nous avons cerné les obligations opérationnelles, juridiques et réglementaires de l'organisation auditée, en examinant la documentation pertinente communiquée par le bureau audité.

Nous avons alors défini des objectifs d'audit par rapport aux objectifs fonctionnels et opérationnels identifiés, ce qui a permis d'en extrapoler la portée de l'audit.

Suite à ces étapes, nous avons effectué une évaluation des risques, en identifiant ceux-ci au sein d'une matrice des risques. Nous avons alors représenté cette dernière au sein d'une cartographie des risques qui nous a permis d'estimer les actions à mettre en place pour réduire les risques en question.

Enfin, en corrélant toutes ces informations nous avons préconisé une solution répondant à chacun des critères énoncés supra, et répondant aux besoins fonctionnels, tout en conservant une cohérence technique et un souci d'optimisation des coûts.

## Réalisation de l'audit

*[La méthodologie de l'audit doit fournir une explication détaillée de la manière dont l'audit a été réalisé pour chaque objectif énoncé. La méthodologie devrait préciser la nature et l'étendue du travail d'audit, les sources des critères d'audit, si l'on a consulté le travail d'autres professionnels, le type d'analyse effectuée et le fondement des conclusions tirées.]*

Notre audit a été mené conformément aux normes d'audit et de certification des SI et aux directives de vérification et d'assurance des SI publiées par ISACA et *[insérer toute norme supplémentaire pertinente pour l'audit]*, et les pratiques généralement acceptées dans le secteur. Les critères utilisés lors de l'audit comprenaient les politiques et procédures de gestion, *[insérer l'identification des références légales ou réglementaires]* et les directives de contrôle de gestion décrites dans le référentiel COBIT® 5, telles que publiées par ISACA.



## Constats de l'audit

Le but de cette section est de fournir une explication détaillée des observations de l'audit, des recommandations et des éléments de réflexion pour la direction.

### Architecture du système

L'analyse d'architecture logicielle de chaque bureau régional a permis d'avoir une vision d'ensemble nécessaire à la réalisation de cet audit et est détaillée dans les sections qui suivent. Il y est décrit de manière symbolique et schématique les différents éléments d'un ou plusieurs systèmes informatiques, leurs interactions et interrelations.

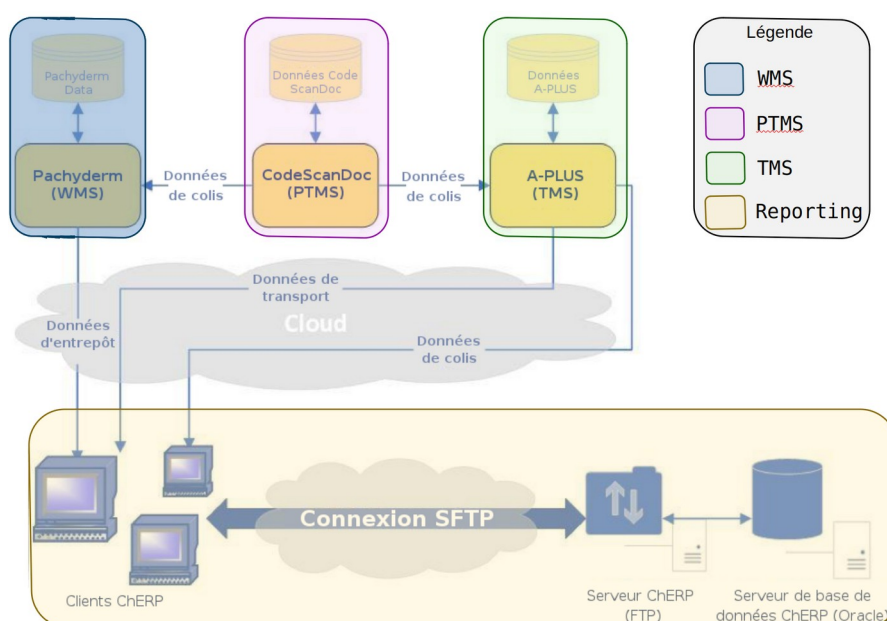
De plus, les modèles d'architecture représentées sont l'image d'un existant communiqué et non constatée par un déplacement sur site. Pour parfaire cet audit, il sera nécessaire d'effectuer des relevés in situ, effectués par un auditeur impartial, externe au bureau régional audité.

### SITE DE BOSTON

Le bureau de Boston a servi de référence quant à l'architecture logicielle choisie pour les autres bureaux régionaux.

Au sein de ce site, toutes les fonctionnalités sont représentées par l'utilisation d'une solution logicielle dédiée à celles-ci, telle que le montre le schéma ci-dessous; chaque solution logicielle possédant sa propre base de données.

Il est à noter que chaque solution réalise ses propres extractions de rapport et les communique aux clients ChERP. Ces derniers les transfèrent ensuite au serveur ChERP pour intégration au sein de la base de données Oracle par FTP.

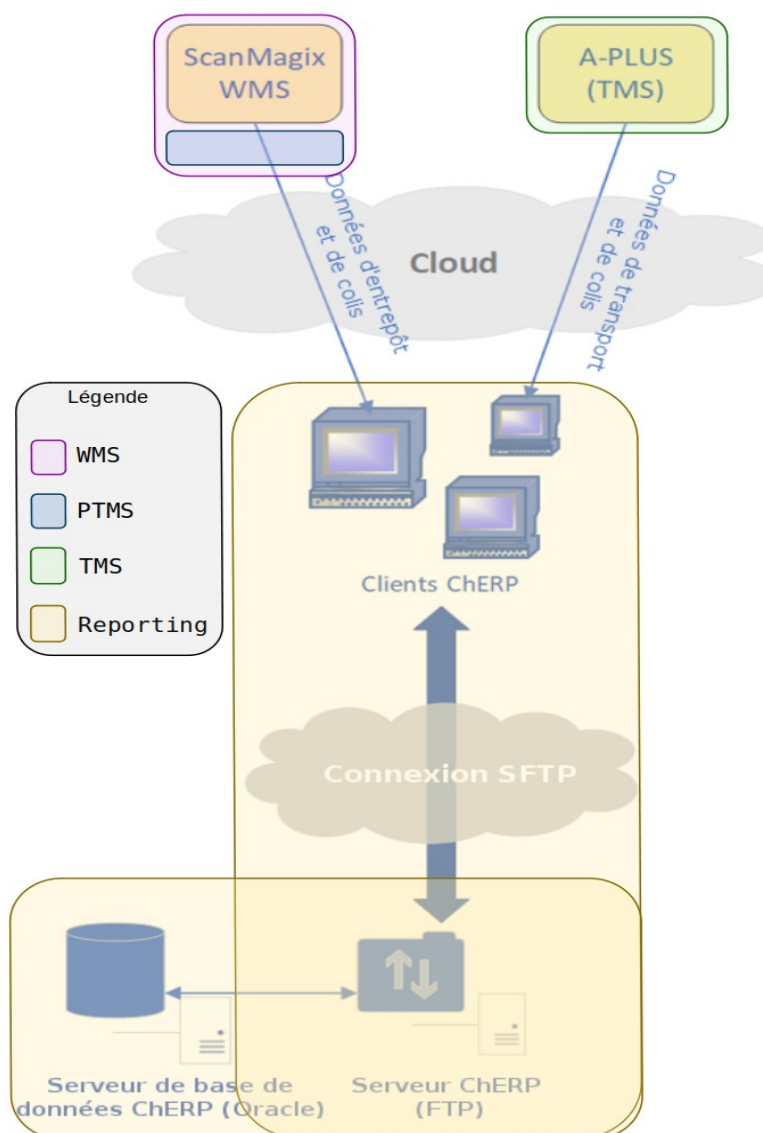


## SITE DE DUBAÏ

Comparativement au bureau de Boston, ce site utilise 2 autres logiciels qui lui sont propres et dont l'intégration et les connexions avec ChERP ont été réalisées suite au développement d'API spécifiques en C++.

Ainsi, la fonction de TMS est assurée par le logiciel A-PLUS et la fonction de WMS est assurée par le logiciel ScanMagix. Il est à noter que ce dernier fait également office de PTMS.

Chacun de ces logiciels génèrent ses propres rapports qui sont envoyés directement aux clients ChERP, puis transférés au serveur ChERP, selon le même schéma expliqué pour le bureau de Boston.



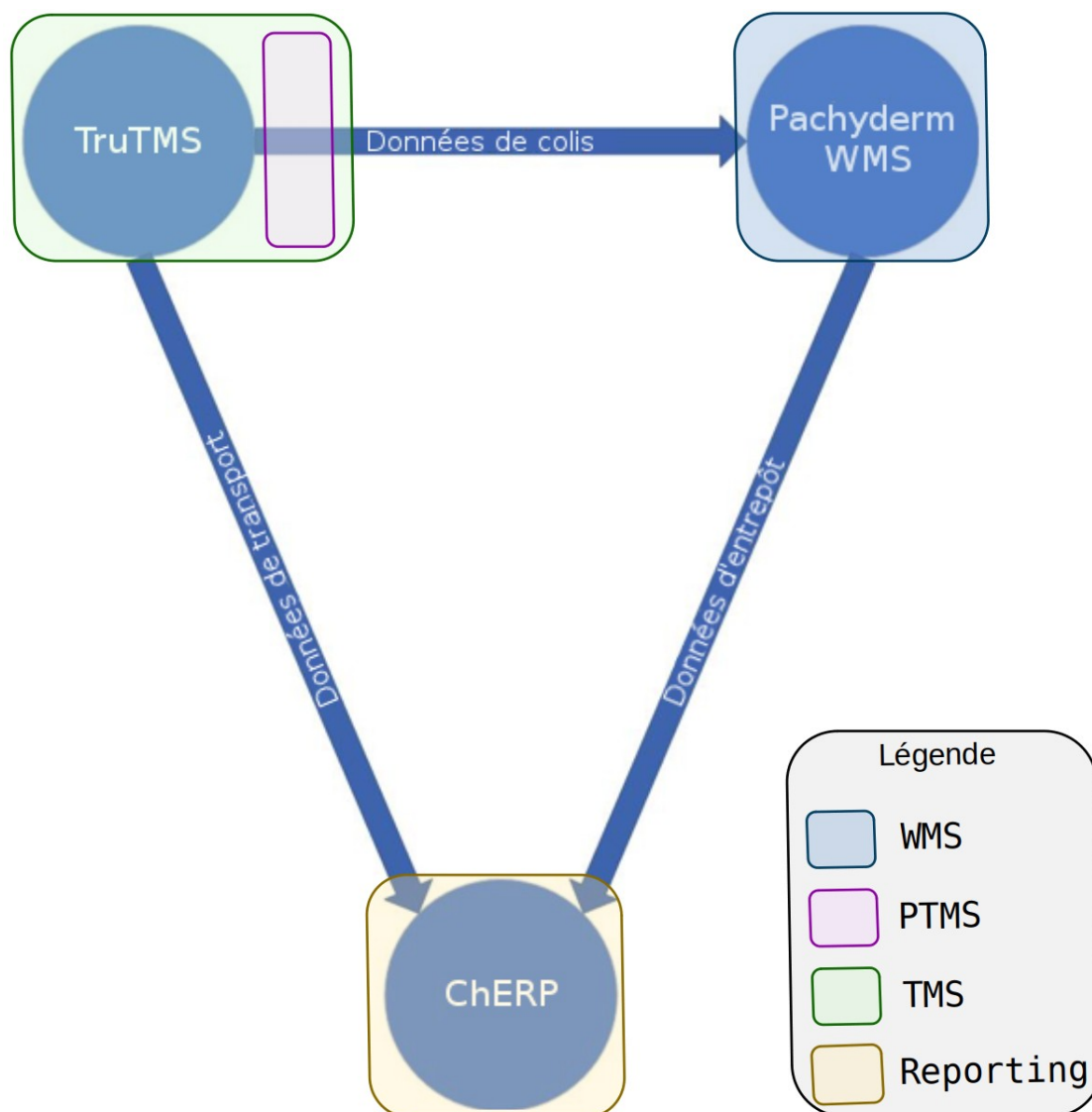
## SITE DE FRANCFORT et LONDRES

Comparativement au bureau de Boston, ces sites utilisent un autre logiciel qui leur est propre et dont l'intégration et les connexions avec ChERP ont été réalisées suite à l'installation d'un plugin.

Ce dernier permet de transmettre les données parcellaires liées à la gestion d'entrepôt pour effectuer les traitements et l'affichage relatif aux données.

Ainsi, la fonction de TMS est assurée par le logiciel TruTMS. Il est à noter que ce dernier fait également office de PTMS et communique les données relatives à cette fonction à Pachyderm WMS.

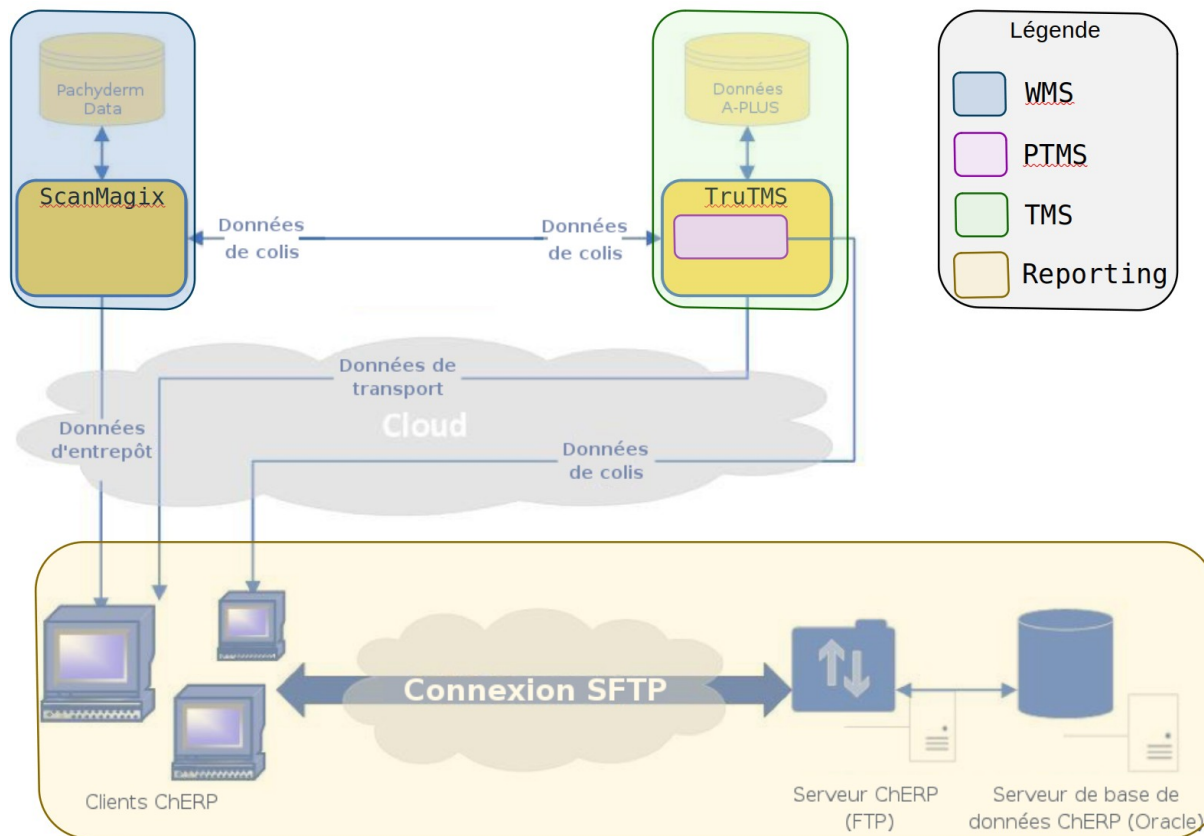
Les rapports sont alors extraits indépendamment pour chaque fonction et seules les données relatives au TMS et au WMS sont envoyées à ChERP ; il n'y a donc pas de rapport relatif à la fonction PTMS envoyé à ChERP.



## SITE DE HONG KONG

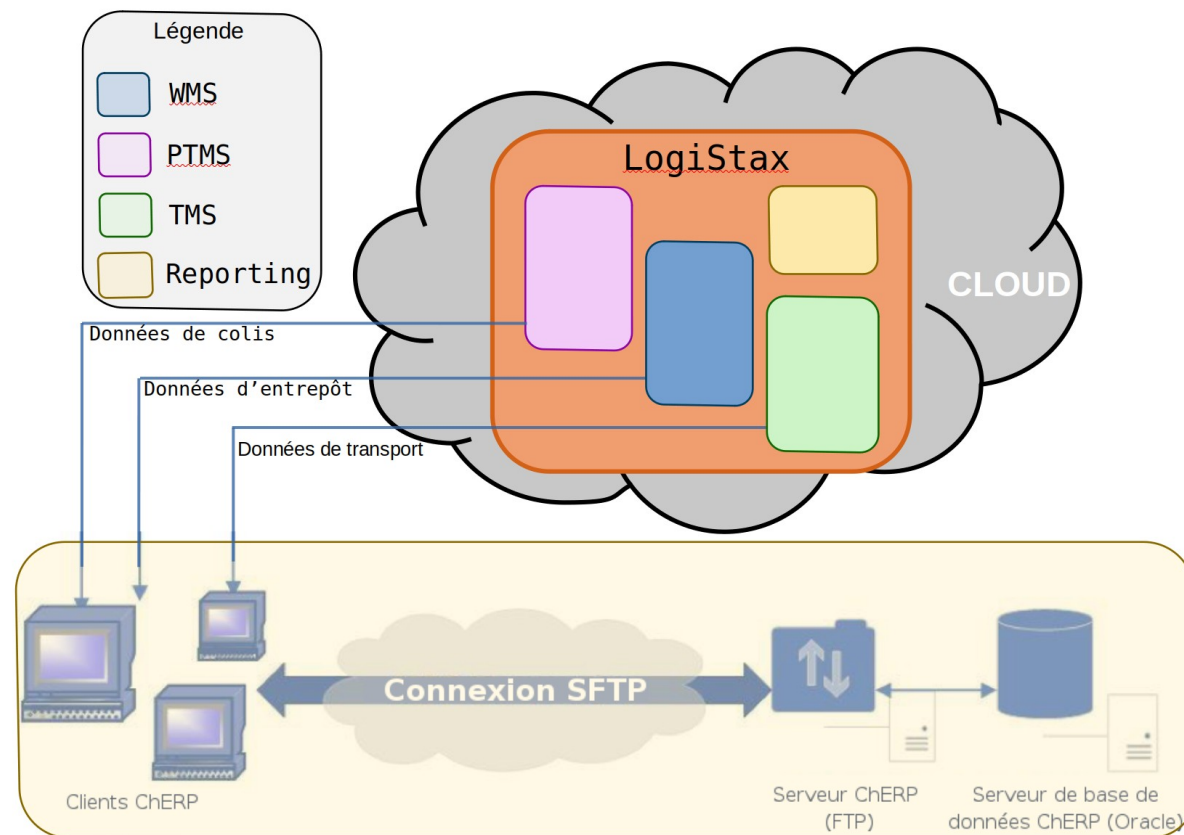
Comparativement au bureau de Boston, ce site utilise 2 autres logiciels qui lui sont propres.

Ainsi, les fonctions de TMS et PTMS sont assurées par le logiciel TruTMS et celle de WMS est assurée par le logiciel ScanMagix. Ces solutions logicielles possèdent leur propre base de données et génèrent leurs propres rapports qui sont envoyés directement aux clients ChERP, puis transférés au serveur ChERP, selon le même schéma expliqué pour le bureau de Boston.



## SITE DE NEW DELHI

Comparativement au bureau de Boston, ce site utilise une suite logicielle qui lui est propre, comprenant les fonctionnalités de TMS, PTMS, WMS et même de reporting, indépendamment de ChERP. Néanmoins, pour des raisons de cohérence de la politique générale de reporting, cette fonction n'est pas utilisée et les rapports sont transmis à ChERP selon les modalités du bureau de Boston.



## Synthèse des architectures

Cet audit a référencé les logiciels utilisés au sein des différents bureaux, ainsi que leur fonctionnalité associée ; vous retrouverez dans le tableau ci-dessous la synthèse de l'existant :

<div> <div>Logiciel</div> <div>Domaine</div> </div>		Pachyderm	ScanMagix	A-PLUS	TruTMS	CodeScanDoc	ChERP	LogiStax
Architecture du système	site de Boston	X		X		X	X	
	site de Dubaï		X	X			X	
	site de Francfort	X			X		X	
	site de Hong Kong		X		X		X	
	site de Londres	X			X		X	
	site de New Delhi							X
	TMS			X	X			X
fonctionnalité	PTMS		X		X	X		X
	WMS	X	X					X
	Reporting						X	X
	Base de données dédiée	X		X	X	X	X	X

Le lecteur est invité à porter une attention toute particulière à la dernière ligne du tableau ci-dessus, à savoir, la présence au sein de chaque logiciel d'une base de données dédiée.

Ainsi, la présence d'une base de données dédiée à un logiciel ajoute un élément de complexité dont il faut tenir compte quant à la prise en compte globale de l'environnement technique.

En outre, une base de données dédiée peut être perçue comme un avantage pour une application unique, par contre elle peut représenter un degré de complexité au sein d'un environnement technique composé de plusieurs applications logicielles, ayant chacune leur base de données dédiée.

## Hypothèses de performance / analyse

L'ensemble des architectures logicielles des bureaux régionaux, décrites dans le paragraphe ci-dessus, est basé sur celle du bureau de Boston structurée en 4 composants fonctionnels :

- un système d'ERP, utilisé pour le reporting;
- un système de TMS, utilisé pour gérer les opérations de transport;
- un système de PTMS, utilisé pour collecter les informations relatives aux propriétés des colis;
- un système de WMS, utilisé pour gérer les opérations d'un entrepôt de stockage.

En terme de ressenti d'utilisation, cette précédente architecture fonctionne de manière lente et, malgré une continuité de service assez régulière, cette dernière fait néanmoins l'objet de coupures de service.

Ainsi, la lenteur de ce système d'information peut s'expliquer par plusieurs causes :

- la complexité des informations échangées entre composants hétérogènes;
- l'inadaptation des modules de traitement aux besoins ou aux informations traitées;
- la déstructuration de fonctions élémentaires par la répartition des responsabilités fonctionnelles, au travers d'outils dédiés, peut être perçue comme un avantage pour des systèmes simples. Néanmoins, cette répartition atomique engendre encore plus de complexité au sein de systèmes complexes ; ainsi, une homogénéisation des composants peut alors pallier à cet état de fait.

En outre, le fait que les différents traitements de l'information soient réalisés de façon décentralisée, par rapport à l'affichage des résultats sur les clients, peuvent également engendrer un ressenti de lenteur.

D'après les remarques présentées ci-dessus, 2 actions seraient profitables au système d'information:

- l'amélioration des traitements de chaque composant technique en cherchant une meilleure compatibilité entre les différents composants;
- l'amélioration des processus de communication entre ces composants, dont, notamment, l'amélioration des temps de réponse par les clients.

Ce dernier point sera en mettre en exergue lors des processus de sauvegarde des données.

De façon générale, l'homogénéisation de l'ensemble des composants logiciels améliorera les temps de traitement, d'accès et de réponse.



## Évaluation de la capacité du système

Le tableau ci-dessous recense les capacités de tous les logiciels rencontrés au sein de chaque bureau régional. Vous trouverez le lexique de chaque critère en dessus de ce tableau.

Critère de Qualité	Logiciel						
	Pachyderm	ScanMagix	A-PLUS	TruTMS	CodeScanDoc	ChERP	LogiStax
Autonomie	?	X	?	X	?	-	X
Compatibilité	?	?	?	?	?	?	X
Composabilité	X	X	X	?	X	X	X
Efficacité	?	?	?	?	?	-	X
Extensibilité	X	X	?	?	?	?	X
Fiabilité	?	?	?	?	?	?	?
Intégrité	?	?	?	?	?	?	?
Interopérabilité extrinsèque	X	X	X	X	X	X	X
Interopérabilité intrinsèque	X	X	X	X	X	X	X
Maintenabilité	?	?	?	?	?	?	X
Portabilité	?	?	?	?	?	?	?
Réutilisabilité	?	?	?	?	?	X	X
Transparence	?	?	?	?	?	?	?
Validité	X	X	X	X	X	X	X
Vérifiabilité	?	?	?	?	?	-	?

Au sein de cet audit, l'évaluation des capacités des systèmes de chaque bureau régional a été effectué en prenant comme base les critères de qualité logicielle suivant :

- L'**autonomie** exprime la capacité de contrôle de son exécution, de ses données et de ses communications ;
- La **compatibilité** exprime la possibilité, pour un logiciel, de fonctionner correctement dans un environnement ancien (compatibilité descendante) ou plus récent (compatibilité ascendante) ;
- La **composabilité** exprime la capacité pour un logiciel de combiner des informations provenant de sources différentes ;
- L'**efficacité** exprime la capacité du logiciel à exploiter au mieux les ressources offertes par la ou les machines où le logiciel sera implanté ;
- L'**extensibilité** exprime la possibilité d'étendre simplement les fonctionnalités d'un logiciel sans compromettre son intégrité et sa fiabilité ;
- La **fiabilité** exprime la faculté du logiciel à gérer correctement ses propres erreurs de fonctionnement en cours d'exécution ;
- L'**intégrité** exprime la faculté du logiciel à protéger ses fonctions et ses données d'accès non autorisés ;
- L'**interopérabilité extrinsèque** exprime la capacité du logiciel à communiquer et à utiliser les ressources d'autres logiciels comme les documents créés par une certaine application ;
- L'**interopérabilité intrinsèque** exprime le degré de cohérence entre le fonctionnement des commandes et des modules à l'intérieur d'un système ou d'un logiciel ;
- La **maintenabilité** exprime la simplicité de correction et de modification du logiciel, et même, parfois, la possibilité de modification du logiciel en cours d'exécution ;
- La **portabilité** exprime la possibilité de compiler le code source et/ou d'exécuter le logiciel sur des plates-formes (machines, systèmes d'exploitation, environnements) différentes ;
- La **réutilisabilité** exprime la capacité de concevoir le logiciel avec des composants déjà conçus tout en permettant la réutilisation simple de ses propres composants pour le développement d'autres logiciels ;
- La **transparence** exprime la capacité pour un logiciel de masquer à l'utilisateur (humain ou machine) les détails inutiles à l'utilisation de ses fonctionnalités ;
- La **validité** exprime la conformité des fonctionnalités du logiciel avec celles décrites dans le cahier des charges ;
- La **vérifiabilité** exprime la simplicité de vérification de la validité.

Ainsi, chaque critère dispose au sein du tableau ci-dessus de la symbologie suivante :

- '**X**' critère validé ;
- '**?**' critère ne pouvant être apprécié avec les éléments à disposition ;
- '**-**' critère dont l'absence est avérée.

## Limites et Risques

Dans un premier, cet audit définira des objectifs de sécurité en présentant les limites fonctionnelles constatée au sein des bureaux régionaux présentés.

Dans un deuxième temps, une fois ces objectifs de sécurité déterminés, les risques pesant sur chacun de ces éléments pourront être estimés en fonction des **menaces**. Le niveau global de sécurité des systèmes d'information est défini par le niveau de sécurité du maillon le plus faible. Les précautions et contre-mesures doivent être envisagées en fonction des vulnérabilités propres au contexte auquel le système d'information est censé apporter service et appui ; Il faut pour cela estimer :

- la **gravité** des conséquences au cas où les risques se réaliseraient ;
- la **vraisemblance** des risques (ou leur *potentialité*, ou encore leur *probabilité d'occurrence*).

## Limites

La représentation actuelle du système d'information d'A++ Logistics présentent certaines limites fonctionnelles associées aux risques énumérés ci-dessus, à savoir :

- la **confidentialité** : dans un système maîtrisé, seules les personnes autorisées doivent avoir accès aux informations qui leur sont destinées (notions de droits et permissions) - tout accès indésirable doit être empêché. En l'état actuel des informations de sécurité, cet audit n'a pas assez d'élément pour se prononcer quant à la réalisation de cet objectif de sécurité.
- l'**authenticité** : dans un système maîtrisé, les utilisateurs doivent prouver leur identité par l'usage de code d'accès. En outre , il ne faut pas mélanger *identification* et *authentification* : dans le premier cas, l'utilisateur n'est reconnu que par son identifiant public, tandis que dans le deuxième cas, il doit fournir un mot de passe ou un élément que lui-seul connaît (secret). Mettre en correspondance un identifiant public avec un secret est le mécanisme permettant de garantir l'*authenticité* de l'identifiant. Cela permet de gérer les droits d'accès aux ressources concernées et maintenir la confiance dans les relations d'échange. Néanmoins, en l'état actuel des informations de sécurité, cet audit n'a pas assez d'élément pour se prononcer quant à la réalisation de cet objectif de sécurité.
- l'**intégrité** : dans un système maîtrisé, les données doivent être celles que l'on attend, et ne doivent pas être altérées de façon fortuite, illicite ou malveillante. En clair, les éléments considérés doivent être exacts et complets. Cet objectif de sécurité utilise généralement des méthodes de calcul de *checksum* ou de hachage. Néanmoins, en l'état actuel des informations de sécurité, cet audit n'a pas assez d'élément pour se prononcer quant à la réalisation de cet objectif de sécurité.
- la **disponibilité** : l'accès aux ressources du système d'information doit être permanent et sans faille durant les plages d'utilisation prévues. Les services et ressources sont accessibles rapidement et régulièrement. Ainsi, au sein du bureau de Boston, servant de base de référence d'architecture logiciel aux autres bureaux régionaux, il a été constaté un qualité de service dégradée due à des coupures de services. Cet état de fait implique que cet objectif de sécurité n'est pas abouti au sein du système d'information actuel d'A++ Logistics.
- la **traçabilité** (ou « *preuve* ») qui garantit les accès et tentatives d'accès aux éléments considérés en les traçant - ces traces sont conservées et exploitables ;
- la **non-répudiation** et l'**imputation**, qui implique qu'aucun utilisateur ne doit pouvoir contester les opérations qu'il a réalisées dans le cadre de ses actions autorisées, et qu'aucun tiers ne doit pouvoir s'attribuer les actions d'un autre utilisateur.

Ainsi, en terme de vue d'ensemble relatif au logiciel ChERP, il apparaît que celui-ci n'est utilisé que pour assurer la cohérence des rapports entre les bureaux régionaux ; ce logiciel spécifique n'a donc qu'une vocation de reporting, oblitérant ainsi ses autres fonctionnalités d'ERP.

Cette solution logicielle, conçue pour le secteur de l'habillement et non pas du transport, est installée indépendamment au sein de chaque bureau régional de manière décorrélée. Cela signifie qu'en l'absence de connexion entre les bureaux régionaux, chacun d'entre eux a sa propre façon d'organiser et d'interpréter les extractions d'information, ce qui peut rendre le reporting incohérent lors de l'import des données au format XML.

## Risques

Dans un premier temps, cet audit attire l'attention sur le tableau indiqué au §*Synthèse des architectures* ci-dessus recensant la présence du logiciel ChERP et de sa base de données dédiée, au sein de tous les bureaux régionaux ; c'est donc un élément commun important à considérer.

Ainsi, en terme de vue d'ensemble relatif à ce logiciel, il apparaît que ChERP n'est utilisé que pour assurer la cohérence des rapports entre les bureaux régionaux ; ce logiciel n'a donc qu'une vocation de reporting, oblitérant ainsi ses autres fonctionnalités d'ERP.

Cette solution logicielle, conçue pour le secteur de l'habillement et non pas du transport, est installée indépendamment au sein de chaque bureau régional de manière décorrélée. Cela signifie qu'en l'absence de connexion entre les bureaux régionaux, chacun d'entre eux a sa propre façon d'organiser et d'interpréter les extractions d'information, ce qui peut rendre le reporting incohérent lors de l'import des données au format XML.

### *Risques humains*

En outre, ces opérations d'exportation/importation peuvent potentiellement être réalisées de façon manuelle, ce qui inclut ici les **risques humains**, à savoir :

- l'**abus de droit** qui caractérise le comportement d'un utilisateur, bénéficiaire de privilèges systèmes et/ou applicatifs, à les utiliser pour des usages excessifs, pouvant conduire à la malveillance ;
- l'**accès illégitime** d'une personne se faisant passer occasionnellement pour une autre en usurpant son identité ;
- le **chantage** qui est une menace exercée vis-à-vis d'une personne privée ou d'une organisation en vue d'extorquer une information « sensible » ;
- le **déguisement** désignant le fait qu'une personne se fait passer pour une autre de façon durable et répétée en usurpant son identité, ses privilèges ou les droits d'une personne visée ;
- le **détournement de mot de passe** tel qu'un administrateur système ou réseau peut modifier les mots de passe d'administration lui permettant de prendre le contrôle d'un système ou d'un réseau ;

- l'**écoute** qui consiste à se placer sur un réseau informatique ou de télécommunication pour collecter et analyser les informations ou les trames qui y circulent ;
- l'**espionnage**, notamment industriel, qui emploie les mêmes moyens, ainsi que bien d'autres (tel que l'influence), pour obtenir des informations sur des activités concurrentes, procédés de fabrication, projets en cours, futurs produits, politique de prix, clients et prospects, etc ;
- l'**inconscience** et l'**ignorance** de nombreux utilisateurs d'outils informatiques inconscients ou ignorants des risques qu'ils encourent vis à vis des systèmes qu'ils utilisent, et qui introduisent souvent des programmes malveillants sans le savoir - des manipulations inconsidérées sont aussi courantes ;
- l'**ingénierie sociale** (*social engineering* en anglais) qui est une méthode pour obtenir d'une personne des informations confidentielles, que l'on n'est pas normalement autorisé à obtenir, en vue de les exploiter à d'autres fins (publicitaires par exemple). Elle consiste à se faire passer pour quelqu'un que l'on n'est pas (en général un administrateur) et de demander des informations personnelles (nom de connexion, mot de passe, données confidentielles, etc.) en inventant un quelconque prétexte (problème dans le réseau, modification de celui-ci, heure tardive, etc.). Elle peut se faire soit au moyen d'une simple communication téléphonique, soit par email, soit en se déplaçant directement sur place ;
- L'**interception** qui est un accès avec modification des informations transmises sur les voies de communication avec l'intention de détruire les messages, de les modifier, d'insérer des nouveaux messages, de provoquer un décalage dans le temps ou la rupture dans la diffusion des messages ;
- la **maladresse**, comme en toute activité, les humains commettent des erreurs et il leur arrive donc, plus ou moins fréquemment, d'exécuter un traitement non souhaité, d'effacer involontairement des données ou des programmes, etc ;
- la **malveillance**, il serait quasiment inconcevable aujourd'hui de prétexter l'ignorance des risques sus-cités, tant les médias ont pu parler des différents problèmes de virus et de vers ces dernières années (même s'ils ont tendance, en vulgarisant, à se tromper sur les causes et les problèmes). Ainsi, certains utilisateurs, pour des raisons très diverses, peuvent volontairement mettre en péril le système d'information, en y introduisant en connaissance de cause des virus (en connectant par exemple un ordinateur portable sur un réseau d'entreprise), ou en introduisant volontairement de mauvaises informations dans une base de données. De même il est relativement aisé pour un informaticien d'ajouter délibérément des fonctions cachées lui permettant, directement ou avec l'aide de complices, de détourner à son profit de l'information ou de l'argent ;
- le **rejeu** qui est une variante du déguisement et qui permet à un attaquant de pénétrer un SI en envoyant une séquence de connexion d'un utilisateur légitime et enregistrée à son insu ;
- le **reniement** (ou **répudiation**) qui consiste pour une partie prenante à une transaction électronique à nier sa participation à tout ou partie de l'échange d'informations, ou à prétendre avoir reçu des informations différentes (message ou document) de ceux réputés avoir été réalisés dans le cadre du SI ;

- le **sabotage** qui vise la mise hors service d'un SI ou de l'une de ses composantes en portant atteinte à l'intégrité des données et surtout à la disponibilité des services.
- la **substitution** qui est présente sur des réseaux comportant des terminaux distants, et qui consiste à intercepter des messages de connexion-déconnexion peut permettre à un attaquant de continuer une session régulièrement ouverte, sans que le système ne remarque le changement d'utilisateur.

Les risques énumérés ci-dessus peuvent également être associés à des risques techniques.

### ***Risques techniques***

Les différents bureaux régionaux utilisent des infrastructures on-premise ou entièrement locales. De par la présence physique des matériels hébergeant les différentes solutions logicielles, celles-ci peuvent être synonyme de risque à plusieurs égards, à savoir :

- **Risques matériels** : s'il est le plus souvent possible de négliger la probabilité d'une erreur d'exécution par un processeur, la plupart des composants électroniques, produits en grandes séries, peuvent comporter des défauts et finissent un jour ou l'autre par tomber en panne. Certaines de ces pannes sont assez difficiles à déceler car intermittentes ou rares.
- **Risques logiciels** : la complexité croissante des systèmes d'exploitation et des programmes nécessite l'effort conjoint de nombreux programmeurs. Individuellement ou collectivement, ces erreurs sont pernicieuses car les meilleures méthodes de travail et les meilleurs outils de contrôle ou de test ne peuvent pas les éliminer en totalité. En effet, des failles permettant de prendre le contrôle total ou partiel d'un ordinateur sont régulièrement rendues publiques et répertoriées sur des sites spécialisés.
- **Risques liés à l'environnement** : les machines électroniques et les réseaux de communication sont sensibles aux variations de température ou d'humidité (tout particulièrement en cas d'incendie ou d'inondation) ainsi qu'aux champs électriques et magnétiques. Il n'est pas rare que des ordinateurs connaissent des pannes définitives ou intermittentes à cause de conditions climatiques inhabituelles ou par l'influence d'installations électriques notamment industrielles (et parfois celle des ordinateurs eux-mêmes).

Pour se prémunir des risques énumérés ci-dessus, cet audit préconise de recourir à des moyens triviaux bien qu'onéreux, en terme de réalisation, à savoir :

- **Redondance des matériels** : la probabilité ou la fréquence de pannes d'un équipement est représentée par un nombre très faible (compris entre 0 et 1, exprimé sous la forme  $10^{-n}$ ). En doublant ou en triplant (ou plus) un équipement, on divise le risque total par la probabilité de pannes simultanées. Le résultat est donc un nombre beaucoup plus faible ; autrement dit l'ensemble est beaucoup plus fiable (ce qui le plus souvent reporte le risque principal ailleurs).

- **Dispersion des sites** : Un accident (incendie, tempête, tremblement de terre, attentat, etc.) a très peu de chance de se produire simultanément en plusieurs endroits distants. Nous sommes ici en présence d'une force ET d'une faiblesse relative à la structure internationale d'A++ Logistics. Le système d'information actuel présente des avantages de délocalisation de l'information, néanmoins cette distance présente aussi des inconvénients en ralentissant l'affichage de l'information elle-même. Ainsi, suite à une étude de l'existant plus poussée, il serait souhaitable d'identifier de manière plus précise les informations et les processus à rapprocher des clients ChERP, de ceux à conserver à distance.
- **Programmes ou procédures de contrôle indépendants** : ils permettent bien souvent de déceler les anomalies avant qu'elles ne produisent des effets dévastateurs.

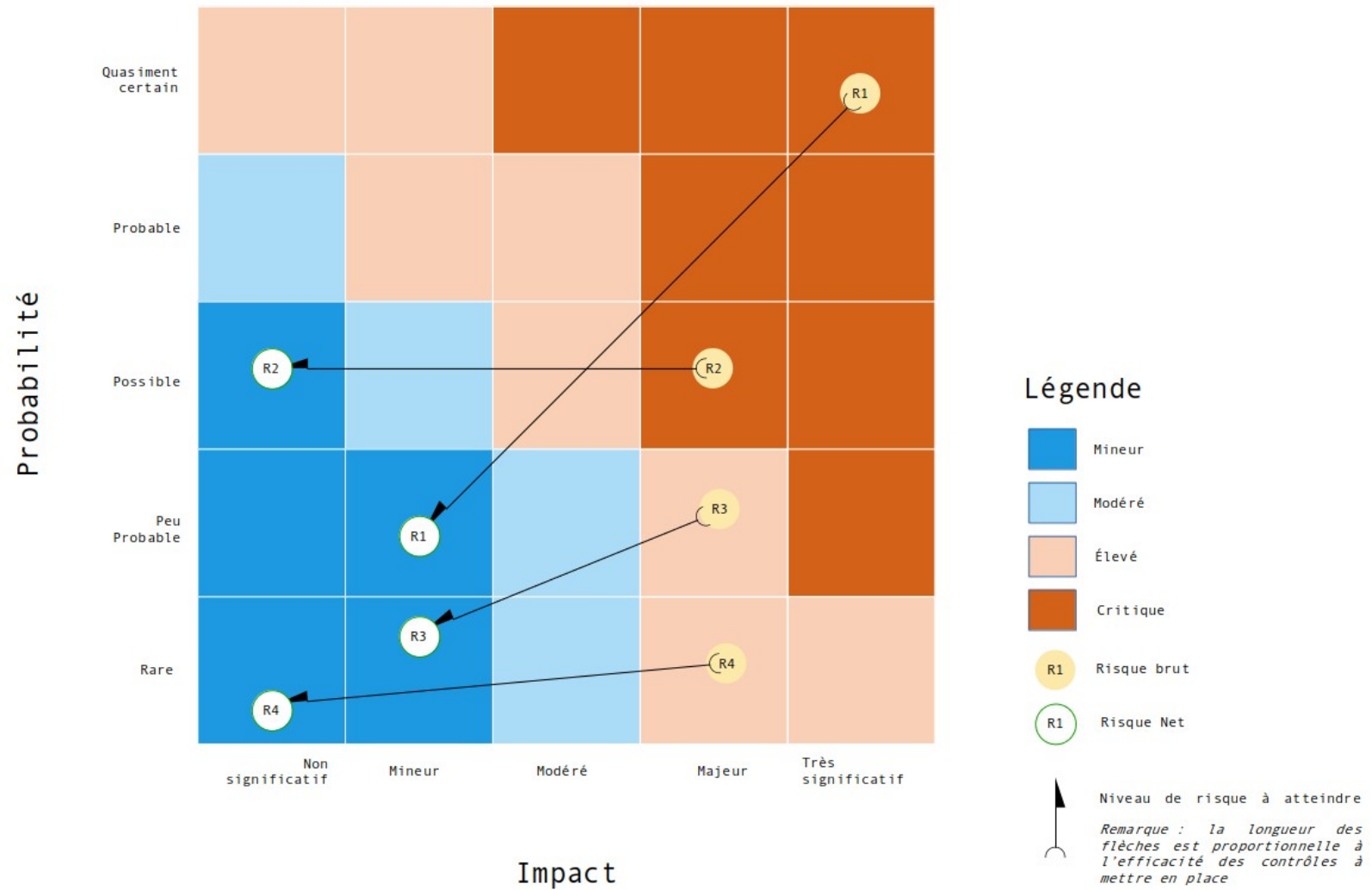
## Matrice des risques

Les risques, énumérés au sein du §Risques ci-dessus, sont recensés dans le tableau ci-dessous :

Id.	Risque	Description du risque	Type de risque	Domaine du risque	Impact	Probabilité	Action	Responsabilité	Date de l'examen	Efficacité des contrôles
R1	Humain	Risque lié à l'intervention humaine	Menace	Intervention	Majeur	Quasiment certain (>90%)	Automatiser de façon systématique toutes les opérations pouvant l'être	A++ Logistics		
R2	matériel	Panne matérielle	Menace	Technique	Majeur	Possible (30-50%)	Tout matériel se dégrade avec le temps, prévoir des backups de matériel en préconisant la redondance	A++ Logistics		
R3	logiciel	Comportement logiciel non désiré et non prévu	Menace	Technique	Majeur	Peu probable (10-30%)	Réaliser des tests unitaires durant la conception et des tests fonctionnels	A++ Logistics		
R4	environnement	Environnement hostile au bon fonctionnement de l'infrastructure matérielle	Menace	Technique	Majeur	Rare (<10%)	Respecter les normes et standards préconisés pour chaque matériel	A++ Logistics		

Cette matrice est représentée graphiquement par la cartographie des risques ci-dessous.

## Cartographie des risques



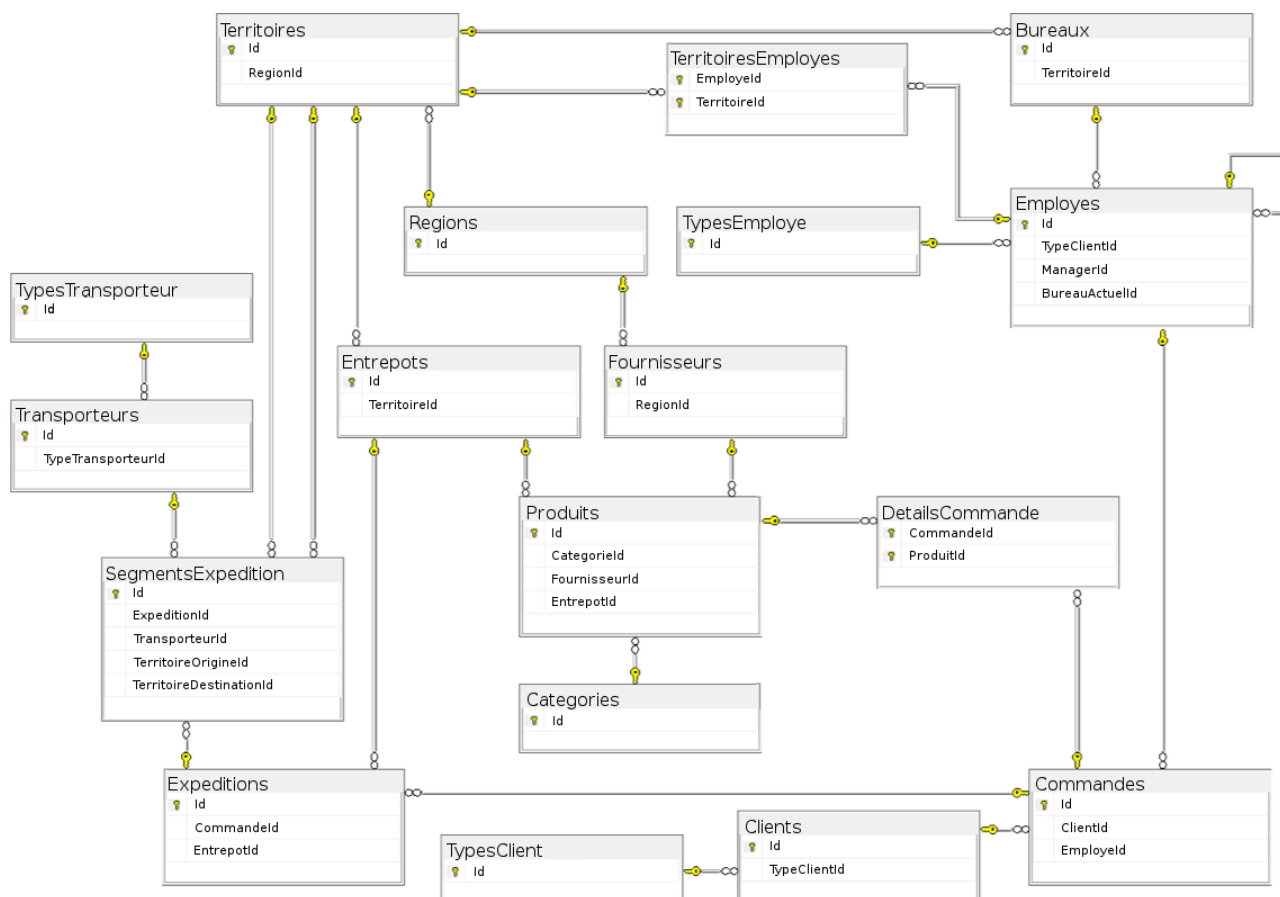


## Exigences fonctionnelles et techniques

La nouvelle solution logicielle devra répondre aux exigences suivantes :

- basée sur le cloud exclusivement, sans installation locale de logiciel ;
- IHM orientée web ;
- Accès utilisateur sécurisé nécessitant des méthodes de double authentification ;
- Sécurité interne à la solution basée sur la notion de rôle ;
- Référentiel des données affichées unique et adapté régionalement ;
- Présence de toutes les fonctionnalités propres à un ERP, dont notamment :
  - Planification à canaux multiples ;
  - Gestion financière ;
  - Achat et réception ;
  - Gestion de la production et des importations ;
  - Gestion prévisionnelle des coûts ;
  - Comparaison des coûts réels et prévisionnels ;
  - Fonctionnalité de reporting ;
- Fonctionnalité de veille économique pertinents adaptées aux procédures de travail d'A++ Logistics ;
- Outils simples de création de tableaux et graphiques ;
- Fonctionnalités TMS et PTMS spécifiques aux méthodologies d'A++ Logistics ;
- Lecture de codes à barres ;
- RFID pour les services de gestion de colis ;
- Suivi GPS pour la localisation durant les phases de transport notamment ;
- Évolutivité pour de futurs systèmes d'analyse et de suivi (pas de restriction en terme de scalabilité) ;
- Système de gestion d'entrepôt intégré (WMS).

En outre, la base de données de la nouvelle solution logicielle devra avoir la structure suivante :



La reprise de la structure ci-dessus, représentant l'actuelle base de données de ChERP utilisée au sein d'A++ Logistics, permettra de faciliter les imports de données relatives aux différentes sauvegardes réalisées à partir du système actuel, vers la nouvelle solution logicielle.

Ainsi, la structure ci-dessus répond aux fonctionnalités principales de TMS (table *Expeditions*, *Transporteurs*...), PTMS (tables *Commandes*, *DétailsCommande*...), WMS (tables *Entrepots*, *Produits*...). Cette structure pourra se voir étendue en fonction des différents besoins supplémentaires mis en exergue lors de l'étude des besoins relatifs à chaque bureau régional.

## Contraintes fonctionnelles

La nouvelle solution devra tenir compte des contraintes suivantes:

- Le délai de déploiement prévu devra être inférieur à 18 mois ;
- Le coût estimé du déploiement devra être inférieur de 25 % aux coûts fixes totaux de tous les systèmes régionaux actuels sur une période de 60 mois, soit un budget maximum de 5100000\$ ; le calcul de ce chiffre est basé sur le tableau d'estimation financière présent au §Annexes.

## Solution proposée

A partir des exigences fonctionnelles et techniques, ainsi que des contraintes mentionnées dans les paragraphes précédents, 3 types de solutions sont proposables :

- Solution propriétaire développée sur mesure ;
- Solution basée sur des composants de type AWS ;
- Solution de type COTS au travers de la suite logicielle Logistax.

### Solution propriétaire développée sur mesure

Dans le cas du développement d'une solution logicielle réalisée sur mesure, plusieurs choix technologiques sont possibles et seront à prendre en compte en fonction de :

- la pile technologique actuellement en place ;
- la matrice des compétences identifiée quant à la constitution de l'équipe projet ;
- l'estimation des coûts relatifs à l'infrastructure matérielle et logicielle à fournir pour produire, mettre en œuvre et maintenir le livrable final.

De plus, plusieurs choix techniques et technologiques peuvent être envisagés pour répondre à ce contexte :

- .Net avec modèle MVVM ;
- Node.js (Javascript) pour le backend et framework Angular (TypeScript) pour le frontend ;
- Java en backend et Javascript pour le frontend .

Les propositions de technologie ci-dessus ne doivent pas être perçues comme limitatives ou finales. D'ailleurs, elles seront toutes détaillées dans les paragraphes qui suivent.

Le choix final de la pile technologique devra être adapté à l'étude de l'existant effectuée durant la phase d'avant-projet découlant de ce choix de contexte de développement d'une solution sur mesure.

## .Net

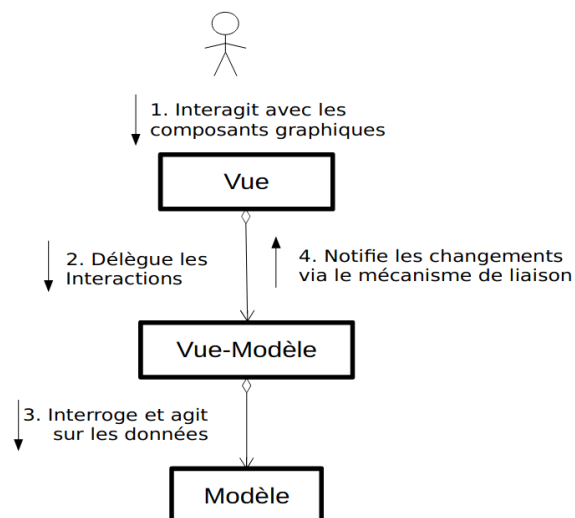
.NET est une plateforme de développement Open source pour la création d'applications, telles que :

- Applications Web, API Web et microservices ;
- Fonctions sans serveur dans le Cloud ;
- Applications cloud natives ;
- Applications mobiles ;
- Applications de bureau ;
  - WPF Windows ;
  - Windows Forms ;
  - Plateforme Windows universelle (UWP) ;
- Internet des objets (IoT) ;
- Apprentissage machine ;
- Applications de console ;
- services Windows...

Avec ce framework, il est possible de partager des fonctionnalités entre différentes applications et types d'applications à l'aide de bibliothèques de classes. Les fichiers du code et du projets sont alors les mêmes, quel que soit le type d'application créée. L'accès aux mêmes fonctionnalités de Runtime, d'API et de langage est donc possible avec chaque application. Les applications créées avec .NET sont portables sur de nombreux systèmes d'exploitation, dont notamment, Unix, Linux, Windows, macOS...

En outre, il est possible d'y appliquer un modèle MVVM (Modèle Vue Vue-Modèle), décrit par la figure ci-dessous, dont le retour d'information est pris en charge par un mécanisme de liaison de données (data binding). Ainsi, la présentation des données devient la vue-modèle, permettant d'accéder aux données prêtes à l'affichage dans la vue, à travers le patron Observateur sur chaque composant graphique de la vue. A chaque fois que la vue-modèle modifie un attribut, alors le composant graphique lié dans la vue est informé et peut se mettre à jour automatiquement.

Le diagramme fonctionnel ci-dessous représente le modèle MVVM :



## Backend Node.js (JavaScript) – Frontend Angular (TypeScript)

Node.js est un framework open source issu de Javascript, portable et utilisable au sein d'environnement d'exécution de type backend ; c'est à dire qu'il s'exécute en dehors d'un navigateur internet.

Ainsi, ce framework permet au développeur de coder des script côté serveur pour produire des pages web dynamiques s'adaptant aux requêtes utilisateur.

En outre, ce framework a une architecture orientée évènement capable de sortie asynchrone. Ses choix de conception cible l'optimisation de la cadence de production et de la scalabilité au sein des applications web nécessitant plusieurs opérations d'entrée/sortie, telle que les applications temps réel.

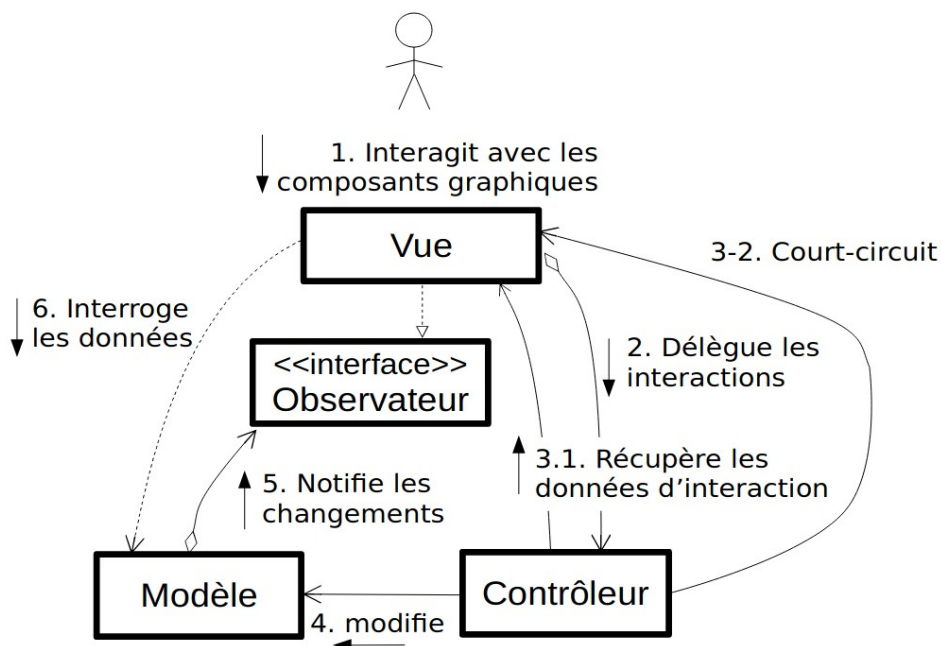
Dans le contexte de cette étude, ce framework serait adapté pour fournir, entre autres, les fichiers système d'entrée/sortie, l'aspect réseau (HTTP, HTTPS, TLS/SSL...), les interactions avec les SGBDR, les données binaires (buffers), les fonctions de cryptographie et les flux de données.

Associé à Node.js, l'utilisation d'Angular permettrait l'affichage des données fournies les applications côté serveur.

En effet, Angular est framework open source ,côté client, basé sur TypeScript. Ce framework permet de créer des applications web accessibles via une page unique, permettant de fluidifier l'expérience utilisateur sans charger de nouvelle page à chaque action de celui-ci.

Angular a une architecture de type MVC (Modèle Vue contrôleur) qui permet de séparer les données, le rendu visuel et les traitements. Ce modèle permet ainsi une meilleur répartition des responsabilités.

Le diagramme fonctionnel ci-dessous représente le modèle MVC :



## Backend Java – Frontend Javascript

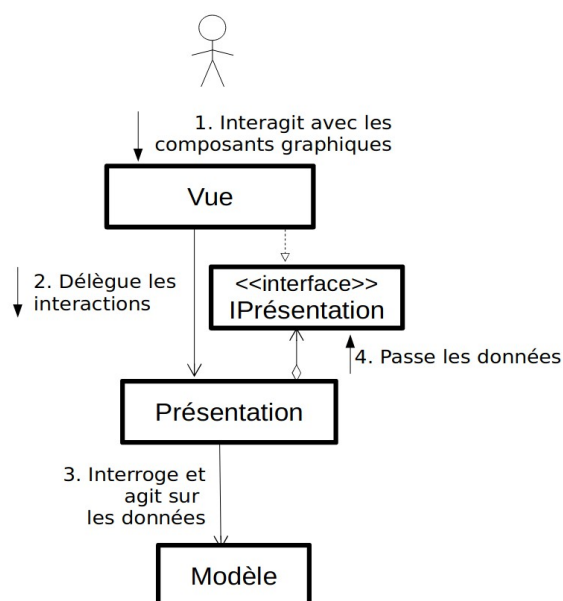
Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisé pour le développement d'applications d'entreprises et mobiles.

Java possède un certain nombre de caractéristiques, telles que :

- Java est interprété ;
- Java est portable, il est indépendant de toute plate-forme ;
- Java est orienté objet ;
- Java est fortement typé ;
- Java assure une gestion de la mémoire de manière automatique, autant à la création des objets qu'à leur suppression via un système de Garbage Collector ;
- Java est sûr, la sécurité fait partie intégrante du système d'exécution et du compilateur ;
- Java est économe, le pseudo code généré a une taille relativement petite car les bibliothèques de classes requises ne sont liées qu'à l'exécution ;
- Java est multitâche, il permet l'utilisation de threads qui sont des unités d'exécutions isolées (la JVM, elle même, utilise plusieurs threads).

Java peut alors se focaliser sur le backend de l'application développée alors que la pile Javascript peut se dédier au frontend, c'est à dire à l'affichage des données ; cet symbiose pourra se concrétiser en lui appliquant, par exemple, un modèle MVP.

Le diagramme fonctionnel ci-dessous représente le modèle MVP :



## ***Conclusion partielle***

Le développement d'une solution propriétaire sur mesure présente 2 principaux inconvénients :

- le coût de développement, de déploiement et d'installation d'une telle solution architecturale sera dispendieuse, au vu de tous les critères fonctionnels à prendre en compte, listés dans le § *Exigences fonctionnelles et techniques* ;
- la durée du projet de développement et d'installation relatives au sein des contraintes fonctionnelles, de 18 mois, n'est tout simplement pas concevable.

**Cet audit ne préconise pas le choix relatif au développement d'une solution logicielle sur mesure**

## Solutions AWS

Amazon Web Services est une division du groupe de commerce électronique Amazon. Cette division se spécialise dans la fourniture de services de cloud computing à la demande, proposant des services d'IaaS, PaaS et SaaS.

Ainsi, dans le cadre de cet audit, ce sera ce dernier service, SaaS, qui sera retenu pour la solution proposée et sera perçu, dans le reste de ce document, comme l'équivalent d'une architecture orientée service (SOA).

Le SaaS est un modèle d'exploitation commerciale des logiciels, au sein duquel les logiciels sont installés sur des serveurs distants plutôt que sur les machines des utilisateurs. Ces derniers ne financent pas de licence d'utilisation pour une version spécifique, mais utilisent librement le service en ligne en payant un abonnement.

Dans le cadre de cet audit, les principales architectures logicielles retenues de ce modèle seront :

- le gestionnaire de relation client (CRM) ;
- les solutions de création de site de commerce en ligne ;
- la gestion des ressources humaines ;
- les communications unifiées ;
- la messagerie et les logiciels collaboratifs ;
- la gestion des achats ;
- la gestion électronique de documents (GED).

Au travers des fonctionnalités susmentionnées, il faudra donc composer un workflow applicatif propre aux besoins métier d'A++ Logistics, afin d'intégrer l'aspect du système d'information souhaité et de l'assimiler à un coût de fonctionnement plutôt qu'à un investissement.

Ainsi, la définition stratégique de l'orientation métier définira un niveau de qualité de service (SLA).

Pour réaliser cela, AWS propose des solutions de migration nécessitant une approche progressive comprenant :

- une évaluation ;
- une préparation stratégique ;
- une planification ;
- des migrations.

Ces étapes sont basées sur la réalisation préalable d'une analyse d'ensemble du contexte fonctionnel, technique et technologique afin de choisir différents composants logiciels propre à une solution d'AWS.



Ainsi, dans le cadre de cet audit, la première suite AWS à choisir sera celle de l'instance de serveur virtuel ; ce dernier se basera donc sur la solution logicielle Amazon EC2 qui sera décrite dans le paragraphe §*AWS EC2*.

Une fois l'instance serveur choisie, il sera alors question de se pencher sur les fonctionnalités décrites au sein du paragraphe §*Exigences fonctionnelles et techniques*.

Ces dernières pourront être prises en compte, développées et déployées par le biais de plusieurs solutions AWS nommées :

- AWS Cloud Control API, décrit dans le §*AWS Cloud Control API* ;
- AWS Cloud Development Kit (CDK), décrit dans le §*AWS Cloud Development Kit* ;
- AWS CodeDeploy, décrit dans le §*AWS CodeDeploy*.

Enfin, l'évaluation financière totale, relative à l'utilisation de l'ensemble de ces services, sera estimée à l'aide de la solution AWS nommée AWS Pricing et décrite au §*AWS Pricing*.

## **AWS EC2**

Cette solution offre une plateforme de calcul variée et complète, possédant plus de 500 instances de composants enfichables disponibles. Cette solution est à considérer comme une solution PaaS.

L'utilisation de cette plateforme fournit, entre autres, un accès au(x) processeur(s) virtuel(s), aux réseaux, au système d'exploitation choisi et au modèle d'achat répondant aux besoins de la charge de travail envisagée. La fourniture de tous ces services s'effectue via un cloud avec des accès à plus de 400Gbit/s.

Il est également envisageable d'effectuer toute sorte de formation spécialisée pour évaluer le besoin stratégique dans des domaines spécifiques, tels que du machine learning ou de l'architecture de composant logiciel.

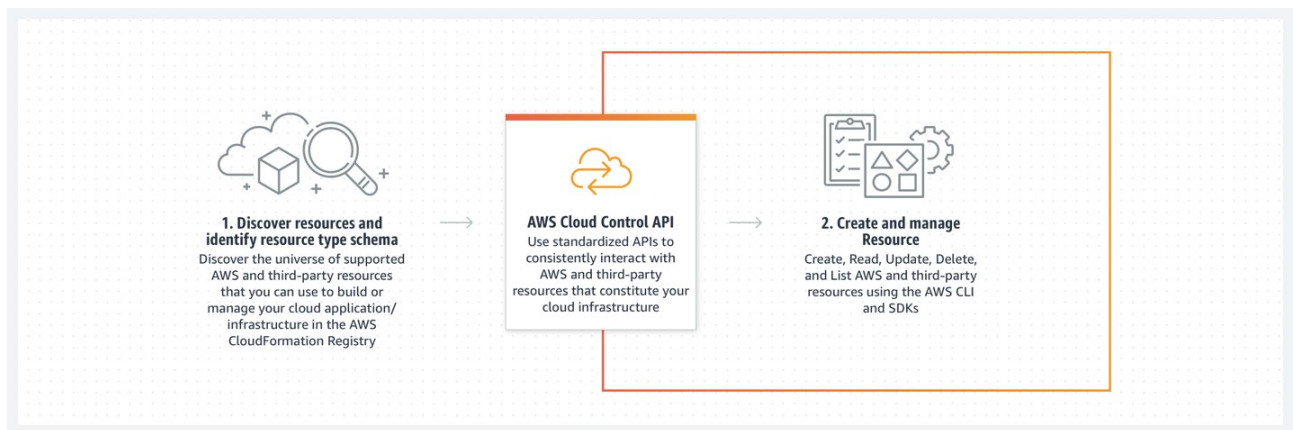
Il est alors possible d'envisager des charges de travail SAP, du calcul de haute performance (HPC) ou des analyses de Business Intelligence prédictive selon des critères évalués finement.

Pour la réalisation de cet audit, la solution proposée pour répondre aux besoins d'A++ Logistics sera basée sur la fourniture d'une infrastructure de calcul sécurisée, fiable, performante et sécurisée.

## AWS Cloud Control API

Cette solution AWS permet aux développeurs d'utiliser un ensemble d'APIs standardisées pour gérer des services de façon intuitive et avec des méthodes descriptives. Elle fournit donc la capacité de gérer programmatiquement des services et des fonctionnalités orientées métier.

Cette solution permet aux développeurs de créer un ensemble d'interfaces de programmation d'application de façon simple et aux partenaires métier de gérer le cycle de vie des applications métier, tel que décrit dans le schéma ci-dessous.



A l'utilisation de cette solution AWS, plusieurs cas d'utilisation sont donc envisageables, tels que :

- Créer, lire, mettre à jour, supprimer et lister des ressources métiers et des composants enfichables AWS ;
- Exposer de nouvelles ressources ou fonctionnalités métier automatiquement ;
- Provisionner des ressources pour des partenaires à partir d'outils conviviaux existant...

Enfin, l'évaluation financière relative à l'utilisation de cette solution AWS sera estimée à l'aide de la solution AWS nommée AWS Pricing et décrite au §AWS Pricing.

## **AWS Cloud Development Kit**

Cette solution AWS propose des services de développement à partir de langages de programmation pour définir une infrastructure AWS et y développer des services sur mesure orientés métier.

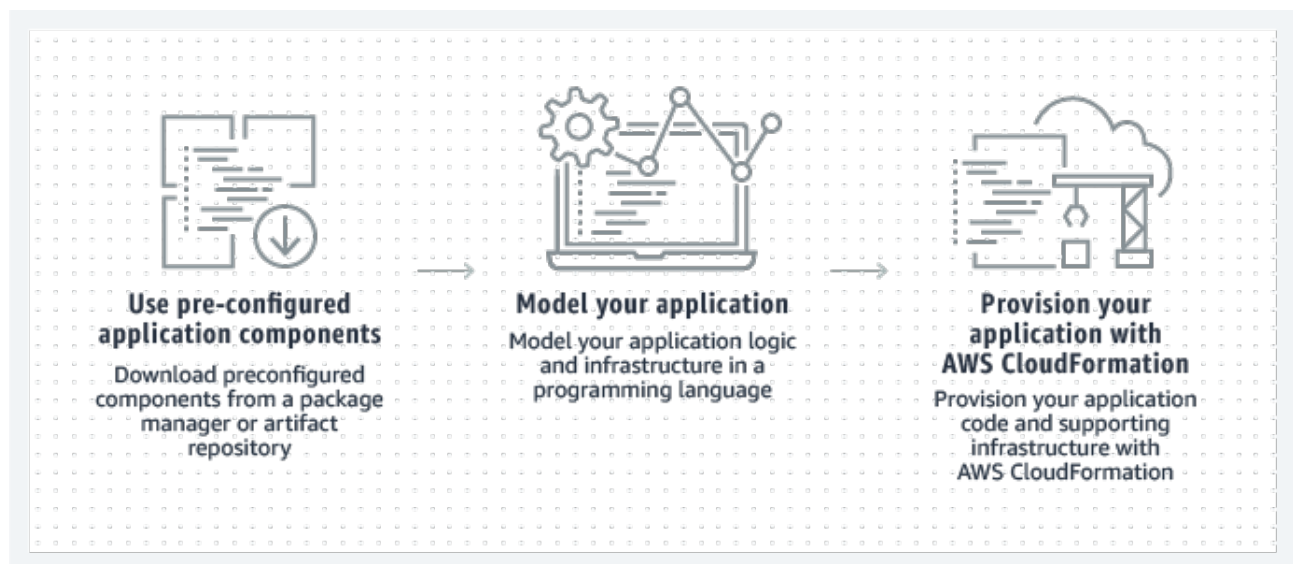
Ainsi, des fonctionnalités familières et communes aux langages de programmation orienté objet seront à la disposition des développeurs, tels que les objets, les boucles ou les concepts de base d'héritage, de polymorphisme, d'encapsulation et d'abstraction.

Les développeurs auront à leur disposition un environnement de développement intégré (IDE) spécifique pour tirer parti des outils de productivité et des frameworks de test existant adaptés aux besoins stratégiques métier définis.

Ces outils de développement seront adaptées au contexte de cloud computing et permettront de créer, tester et déployer des applications cloud sans quitter l'environnement de développement intégré (IDE). Il sera donc possible de :

- créer des codes d'exécution et définir des ressources AWS avec le même langage de programmation ;
- visualiser les piles ainsi créées ;
- afficher de manière conviviale et simple les ressources de votre application métier avec des outils spécifiques, tels que *AWS Toolkit for VSCode*.

Le schéma ci-dessus représente fonctionnellement l'utilisation de cette solution AWS.



Enfin, l'évaluation financière relative à l'utilisation de cette solution AWS sera estimée à l'aide de la solution AWS nommée AWS Pricing et décrite au §*AWS Pricing*.

## AWS CodeDeploy

Cette solution est un service entièrement géré qui automatise les déploiements de logiciels développés à partir d'outils tels qu'*AWS Cloud Development Kit*.

AWS CodeDeploy permet de lancer rapidement et facilement de nouvelles fonctionnalités, et d'éviter les temps d'arrêt pendant le déploiement d'une application, tout en gérant la complexité de la mise à jour de vos applications.

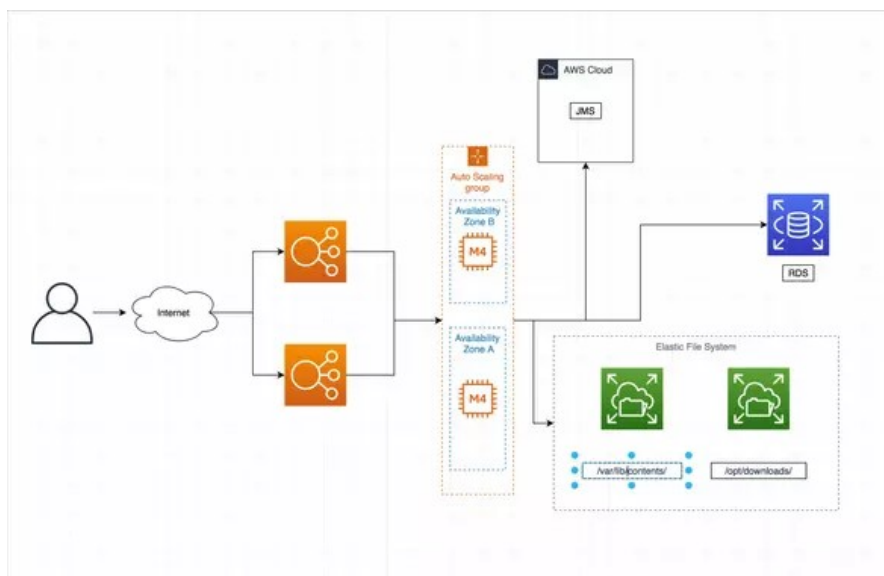
Il est alors possible d'automatiser les déploiements de logiciels, éliminant ainsi le besoin d'opérations manuelles sujettes aux erreurs. Le service se met à l'échelle pour correspondre à vos besoins de déploiement.

Cette solution AWS permet de lancer et de suivre facilement le statut des déploiements d'applications et fournit un rapport détaillé permettant de voir quand et où chaque révision d'application a été déployée. Il est également possible de créer des notifications, de type *push*, et de recevoir des mises à jour en direct, à propos de vos déploiements, sur n'importe quel type de média (smartphone, tablette, PC...).

Ainsi, AWS CodeDeploy permet d'optimiser la disponibilité des applications métier lors du processus de déploiement logiciel. Elle introduit les modifications de manière incrémentielle et suit la santé de l'application en fonction des règles configurables. Les déploiements de logiciels peuvent être facilement arrêtés et restaurés si des erreurs se produisent.

Cette solution AWS est indépendante de la plate-forme et de la langue, fonctionne avec n'importe quelle application et offre la même expérience peu importe la plateforme de déploiement cible, par exemple AWS EC2. Il est alors très simple de réutiliser facilement du code de configuration existant et de l'intégrer avec votre processus de lancement de logiciel existant ou votre chaîne d'outils de livraison continue (par exemple AWS CodePipeline, GitHub, Jenkins...).

Le diagramme ci-dessous décrit fonctionnellement le processus de déploiement continu associé à AWS CodeDeploy :



Enfin, l'évaluation financière relative à l'utilisation de cette solution AWS sera estimée à l'aide de la solution AWS nommée AWS Pricing et décrite au §AWS Pricing.

## **AWS Pricing**

AWS Pricing Calculator est un outil d'estimation qui fournit un coût approximatif d'utilisation des services AWS en fonction des paramètres d'utilisation que l'utilisateur spécifie.

AWS Pricing Calculator n'est pas un outil de calcul de devis et ne garantit pas le coût d'utilisation réelle des services AWS. Le coût estimé par AWS Pricing Calculator peut différer des coûts réels pour un certain nombre de raisons. Les raisons courantes pour lesquelles l'estimation peut être différente du coût réel incluent :

1. **Utilisation réelle** : le coût réel sera basé sur l'utilisation réelle des services et non pas du calculateur. La précision des entrées dans le calculateur détermine celle des coûts réels. Par exemple,
  1. Si vous estimez que vous allez utiliser un certain nombre d'instances EC2 et que vous en utilisez réellement plus, vous êtes facturé pour votre utilisation réelle.
  2. Si vous utilisez des services AWS que vous n'avez pas inclus dans votre estimation, cette dernière sera différente de votre facture réelle.
  3. Si vous surestimez ou sous-estimez les entrées ou sorties de transfert de données pour les services AWS, votre facture réelle peut varier.
  4. Si vous estimez votre utilisation des services sur la base de l'achat d'une réservation de 3 ans pour un service et que vous achetez une réservation d'un an, vos prix mensuels et initiaux diffèrent de l'estimation.
2. **Région utilisée** : les prix des services AWS varient selon les régions. L'utilisation d'une région différente de celle utilisée dans le calculateur affecte l'exactitude des résultats.
3. **Évolution des prix** : AWS propose une tarification à l'utilisation pour la plupart des services, et les prix peuvent changer au fil du temps. Si vous achetez des services à la demande, votre facture peut être inférieure ou supérieure aux estimations, en fonction des tarifs actuels du service à la demande. Pour certains services, vous pouvez acheter une réservation de 1 ou 3 ans pour bloquer le prix à la date de votre réservation.
4. **Taxes** : le calculateur n'inclut aucune taxe pouvant être appliquée à votre achat des services.
5. **Période considérée** : le calculateur suppose 730 heures par mois (365 jours par an x 24 heures par jour/12 mois dans une année). Le total estimé pour les douze premiers mois est calculé selon la formule suivante :  $12 \times \text{le coût total mensuel} + \text{total des coûts initiaux}$ . Le calculateur ne tient pas compte des années bissextiles, ce qui ajoute un jour.
6. **Total sur les 12 premiers mois** : le calculateur n'affiche que vos totaux estimatifs jusqu'aux 12 premiers mois d'utilisation. Si vous estimez les coûts avec une politique de tarification initiale partielle de 3 ans, vous ne voyez que les coûts estimés ajustés pour les 12 premiers mois de la réservation de 3 ans.
7. **Offre gratuite, crédit promotionnel et réductions** : le calculateur ne tient pas compte des offres gratuites, des crédits promotionnels ni des autres réductions. Vous pouvez être éligible à des rabais supplémentaires qui ne sont pas inclus dans l'estimation.

8. **Période de facturation mensuelle** : AWS facture tous les mois. Si votre utilisation commence au milieu du mois, vous ne voyez qu'une partie des coûts complets d'un mois.
9. **Arrondi** : le calculateur arrondit mathématiquement les données de tarification. Pour plus d'informations sur la façon dont le calculateur calcule l'arrondi, reportez-vous au calcul de tarification dans la section « Afficher le calcul » de chaque configuration de service.
10. **Tarification différenciée** : vous pouvez bénéficier de réductions sur le volume en fonction de votre utilisation actuelle des services AWS, en dehors de ce que vous estimez dans l'AWS Pricing Calculator. Le calculateur ne tient pas compte de l'utilisation que vous n'incluez pas dans l'estimation. Par exemple, si vous utilisez déjà 500 To par mois de stockage Amazon S3 Standard, vous payez 0,022 USD par Go. Si vous utilisez le calculateur pour estimer 50 To supplémentaires par mois, il suppose qu'il s'agit des 50 premiers To par mois, dont le coût est supérieur à 0,023 USD par Go. Vos dépenses réelles cumulées sur 550 To par mois sur Amazon S3 seront inférieures au coût supplémentaire estimé de vos dépenses actuelles.
11. **Tarification à la seconde** : certains services AWS utilisent la facturation par seconde. Le calculateur ne prend pas en compte les options de facturation par seconde.
12. **Frais de licences tierces** : le calculateur ne prend pas en compte les frais de licence tierces, tels que les solutions logicielles déployées à partir d'AWS Marketplace.
13. **Devise** : le calculateur calcule les prix en dollars américains. Les taux de change mondiaux varient dans le temps. Si vous convertissez votre estimation dans une devise différente s taux de change actuel, les variations du taux de change affectent l'estimation.

Au vu des nombreux des critères recensés ci-dessus, la variation de prix d'installation d'un système complexe basé sur des composants AWS peut fortement varier. A cela, il faudra ajouter l'estimation des coûts d'utilisation qui peuvent ,eux-aussi, fortement varier en fonction du nombre d'utilisateurs et d'utilisations à prévoir.

## ***Conclusion partielle***

Ainsi, les estimations financières relatives à cet audit seront basées sur les critères suivants :

- toutes les solutions AWS mentionnées ci-dessus représentant la chaîne d'intégration continue, de déploiement continu et de livraison continue ;
- une durée du projet de 18mois ;
- 1000 utilisateurs finaux.

**La valeur d'utilisation annuelle d'une telle architecture logicielle est estimée à 1600000€ annuellement.**

En outre, cette estimation n'est que l'estimation de l'utilisation de la solution AWS, si cette option était entérinée. Ce chiffre ne prend en considération ni le temps de développement ni les coûts associés. En effet, il est difficile d'estimer de tels coûts sans disposer des matrices de compétence à disposition, relative aux ressources humaines en interne.

La principale connaissance à mettre en exergue pour le développement d'une telle solution est une connaissance exhaustive des composants architecturaux AWS, dont il faudrait disposer pour répondre aux besoins.

La prise de conscience relative à la présence ou à l'absence des compétences nécessaires quant au développement d'APIs basées sur un environnement de cloud computing, à la formation et la prise en main des utilisateurs d'une telle solution et à sa maintenance durant son cycle de vie, sont des données nécessaires et suffisantes pour une analyse financière plus détaillée.

Ainsi, cet audit ne dispose pas d'assez de données pour estimer de façon correcte l'emploi de cette solution logicielle.

**Cet audit ne préconise pas le choix des solutions AWS**

## Solution COTS : LogiStax Suite 20

La suite logicielle LogiStax Suite 20 effectue les opérations de TMS, PTMS, WMS et de reporting. Elle est notamment composée d'une base de données contenant toutes les informations relatives à toutes les fonctions listées précédemment.

Ainsi, cette solution est actuellement utilisée de façon on-premise, c'est à dire qu'elle est installée et exécutée sur des ordinateurs locaux et dont l'architecture logicielle est orientée cloud. Un tel type d'architecture est associée à du SaaS et dénommée cloud computing. Ce genre de logiciel comprend généralement des modules de traitement dédiés au métier et des bases de données adaptées aux fonctionnalités appréhendées. Cette infrastructure logicielle sert particulièrement des besoins uniques au sein d'organisations professionnelles.

Néanmoins, de par la répartition géographique des bureaux régionaux d'A++ Logistics, une architecture full cloud serait plus appropriée et à envisager de par les besoins exprimés.

Partant de ce constat, il est tout à fait concevable de porter une architecture on-premise vers du full cloud, à moindre coût d'investissement.

Pour réaliser cette migration, il est alors envisageable de partir sur une solution de type AWS EC2, décrite dans le §AWS EC2 ci-dessus, puis d'y installer les couches logicielles nécessaires et suffisantes à l'environnement d'exécution de *LogiStax Suite*.

Une fois l'environnement technique installé, les opérations de création de conteneur de données devront être réalisées, telles que la création de la structure de base de données décrite au §*Exigences fonctionnelles et techniques*, la création des workflows d'échange de données des applications fonctionnelles,...

La phase préparatoire de création des conteneurs de données achevées, l'exportation et l'importation de données seront réalisées à partir de formats XML ou JSON, disponibles également depuis les architectures logicielles des autres bureaux régionaux.

En outre, cette suite logicielle disposant d'une API Python, il est tout à fait possible et envisageable d'en étendre ses fonctions embarquées en développant des produits spécifiques aux besoins exprimés.

En ce qui concerne spécifiquement le reporting, cette fonction pourra être alors entièrement prise en charge par LogiStax Suite pour homogénéiser les présentations et publications information, ainsi que pour en réduire les échanges.



Cette suite logicielle présente plusieurs avantages fonctionnels indéniables, en terme de garantie relative à :

- la pérennité des informations ;
- la portabilité des fonctionnalités ;
- la limite des risques au niveau de la compliance (conformité) ;
- le respect des standards internationaux, de par la qualité internationale d'A++ Logistics ;
- la sécurité d'accès et de stockage des données, de par l'homogénéisation des procédures de Sécurité ;
- aucune dépendance vis-à-vis de plusieurs standards ;
- une excellente extensibilité, de par la présence de son API Python ;
- un contrôle total du logiciel en centralisant les accès ;
- une disponibilité des données ;
- la fait que nous ayons déjà un bureau régional qui dispose d'une bonne expérience quant l'installation et l'utilisation de cette application ;
- un prix d'installation et d'utilisation attractif.

## **Conclusion partielle**

La suite logicielle Logistax répond à tous les critères listés au sein du §*Exigences fonctionnelles et techniques*, dont :

- les fonctions de TMS, PTMS et WMS ;
- la numérisation de codes à barres et de RFID ;
- le suivi GPS qu'il sera possible d'étayer en utilisant l'API Python ;
- les fonctions de reporting, autant en terme de création de rapport que de moteur de recherche relatif aux rapports générés.

Le coût d'installation d'une telle solution est estimée à 3500000€, comprenant ainsi le coût d'utilisation d'AWS EC2, en tant qu'infrastructure PaaS. Ce coût est donc inférieur à celui énoncé dans le §*Contraintes fonctionnelles*, de 5100000\$, soit environ 4700000€.

En outre, le coût annuel d'utilisation de cette solution logicielle est estimé à 180000€, dont 139000€ pour la licence annuelle de LogiStax Suite 20 et 41000€ pour l'utilisation d'AWS EC2.

Enfin, la durée de projet d'installation, de déploiement et de mise en œuvre d'une telle solution est estimé à 18 mois., durée préconisée au sein du §*Contraintes fonctionnelles*.

**cet audit préconise l'emploi de la solution LogiStax Suite 20, sur une infrastructure de type cloud, pour tous ces bureaux régionaux et les régions satellite qui leur sont affiliées**



## Conclusions et opinions de l'audit

La présente section a pour objet de fournir une conclusion générale ou une opinion concernant les objectifs de la mission d'audit. Naturellement, la mission a été effectuée sous la forme d'un examen comportant un niveau approprié d'audit par sondages, conformément à toutes les normes d'audit pertinentes, et des conclusions fondées sur des éléments probants suffisants, pertinents et valides.

Les critères utilisés lors de l'audit comprenaient les politiques et procédures de gestion, *[insérer l'identification des références légales ou réglementaires]* et les directives de contrôle de gestion décrites dans COBIT® 5, telles que publiées par ISACA.

**[ A COMPLETER ]**



## Annexes

### Tableau de sourcing

<div> Document Domaine </div>			Domaine à évaluer							
			Architecture du système	Hypothèse de performance	Technologies existantes	Capacité du système	Risques et limites	Solution proposée	Force	Faiblesse
Ressources documentaires liées au projet	E-mail d'Evan Potter		X	X			X	X		
	Architecture du système	site de Boston	X	X	X	X			+ça marche la plupart du temps +confortable à utiliser	-ça ne marche pas de temps en temps -c'est un peu lent -estimer comme inefficace et déroutant par Francfort
		site de Dubaï	X		X	X				
		site de Francfort	X	X					+estimer comme plus efficace et plus approprié	
		site de Hong Kong	X	X	X	X			+estimer comme plus efficaces que les composants de Boston +moins de composants constituant la solution	
		site de Londres	X		X	X				
		site de New Delhi	X	X	X	X	X	X	+améliorer l'efficacité et la gestion de données	
	Processus de développement		X							
	Descriptif du projet		X	X	X	X	X	X		
	Système actuel – Technologies utilisées		X	X	X	X	X	X		

<div> <div>Document</div> <div>Domaine</div> </div>		Référentiel TOGAF® 9.1				
		A. Vision architecturale	B. Architecture d'entreprise	C. Architecture des Systèmes d'Information	D. Architecture technologique	E. Solutions et Opportunités
Ressources documentaires liées au projet	E-mail d'Evan Potter	X	X	X		X
	Architecture du système	X		X	X	X
		X		X	X	X
		X		X		
		X		X	X	
		X		X	X	
		X		X	X	X
	Processus de développement	X				
	Descriptif du projet	X	X	X		X
	Système actuel – Technologies utilisées	X	X	X	X	X

SI actuel Site	Logiciel	Nombre de licence	Coût annuel		Coût total annuel		Coût fixe	
			\$	€	\$	€	\$	€
site de Boston	ChERP	23	240000	220746	265885	244555		
	A-PLUS TMS	1	9990	9188,57				
	CodeScanDoc	1	3995	3674,51				
	Pachyderm WMS	1	11900	10945,3				
site de Dubaï	ChERP	11	150000	137967	177297	163074		
	A-PLUS TMS	?	9742,578	8961				
	ScanMagixWMS	?	17554,25	16146			1456,87	1340
site de Francfort	ChERP	17	195000	179357	212270	195242		
	TruTMS + Pachyderm WMS	?	17270,49	15885				
site de Hong Kong	ChERP	13	165000	151763	175909	161797		
	TruTMS	?	5454,583	5017				
	ScanMagix	?	5454,583	5017			2348,4	2160
site de Londres	ChERP	17	195000	179357	212673	195612		
	TruTMS	?	5653,544	5200				
	Pachyderm WMS	?	12019,22	11055				
site de New Delhi	ChERP	13	165000	151763	315734	290405		
	Logistax	?	150734,4	138642				

	\$	€
Taux de change Au 12 avril 2022	1,08722	1
75 % Somme des Coûts annuels et Coûts fixes	1022680	940638
75 % Somme des Coûts annuels et Coûts fixes Sur 5ans	5113402	4703190



# A++ Logistics