**ÉVALUATION DE LA CONFORMITÉ**

**- SuperTechSoft –**

****

**David EVAN**

**24/06/2022**

**Version 1.0**

**SCS Gestion Electronique des Documents – SuperTechSoft**

**Historique des révisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Numéro de version** | **Auteur** | **Description** | **Date de modification** |
| 1.0 | EVAN David  *(Architecte logiciel)* | Livraison initiale | 24/06/2022 |
|  |  |  |  |
|  |  |  |  |

Tableau - Historique des révisions

**Objectif du document**

Dans le cadre du projet de création d’une GED pour SCS Magazine, ce document a pour objectifs de s’assurer que la conformité de l’architecture livrée est en conformité avec, d’une part, la vision architecturale telle que défini dans le *Cahier des charges d’architecture* et le *Document de Définition d’Architecture, ainsi que,* d’autre part, en conformité avec le référentiel d’architecture de l’entreprise.

L’architecture n’ayant pas encore été livrée, l’analyse de la conformité sera basée sur le document détaillant *les spécifications techniques* de l’architecture.

**Note :**

Ce document devra être révisé une fois l’architecture livrée afin de s’assurer du maintien de la conformité et du respect des spécifications techniques.

Par ailleurs, une majeure partie du référentiel d’architecture de l’entreprise étant non documenté au moment de la rédaction de ce document, une révision sera nécessaire pour s’assurer que les conceptions sont conformes aux principes, normes et exigences d’architectures définies à l’échelle de l’entreprise.

**Table des matières**

[CONTEXTE & ÉTAT D’AVANCEMENT 4](#_Toc106973047)

[Contexte 4](#_Toc106973048)

[État de l’avancement 4](#_Toc106973049)

[RÉFÉRENCIEL POUR L’ÉVALUATION 5](#_Toc106973050)

[Référentiel d’architecture 5](#_Toc106973051)

[Principes d’architectures 5](#_Toc106973052)

[Spécifications non fonctionnelles et contraintes imposées 5](#_Toc106973053)

[Vision d’architecture 6](#_Toc106973054)

[Implémentation de l’architecture 7](#_Toc106973055)

[ÉVALUATION DE LA CONFORMITÉ 8](#_Toc106973056)

[Terminologie 8](#_Toc106973057)

[Évaluation 8](#_Toc106973058)

[Architecture technique 8](#_Toc106973059)

[Architecture logicielle - Applicatif 9](#_Toc106973060)

[Services logiciels et middlewares 10](#_Toc106973061)

[Sécurité 11](#_Toc106973062)

[Gestion de l’information 12](#_Toc106973063)

[Gestion des systèmes informatiques 13](#_Toc106973064)

[Gestion de projet et outillage 14](#_Toc106973065)

[PROCESSUS DE RÉVISION DE LA CONFORMITÉ 15](#_Toc106973066)

[APPROBATIONS 16](#_Toc106973067)

[Approbation du document 16](#_Toc106973068)

[TABLES DES RÉFÉRENCES 17](#_Toc106973069)

[Figures 17](#_Toc106973070)

[Tableaux 17](#_Toc106973071)

# CONTEXTE & ÉTAT D’AVANCEMENT

## Contexte

Le projet SCS GED consiste au développement d’un nouveau produit pour le magazine scientifique « SCS Magazine ».

La rédaction d’un article engage de nombreux interlocuteurs : chercheurs, rédacteurs et éditeurs.Cette hétérogénéité d’acteurs entraîne de nombreux aller-retours afin d’aboutir au résultat final.  
En l’absence d’outil spécifique pour traiter ces besoins, SCS a identifié des problématiques de communication entre les différentes parties-prenantes que et souhaite les résoudre avec un nouvel outil de GED (Gestion Électronique des Documents). Cet outil devra être intégré au SI existant.

## État de l’avancement

L’avancement du projet est en conformité avec la Roadmap définie (*voir document associé*).

Les livrables déjà produits sont :

* Le cahier des charges d’architecture (tenant lieu de référentiel d’exigences et de vision d’architecture)
* Le document de définition d’architecture
* La roadmap d’architecture
* Les spécifications techniques

D’un point de vue technique, le projet est suffisamment mature pour être transféré au prestataire chargé de la MOE. Un point de vigilance toutefois cependant être apporté à la chefferie de projet actuellement mal identifié.

Les livrables suivant sont **manquant** ou à **approfondir** pour de garantir le succès de la réalisation :

* Identification des parties prenantes et du rôle associé à chacun (notamment le client final)
* Plan de test et d’implémentation

Il est actuellement noté l’absence d’un plan de test et visant à mettre en relation les exigences fonctionnelles / non fonctionnelles avec la livraison du futur système.

Afin de garantir la conformité de la livraison, il est recommandé que les exigences fonctionnelles soient à nouveau révisées (précisés et complétés, si nécessaires) et qu’un plan de test soit définies.

Notons par ailleurs qu’à ce stade, aucune approbation des parties prenantes n’a pu être constaté sur les livrables terminées. Cette approbation devrait accordée avant de démarrer les développements afin de s’assurer de la validation des documents par l’ensemble des parties prenantes.

# RÉFÉRENCIEL POUR L’ÉVALUATION

## Référentiel d’architecture

### Principes d’architectures

Le système proposé doit répondre aux bonnes pratiques de conception générales, à savoir :

* L’architecture doit être au service de l’expérience utilisateur. (**Performance**)
* L’architecture doit faciliter l’innovation rapide et s’adapter aux besoins business. (**Évolutivité**)
* L’architecture doit faciliter les interactions entre composants logiciels. (**Interopérabilité**)
* L’architecture doit être capable de s’adapter aux besoins des utilisateurs. (**Scalabilité**)
* L’architecture doit être orientée sécurité et garantir la protection des données et l’accès aux ressources en fonction des habilitations. (**Sécurité**)

### Spécifications non fonctionnelles et contraintes imposées

Les contraintes d’architecture et les exigences de conception pour le développement de la solution imposent :

* Que les couches backend du produit soient développées en **Java (Spring)**.
* Que les interfaces utilisateurs (Front layer) repose sur un Framework adaptable à une architecture micro frontend.
* Que la couche service soit décorrélée du front et exposée à travers un service de fédération d’API (**API Gateway**).
* Que les composants développés puissent être déployés au travers de conteneurs.
* Que l’accès à la plateforme doit être sécurisé (tant fonctionnellement que techniquement).
* Que la plateforme soit accessible sur tout support web (responsive).

D’une manière générale, la conception doit répondre aux bonnes pratiques de l’industrie et assurer la sécurité des données et des utilisateurs.

**Note :**

La présente section devrait être complétée avec un lien vers le référentiel d’architecture de l’entreprise SCS Magazine afin de s’assurer que l’architecture produite est en conformité avec les principes établis à l’échelle de l’entreprise.

## Vision d’architecture

La vision de l’architecture est détaillée dans le *cahier des charges d’architecture* et traduite dans le *document de définition d’architecture*.

Le schéma ci-après (*figure 1*) présente *le diagramme d’architecture* de la future solution SCS GED.

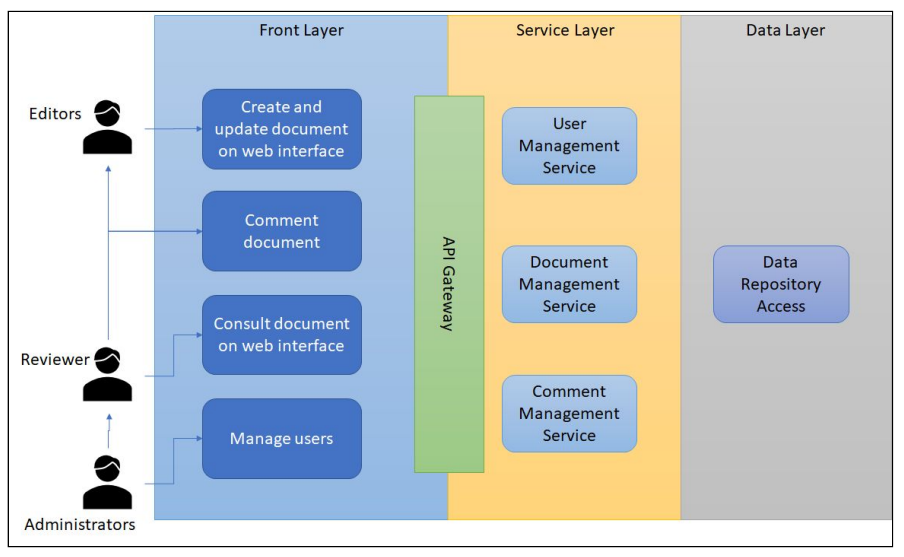


Figure : Vision de l'architecture cible du projet "SCS GED".

La consultation des documents référencés ci-avant permet de préciser la vision d’architecture.

## Implémentation de l’architecture

L’implémentation de l’architecture n’étant actuellement pas finalisée, l’évaluation de la conformité portera sur *les* *spécifications techniques* établies et supposera une livraison conforme à ces spécifications.

Le schéma ci-après (*figure 2*) rappel l’implémentation de l’architecture.



Figure : Diagramme d'implémentation de l'architecture du projet "SCS GED"

Le lecteur est invité à consulter *le document de spécifications techniques* afin d’obtenir une vision plus détaillée de l’implémentation de l’architecture.

# ÉVALUATION DE LA CONFORMITÉ

## Terminologie

La terminologie utilisée pour définir le statut de l’évaluation dans les tables des sections suivante est précisée ci-après :

* **Conforme** : Le statut « *conforme* » indique que les éléments disponibles permettent de confirmer que le critère évalué est conforme à l’attendu.
* **Non conforme** : Le statut « *non-conforme* » indique que l’information disponible permet d’affirmer que tout ou partie du système évalué ne satisfait pas au critère évalué.
* **Non applicable** : Le statut « *non applicable* » indique que le critère ne peut pas être évalué dans le système car aucun artefact connus n’est concerné par l’affirmation.
* **Non évaluable** : La statut *« non évaluable* » indique que l’information disponible ne permet pas de connaître le degré de conformité du critère évalué. Ces points devraient faire l’objet d’une demande de précision auprès des parties prenantes pour obtenir l’information manquante et réviser l’évaluation.

## Évaluation

### Architecture **technique**

|  |  |  |  |
| --- | --- | --- | --- |
| **Critère d’évaluation** | **Statut** | | **Commentaire** |
| Les supports de stockage des données (base de données, stockage de fichier ...) sont fiables et tolérants aux erreurs critique. |  |  |  |
| La scalabilité du système est assuré par un mécanisme automatique permettant la monté en charge sans intervention humaine pour répondre aux pics d’activité. |  |  |  |
| Tous les actifs (applicatifs, composants technique ou base de données) déployés bénéficie d’une réplication sur au moins 2 instances. |  |  |  |
| L’architecture technique est tolérante aux erreurs critiques sur les instances individuelle d’un actif (pertes d’une instance par exemple) sans affecter le bon fonctionnement. |  |  |  |
| Les API sont exposées sur le web sur un end-point unique au travers d’une API Gateway. |  |  |  |
| L’architecture technique embarque des mécanismes de reprise automatique pouvant redémarrer tout ou partie des composants sans intervention humaine. |  |  |  |
| L’architecture technique est déployée sur une plateforme cloud (privée ou publique). |  |  |  |
| L’architecture technique est construite sur la base du respects des standards (composants ou protocoles) autant que possible. Les systèmes ou protocoles non-standards déployés répondent à un besoin clairement identifié et justifié. |  |  |  |

Tableau : Table d'évaluation de l'architecture technique

### Architecture logicielle - Applicatif

|  |  |  |  |
| --- | --- | --- | --- |
| **Critère d’évaluation** | **Statut** | | **Commentaire** |
| La conception de l’architecture logicielle est conforme aux différents pattern architecture exigées (micro-front end, micro service …). |  |  |  |
| L’architecture logicielle est construite de manière à réduire au maximum le couplage entre les différents composants. |  |  |  |
| Les applicatifs utilisent des technologies (format, langages …) standardisés. Les technologies non-standards utilisées répondent à un besoin clairement identifié et justifié. |  |  |  |
| Les applicatifs exposés sur le web sont conformes aux exigences de « responsives design » exprimées. |  |  |  |
| Les applicatifs exposés sur le web sont exploitables sur la liste de navigateurs exprimées par les exigences non fonctionnelles. |  |  |  |
| Les applicatifs utilisés ou développés sont correctement documentés. La documentation permet d’identifier facilement les fonctionnalités rendues par l’applicatif. |  |  |  |
| Les applicatifs disposent d’un mode « maintenance » permettant l’interruption temporaire des services avec information à l’utilisateur d’une opération de maintenance / mise à jour en cours. |  |  |  |
| L’architecture logicielle est conçue pour favoriser l’interopérabilités entre les différents composants. |  |  |  |
| L’architecture logicielle favorise la réutilisation des systèmes existant et l’absence de redondance. Il n’existe pas d’applicatif déjà déployés permettant de satisfaire aux besoins exprimées. |  |  |  |
| Les actifs exposés embarquent une gestion des cookies conforme aux exigences définies par la CNIL ou autre réglementation applicables. |  |  |  |
| Les applicatifs exploités disposent d’un manifeste de gestion de version (changelog) permettant d’identifier les changements et impacts de chaque nouvelle version. |  |  |  |

Tableau : Table d'évaluation de l'architecture logicielle et des applicatifs

### **Services logiciels et** middlewares

|  |  |  |  |
| --- | --- | --- | --- |
| **Critère d’évaluation** | **Statut** | | **Commentaire** |
| La qualité du code source des composants développés est conforme au niveau de qualité attendu par les bonnes pratiques de l’industrie (respect philosophie SOLID, Injection de dépendance, manipulation des données API …). |  |  |  |
| Les composants développés embarquent des jeux de tests unitaires automatisés conforme aux exigences définies par l’entreprise. |  |  |  |
| Les composants développés utilisent un système de cache chaque fois que possible / nécessaire pour améliorer les performances du système. |  |  |  |
| Les traitement d’action à longue durée ou volumineuses sont réalisés en asynchrone. |  |  |  |
| Les erreurs renvoyés par les API REST sont conforme à la RFC 7807 (Problem Details for HTTP APIs) |  |  |  |
| Les informations de date ou heures circulants entre les applicatifs (via réponse API, DTO ou autre) sont conformes à la RFC 3339 (Date and Time on the Internet : Timestamps). |  |  |  |
| Les composants logiciels exploitées (développées ou consommés) sont conformes aux exigences non fonctionnelles définies pour le projet ou dans la bibliothèque d’exigences commune définie dans l’entreprise. |  |  |  |
| Les API Web développées sont conformes au format « REST » et conformes au niveau 2 du modèle de maturité de Richardson. |  |  |  |
| Les données renvoyés par les API Web sont au format JSON. |  |  |  |

Tableau : Table d'évaluation des services logiciels et middlewares (1/2)

|  |  |  |  |
| --- | --- | --- | --- |
| **Critère d’évaluation** | **Statut** | **Commentaire** | **Critère d’évaluation** |
| Les librairies utilisées dans les composants logiciels sont à jours. Un gestionnaire de paquet est utilisé (front ou back). |  |  |  |
| Le code source des applicatifs développés est disponible sur un contrôleur de code source et toujours accessible au client final (SCS Magasine). |  |  |  |
| Les composants applicatifs sont capables de gérer « proprement » (sans blocage) les erreurs de type « timeout » après une limite de temps définies pour chaque logiciel lorsqu’une ressource externe devient indisponible. |  |  |  |
| Les applicatifs développées sont tous *« stateless »* et ne stocke jamais de données essentielles au bon fonctionnement en local (conforme pour amélioration des performances via cache local). |  |  |  |
| La documentation du code source est compatible avec les standards de l’industrie (JavaDoc, TypeDoc …) |  |  |  |
| Les applicatifs développées (back-end) utilisent le langage Java et le Framework Spring (sauf incompatibilités / justification contraire). |  |  |  |

Tableau : Table d'évaluation des services logiciels et middlewares (2/2)

### Sécurité

|  |  |  |
| --- | --- | --- |
| **Critère d’évaluation** | **Statut** | |
| Les applicatifs exploités ne présente aucune faille de sécurité mentionné dans l’OWAPS top 10. |  |  |
| Les informations sur la plateforme utilisée (les logiciels, versions …) par le système sont anonymisées et ne peuvent pas être devinées depuis l’externe. |  |  |
| Les flux de données échangés sont tous basé sur un protocole embarquant un chiffrement TLS. Les flux http reposent tous sur le protocole HTTPS (y compris en interne). |  |  |
| Les échanges entre les composants front-end ↔︎ back-end sont sécurisés à l’aide de jeton CSRF. |  |  |
| Les API Web embarque un mécanisme de contrôle de l’authentification (niveau utilisateur ou serveur) sécurisé (exemple : Pas d’API Key JS). |  |  |
| Les API Web embarque un mécanisme de contrôle d’accès permettant d’accorder l’accès au seul périmètre de données accessible (niveau utilisateur ou serveur). |  |  |
| Les applicatifs embarque en entrée des mécanismes de nettoyage des données saisie par l’utilisateur et de contrôle de la conformité. (Optionnel en sortie, mais fortement recommandé). |  |  |
| Les formulaires de données vulnérables (non authentifié et exposé sur le web) utilisent une protection par CAPTCHA ou tout autre protection permettant de les protéger contre les risques de SPAM. |  |  |
| Les environnement hors production (développement, recette …) exposant la plateforme ne sont pas accessible au public ou exposés sur internet. |  |  |
| Les applicatifs ou la plateforme technique (cloud) ne contiennent aucun accès administrateur non identifié ou non justifié par un besoin clairement exprimé. |  |  |
| Les actifs exposés sur le web sont tous soumis à un scan régulier par un outil de détection des vulnérabilités (Qualys, Nexpose …). |  |  |
| Tous les actifs ont été soumis à un audit de sécurité (interne ou externe) avant déploiement. |  |  |
| Une authentification 2FA est exigée pour tous les accès disposant de privilège élevés permettant de manipuler les données de plusieurs utilisateurs ou les droits d’accès à la plateforme.  (Le 2FA est fortement recommandé pour tous les accès mais non imposé). |  |  |
| Les actifs web sont exposés derrière un Web Application Firewall permettant le contrôle des requêtes et la protection contre les attaques DoS. |  |  |
| Il n’existe pas d’accès externes à la solution (prestataires, compte de démo, de test …) non maitrisés et non identifié. |  |  |
| Les mots de passes des utilisateurs doivent satisfaire aux exigences de l’entreprise, ou, à minima aux bonnes pratiques de sécurité recommandés. (12 caractères de 3 catégories différentes). |  |  |

Tableau : Table d'évaluation de la sécurité

### Gestion de l’information

|  |  |  |
| --- | --- | --- |
| **Critère d’évaluation** | **Statut** | |
| Tout accès aux données du système est contrôlé et sécurisé (fonctionnement et techniquement). |  |  |
| Pour chaque actif logiciel, les données à caractère personnel (DCP) manipulés sont clairement identifiés et protégées. |  |  |
| Pour chaque actif logiciel, les données sensibles (mot de passe, numéros CB …) manipulés sont clairement identifiés. |  |  |
| Les données sensibles sont toujours chiffrées ou « hachés » (selon le besoin) sur le support de stockage. Le chiffrement est effectué à l’aide de l’algorithme AES-256 bits minimum. |  |  |
| Les données sont manipulées et conservées conformément au RGPD ou tout autre réglementation applicable (y compris pour les délais de conservation). |  |  |
| Un délégué à la protection des données (DPO) est désigné pour garantir la conformité de la protection des données. |  |  |
| Les modifications de données sont transactionnés par groupe chaque fois que nécessaire (transaction SQL, transaction distribuées ..) pour garantir la cohérence du référentiel de données. |  |  |
| La conception des bases de données est confiée à un expert assurant la définition des données, l’élimination des redondances et garantissant une information fiable dans le temps et à toujours jour. |  |  |
| Un temps de perte de données acceptable (RPO) est défini et validé par l’ensemble des parties prenantes, y compris au niveau technique (crash-test total du système réalisé et conforme). |  |  |

Tableau : Table d'évaluation de la gestion de l'information

### Gestion des systèmes informatiques

|  |  |  |
| --- | --- | --- |
| **Critère d’évaluation** | **Statut** | |
| La plateforme est disponible sur un environnement sécurité de développement et de test (n’exploitant pas les données de production).  Des environnements supplémentaires (préproduction etc..) peuvent être existanter mais ne sont pas imposés. |  |  |
| La plateforme dispose d’une Documentation de l’architecture technique (DAT) exploitable par les administrateurs (modélisation de l’architecture, informations liés aux caractéristiques de la plateforme, échanges de flux …) |  |  |
| La plateforme et les systèmes associés sont intégrés aux outils de monitoring de l’entreprise. Des alertes sont définies selon et classifiées un système de priorité. |  |  |
| Les erreurs & exceptions de la plateforme et des systèmes sont correctement “loggées” dans un système centralisé et sécurisé (type Graylog). |  |  |
| Les logs embarquent un mécanisme garantissant l’absence de données sensibles (ex : mot de passe). |  |  |
| Les déploiements des logiciels et de leurs mises à jour sont automatisés. (Peut nécessiter une validation manuelle avant déclenchement des pipelines). |  |  |
| Les versions des logiciels utilisés et de leurs dépendances sont à jour. |  |  |
| Une politique de mise à jour des actifs IT est définie au niveau de l’entreprise. |  |  |
| Un Plan de Continuité de l’Activité (PCA) est défini au sein de l’entreprise et mis à jour avec la nouvelle plateforme. |  |  |
| Les fournisseurs d’infrastructure retenus garantissent un SLA de 99,9% à minima. (99.99% souhaité). |  |  |

Tableau : Table d'évaluation de la gestion des systèmes informatiques

### Gestion de projet et outillage

|  |  |  |
| --- | --- | --- |
| **Critère d’évaluation** | **Statut** | |
| Les artefacts d’architecture du projet sont intégrés au référentiel d’architecture de l’entreprise. |  |  |
| Les développements sont réalisés sur un environnement de développement.  Les tests logiciels sont réalisés sur un environnement différent de l’environnement de développement et différent de la production (test, recette, qualification …). |  |  |
| Des outils de d’ingénierie inversé (reverse engineering) sont utilisé pour l’analyse des failles logiciels. |  |  |
| Une matrice RACI est établie pour le projet et désigne un responsable (Accountable - A) unique pour chaque item. |  |  |
| Un plan de test est défini et prévoit une méthodologie permettant de garantir la conformité de la livraison aux exigences définies pour le projet. |  |  |
| Les parties prenantes et leurs rôles sont clairement identifiés. |  |  |
| Un outils de gestion de projet est défini et accessible à l’ensemble de l’équipe en charge du projet (Architectes, coordinateur projet, sponsors …) |  |  |
| Les schéma, diagrammes et autres artefacts produits s’appuie sur une norme ou un langage standardisé (UML, SysML, Archimate, BPMN …). |  |  |
| L’ensemble des systèmes et logiciels tiers installés dans la solutions disposent d’un contrat de support. |  |  |
| Les risques du projet sont identifiés dans une matrice, classifiés et suivi. Des mesures d’atténuations sont prévues pour les risques critique (à minima). |  |  |
| Les impacts du projets sont identifiés et maitrisés (sur les logiciels existants, les acteurs, les processus business …) |  |  |
| Le contrôleur de code source dispose de pipeline permettant les opérations CI/CD. Des métriques de qualité de code sont disponible dans un outil d’analyse statique (ex : Sonarqube). |  |  |

Tableau : Table d'évaluation de la gestion projet et de l'outillage

# PROCESSUS DE RÉVISION DE LA CONFORMITÉ

# APPROBATIONS

## Approbation du document

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Poste** | **Signature** | **Date** |
| **David EVAN** | Architecte Logiciel | David Evan | 24/06/2022 |
| **Jannette BOND** | CEO | *[En attente]* | *[En attente]* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tableau : Liste des parties prenantes approuvant le présent document

# TABLES DES RÉFÉRENCES

## Figures

[Figure 1 : Vision de l'architecture cible du projet "SCS GED". 6](#_Toc106973032)

[Figure 2 : Diagramme d'implémentation de l'architecture du projet "SCS GED" 7](#_Toc106973033)

## Tableaux

[Tableau 1 - Historique des révisions 2](#_Toc106973023)

[Tableau 2 : Table d'évaluation de l'architecture technique 8](#_Toc106973024)

[Tableau 3 : Table d'évaluation de l'architecture logicielle et des applicatifs 9](#_Toc106973025)

[Tableau 4 : Table d'évaluation des services logiciels et middlewares 10](#_Toc106973026)

[Tableau 5 : Table d'évaluation de la sécurité 11](#_Toc106973027)

[Tableau 6 : Table d'évaluation de la gestion de l'information 12](#_Toc106973028)

[Tableau 7 : Table d'évaluation de la gestion des systèmes informatiques 13](#_Toc106973029)

[Tableau 8 : Table d'évaluation de la gestion projet et de l'outillage 14](#_Toc106973030)

[Tableau 9 : Liste des parties prenantes approuvant le présent document 16](#_Toc106973031)