

Spécifications techniques

Solution Building Blocs

Projet : GED Recherche Scientifique

Client : SCS magazine



Table des matières

1	Objet de ce document.....	5
2	Blocs de construction.....	5
2.1	Fonctionnalité spécifique.....	5
2.2	Attributs.....	5
2.2.1	Sécurité.....	5
2.2.2	Gestion.....	5
2.2.3	Localisation.....	5
2.2.4	Évolutivité.....	5
2.3	Performances.....	6
2.4	Configurations.....	6
2.5	Pilotes et contraintes.....	6
3	Interfaces.....	7
3.1	Aperçu.....	7
3.2	Interopérabilité.....	8
3.3	Blocs de construction dépendants.....	8
3.3.1	Croisement des fonctionnalités du rôle <i>input</i> avec les applications.....	8
3.3.2	Croisement des fonctionnalités du rôle <i>review</i> avec les applications.....	9
3.3.3	Croisement des fonctionnalités du rôle <i>admin</i> avec les applications.....	9
3.4	Éléments constitutifs requis pour la solution.....	10
3.4.1	Couche interfaces.....	10
3.4.1.1	Point d'entrée du GED de SCS Magazine.....	10
3.4.1.2	Gestionnaire d'utilisateurs.....	11
3.4.1.3	Gestionnaire de documents.....	12
3.4.1.4	Gestionnaire de commentaire.....	13
3.4.2	API Gateway.....	14
3.4.3	Couche services.....	15
3.4.3.1	Gestion des utilisateurs.....	15
3.4.3.2	Gestion des documents.....	16
3.4.3.3	Gestion des commentaires.....	17
3.4.4	Couche données.....	18
3.4.4.1	Répertoire de données.....	18
4	Cartographie.....	19
4.1	Cartographie SBB.....	19



Spécifications techniques
GED Recherche Scientifique
Client : SCS magazine

Préparé par :
Rudy HOARAU
Architecte Logiciel

4.2 Relations entre SBB et ABB.....	20
Pour rappel, le ABB issue du document d'architecture technique est le suivant :.....	20



Information sur le document

Nom du projet :	GED Recherche Scientifique		
Préparé par :	Rudy Hoarau	Version du document :	1.0
Titre :	Architecture Roadmap	Date de la version :	25/07/2022
Revu par :		Date de revu :	

Liste de distribution

De	Date	Téléphone/Fax/Mail
Rudy Hoarau	25/07/2022	-

À	Action*	Date d'échéance	Téléphone/Fax/Mail
Nicolas	Approuver	01/08/2022	-
Client	Informé	08/08/2022	-

* Types d'action : Approuver, Réviser, Informer, Classer, Action requise, Assister à la réunion, Autre (veuillez préciser)

Historique de version du document

Version	Date	Revu par	Description	Nom du fichier
1.0	25/07/2022	Rudy H	Édition et première version	xx_specification_xx.odt



1 Objet de ce document

Ce document présente les blocs de construction de solutions (SBB) qui se rapporte au projet de création d'un gestionnaire électronique des documents (GED) pour SCS Magazine. Il est composé des solutions qui peuvent être achetés ou développés.

Aussi il présente :

- la définition des produits et composants qui implémenteront la fonctionnalité,
- la définition de la mise en œuvre,
- la réponse aux exigences de l'entreprise,
- et enfin, la solution du client SCS Magazine.

2 Blocs de construction

2.1 Fonctionnalité spécifique

Les fonctionnalités voulues pour le gestionnaire sont présentées dans le cahier des charges.

2.2 Attributs

2.2.1 Sécurité

Tous utilisateurs voulant se connecter aux systèmes devra être répertorié dans la base de données de l'application gestion des utilisateurs.

2.2.2 Gestion

Les administrateurs auront accès à l'entièreté de la plateforme.

Une équipe sera dédiée à la maintenance du système.

2.2.3 Localisation

Le système sera hébergé chez le client.

2.2.4 Évolutivité

Le site est web est une architecture micro frontend, cela implique des possibilités d'évolution par :

- la flexibilité en modification du front,
- l'ajout et la modification de micros applications,



- et enfin, l'ajout ou la modification des microservices associés.

2.3 Performances

Les performances de la couche service seront gérés au travers de la plateforme PaaS.

Les performances de la couche donnée seront gérés par le client.

2.4 Configurations

Une connaissance des configurations des web applications dans le cloud est suggérée.

Plusieurs environnements de développement sont à mettre en place, dont :

ID	ENV.	Description
1	DEV	Local au développeur
2	VCTRL	Serveur de contrôle de version (GIT)
3	CI	Serveur de Build et test automatique
4	CERT	Serveur associé à un contrôleur de livraison (Sigma) pour les tests de qualifications
5	PROD	Serveur associé à un contrôleur de livraison (Sigma) pour la mise en production

Dans le cadre de la première itération de la feuille route (PoC), le serveur de production (id 5) n'est pas obligatoire pour le moment.

2.5 Pilotes et contraintes

Les contraintes fonctionnelles sont présentes dans le cahier des charges.

Les contraintes techniques sont présentes dans le document d'architecture technique.

Au sujet des pilotes, il est à noter que le datacenter de l'entreprise possède déjà sont propres système de stockage. Il est donc possible que certain composant de ce système pourrait être réutilisé dans le cadre de projet, afin de réduire les temps de développement. De ce fait, il est recommandé d'auditer ce système afin d'évaluer sa compatibilité avec la nouvelle architecture et notamment le mécanisme de *versioning*.

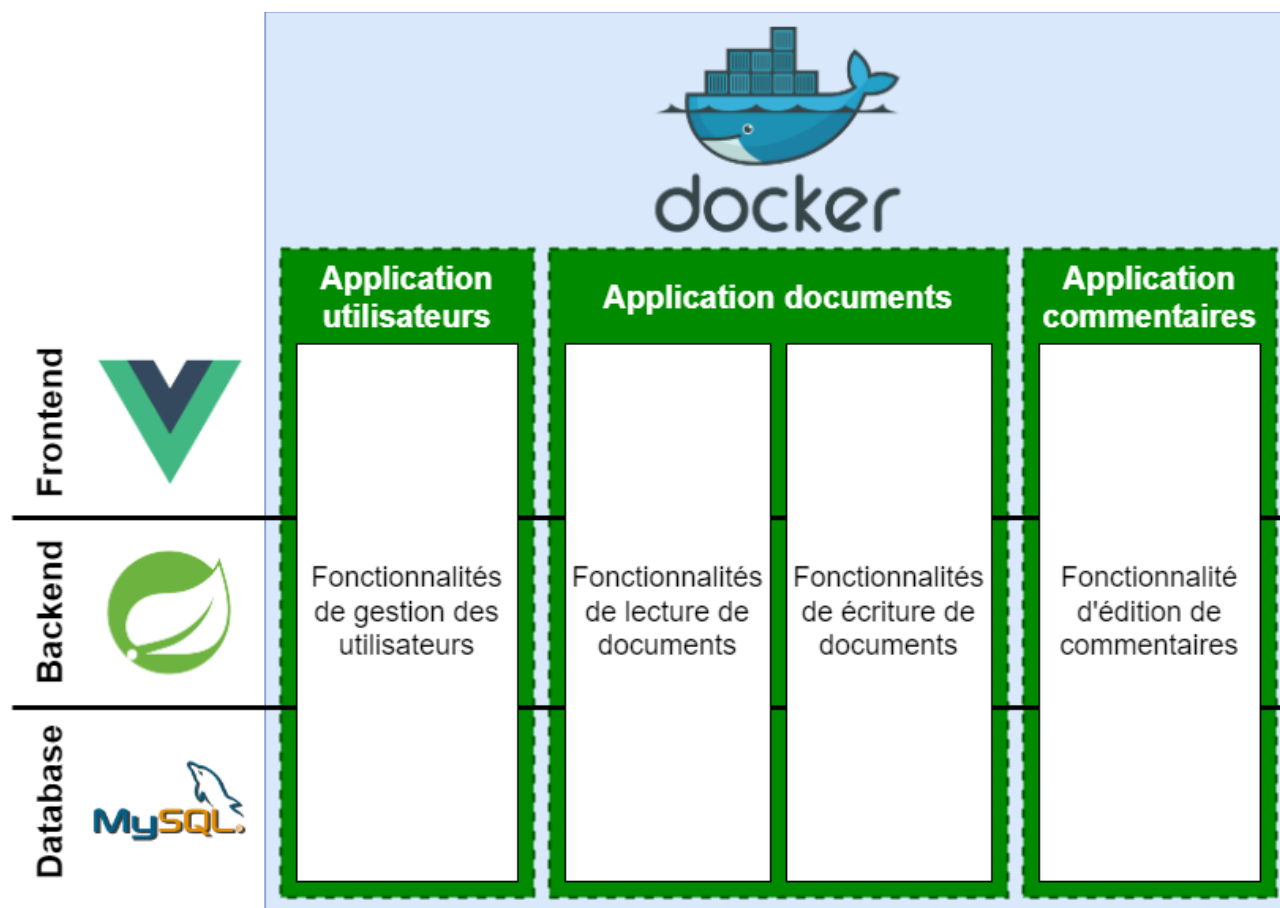


3 Interfaces

3.1 Aperçu

Comme décrit dans le document d'architecture, le produit reposera sur une implémentation Java EE pour les couches 'Service Layer' et 'Data Layer'. Le Front Layer doit se reposer sur un framework adaptable à une architecture micro-frontend.

Le cadre architectural définit également la nécessité de déployer les composants du produit à travers des conteneurs.



Ainsi, comme décrit dans l'image ci-dessus, chaque application sera déployé dans des conteneurs Docker. Elles seront toutes composées de :

- frontend : Vue. JS
- backend : Spring
- database : MySql

Aussi, il est comprendre que les applications seront implémentées par les équipes de

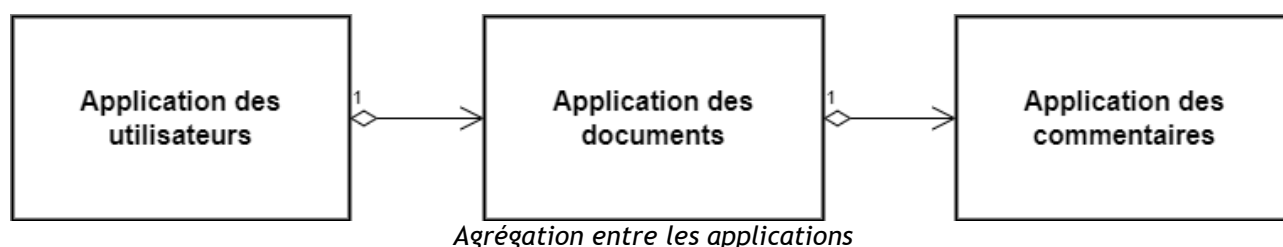


développeurs. Ici, jusqu'à trois équipes peuvent être mises en place pour répondre aux prérogatives du projet, soit développer le front, le back et database de chaque micro-frontend de bout en bout :

- L'interface de gestion des utilisateurs
- L'interface de gestion des documents
- L'interface de gestion de commentaires

En fonction de son rôle (input, review et admin) dans le système chaque utilisateur aura accès aux fonctionnalités comme prévu dans le cahier des charges.

3.2 Interopérabilité



Ainsi, un utilisateur est associable à plusieurs documents, et ce même utilisateur peut éditer plusieurs commentaires dans un seul document.

3.3 Blocs de construction dépendants

3.3.1 Croisement des fonctionnalités du rôle *input* avec les applications

Fonctionnalités du rôle « input »	Gestion des utilisateurs	Gestion des documents		Gestion des commentaires
		Lecture	Écriture	
I.1 Je peux me connecter sur la plateforme.	✓			
I.2 Je peux demander une réinitialisation de mon mot de passe.	✓			
I.3 Je peux visualiser ma liste de documents et trier par état.		✓		
I.4 Je peux créer un nouveau document.			✓	
I.5 Je peux modifier un document.			✓	
I.5 Je peux indiquer que mon document doit être validé.		✓		
I.6 Je peux être notifié (par mail) à chaque modification d'un de mes documents "en cours de revue".			✓	



Fonctionnalités du rôle « input »	Gestion des utilisateurs	Gestion des documents		Gestion des commentaires
		Lecture	Écriture	
I.7 Je peux être notifié (par mail) à chaque changement d'état du document.			✓	
I.8 Je peux ajouter des commentaires à mon document et répondre à des commentaires.				✓
I.9 Je peux indiquer la version de mon document.		✓		
I.10 Je peux visualiser une version antérieure de mon document			✓	

3.3.2 Croisement des fonctionnalités du rôle *review* avec les applications

Fonctionnalités du rôle « review »	Gestion des utilisateurs	Gestion des documents		Gestion des commentaires
		Lecture	Écriture	
R.1 Je peux visualiser la liste des documents à évaluer (état : en cours de revue).		✓		
R.2 Je peux ajouter et répondre à des commentaires.				✓
R.3 Je peux indiquer une version sur le document et visualiser les versions antérieures.		✓		
R.4 Je peux indiquer que le document est prêt pour publication (état clôturé).		✓		
R.5 Je peux télécharger un document au format pdf.		✓		

3.3.3 Croisement des fonctionnalités du rôle *admin* avec les applications

Fonctionnalités du rôle « admin »	Gestion des utilisateurs	Gestion des documents		Gestion des commentaires
		Lecture	Écriture	
A.1 Je peux créer un compte pour un nouveau collaborateur.	✓			
A.2 Je peux supprimer le compte d'un collaborateur.	✓			
A.3 Je peux donner l'accès d'un ou plusieurs documents à un utilisateur.	✓			



3.4 Éléments constitutifs requis pour la solution

3.4.1 Couche interfaces

3.4.1.1 Point d'entrée du GED de SCS Magazine

Nom du bloc de construction :

fr.scsmagazine.main

Fonctionnalité fournie :

Site web principal du système cible, il contient les fonctionnalités d'authentification des utilisateurs. Une fois authentifié les utilisateurs, seul leur rôle, ont accès aux trois micro-applications : gestion des utilisateurs, des documents et des commentaires.

Lien vers des exemples d'implémentation ou d'interfaces :

<https://micro-frontends.org/>

Travail exceptionnel pour compléter ce bloc de construction :

Afin de réaliser ce bloc, les équipes de développement devront mettre en place un serveur client léger. Des technologies comme Spring comblera ce besoin.

Le point d'entrée du site sera une page d'authentification, une fois authentifié, l'utilisateur courant visualise, en fonctions de son rôle, soit la page web du gestionnaire d'utilisateur ou soit la page de web du gestionnaire de document.

Alignement architectural :

1 Objectif Réaliser un PoC du point d'entrée du GED

Principe

La démonstration de faisabilité montre la faisabilité d'un procédé. Aussi, en sécurité informatique, une preuve de concept est la démonstration de l'existence d'une faille logicielle par un programme qui la met en évidence.

2 Objectif Proposer une méthode d'authentification robuste.

Principe

L'accès aux ressources d'un système d'information par une entité se décompose en trois sous-processus, l'authentification, l'identification et le contrôle d'accès.

3 Objectif Réaliser un site web interactif

Principe

La notion de web réactif repense la manière de concevoir les parcours de navigation sur Internet, puisqu'il ne s'agit plus de concevoir autant de parcours qu'il y a de familles de terminaux mais de concevoir une interface réactive. Ainsi, les sources d'information et les socles techniques ne sont pas dupliqués. Ce qui engendre des économies d'échelle dans la conception et la maintenance de sites.

Bonne pratique : réaliser la version navigateur mobile avant celle sur ordinateur.



3.4.1.2 Gestionnaire d'utilisateurs

Nom du bloc de construction :

fr.scsmagazine.usermanagement

Fonctionnalité fournie :

Micro application de gestion des utilisateurs. Ce bloc est constitué d'une ou plusieurs IHM permettant aux administrateurs de réaliser la gestion des utilisateurs.

Comme décrit dans les premiers chapitre de ce documents, l'administrateur pourra CRUD les utilisateurs, associés des documents et donner des droits en éditions de commentaires.

Lien vers des exemples d'implémentation ou d'interfaces :

<https://micro-frontends.org/>

Travail exceptionnel pour compléter ce bloc de construction :

Ce bloc est composé d'une vue contenant la liste des utilisateurs. Sur cette même vue, il sera possible d'ajouter, supprimer des utilisateurs.

Une autre vue permettra d'éditer ou de modifier un utilisateur sélectionné. Aussi, cette vue permettra d'associer l'utilisateur en cours d'édition à des documents. Pour chaque document auquel l'utilisateur sera associé, il sera possible de lui donner des droits en lecture/écriture de commentaire.

Alignement architectural :

1	Objectif	Réaliser un PoC du gestionnaire des utilisateurs.
	Principe	La démonstration de faisabilité montre la faisabilité d'un procédé. Aussi, en sécurité informatique, une preuve de concept est la démonstration de l'existence d'une faille logicielle par un programme qui la met en évidence.
2	Objectif	Proposer pattern simple et facile d'usage permettant l'édition d'utilisateur.
	Principe	Les utilisateurs doivent être à l'aise dans le site, utiliser des designs patterns afin de structurer vos composants de manière standardisée.
3	Objectif	Le gestionnaire d'utilisateur est une micro-applications intégré au site principal.
	Principe	L'architecture micro-frontend est une stratégie dans laquelle la base de code frontend traditionnellement monolithique est divisée en applications plus petites, s'adressant idéalement aux différents domaines d'activité. Tous ces petits éléments forment une interface frontale transparente qui offre une expérience utilisateur de premier ordre et est facile à modifier et à mettre à l'échelle.



3.4.1.3 Gestionnaire de documents

Nom du bloc de construction :

fr.scsmagazine.ged

Fonctionnalité fournie :

L'IHM de la gestion de document est un micro-applications intégré au site web principal. Une fois qu'un utilisateur rédacteur ou validateur s'y connecte, il aura accès à tous les documents qui lui sont associés. En fonction de son rôle cet utilisateur pourra lire ou écrire des documents scientifiques.

Lien vers des exemples d'implémentation ou d'interfaces :

Comportement proche de celui d'un répertoire de fichier partagés comme Filezilla.

Travail exceptionnel pour compléter ce bloc de construction :

Proposer une vue exposant la liste de document accessible par l'utilisateur courant (selon son rôle). Lorsqu'il clique sur un document, l'utilisateur aura la possibilité d'accéder au contenu du document, soit en édition ou soit en lecture comme sur un répertoire partagé.

Alignement architectural :

1	Objectif	Réaliser un PoC du gestionnaire des utilisateurs.
	Principe	La démonstration de faisabilité montre la faisabilité d'un procédé. Aussi, en sécurité informatique, une preuve de concept est la démonstration de l'existence d'une faille logicielle par un programme qui la met en évidence.
2	Objectif	Le gestionnaire de document est une micro-applications intégré au site principal.
	Principe	L'architecture micro-frontend est une stratégie dans laquelle la base de code frontend traditionnellement monolithique est divisée en applications plus petites, s'adressant idéalement aux différents domaines d'activité. Tous ces petits éléments forment une interface frontale transparente qui offre une expérience utilisateur de premier ordre et est facile à modifier et à mettre à l'échelle.



3.4.1.4 Gestionnaire de commentaire

Nom du bloc de construction :

fr.scsmagazine.commentmanager

Fonctionnalité fournie :

Ce bloc est une micro-applications intégré à la vue d'un document en cours de surlignage. Il sera possible pour un utilisateur ayant les droits de CRUD des commentaires sur des éléments en cours d'observation.

Lien vers des exemples d'implémentation ou d'interfaces :

Visualisation des commentaires en cascade identique à celle de [Reddit](#).

Travail exceptionnel pour compléter ce bloc de construction :

Intégrer un bandeau à droit de la visualisation unitaire d'un document. Ce bandeau contiendra l'éditeur de commentaire. Il sera possible depuis le document de surligner un élément et d'y ajouter un commentaire qui sera directement visible à droit de la page web (dans le bandeau). Les commentaires ajoutés peuvent être modifiés ou supprimer par l'utilisateur qui les a édités.

Alignement architectural :

1	Objectif	Réaliser un PoC du gestionnaire des utilisateurs.
	Principe	La démonstration de faisabilité montre la faisabilité d'un procédé. Aussi, en sécurité informatique, une preuve de concept est la démonstration de l'existence d'une faille logicielle par un programme qui la met en évidence.
2	Objectif	Le gestionnaire de commentaires est une micro-applications intégré au site principal.
	Principe	L'architecture micro-frontend est une stratégie dans laquelle la base de code frontend traditionnellement monolithique est divisée en applications plus petites, s'adressant idéalement aux différents domaines d'activité. Tous ces petits éléments forment une interface frontale transparente qui offre une expérience utilisateur de premier ordre et est facile à modifier et à mettre à l'échelle.



3.4.2 API Gateway

Nom du bloc de construction :

fr.scsmagazine.apiGateway

Fonctionnalité fournie :

La passerelle d'API est un des éléments du système de gestion d'API. La passerelle d'API intercepte toutes les requêtes entrantes et les transmet au système de gestion d'API, qui se charge de nombreuses fonctions essentielles.

Les actions d'une passerelle d'API varient d'une mise en œuvre à l'autre. Parmi les fonctions communes figurent l'authentification, le routage, le bridage du débit, la facturation, la surveillance, l'analyse, les politiques, les alertes et la sécurité.

Lien vers des exemples d'implémentation ou d'interfaces :

[Amazon API Gateway - Guide du développeur](#)
Spring

Travail exceptionnel pour compléter ce bloc de construction :

1. Créer une fonction Lambda
2. Créer l'API HTTP
3. Tester l'API

Alignement architectural :

1 Objectif Simplifier et de stabiliser les interfaces

Principe

- Sécuriser l'accès à une API
- Limiter la sur-sollicitation d'un service grâce à l'API Gateway
- Surveiller les logs des API
- Transformer les requêtes de l'API

Source : API Gateway : [le guide définitif par Talend](#)

2 Objectif Centraliser les différentes tâches transverses

Principe

Permettre aux développeurs de gagner du temps côté service back end, puisque ceux-ci ont meilleur intérêt de se concentrer sur le développement de la logique métier du côté des services back end, plutôt que se concentrer à nouveau sur le développement de l'authentification de chaque service par exemple.

3 Objectif Développer une approche DevSecOps

Principe

La sécurité est la responsabilité commune entre les différents intervenants (développeurs, programmeur, ingénieurs...) d'une entreprise, où culture de confiance, collaboration et conduite du changement sont indispensables.



3.4.3 Couche services

3.4.3.1 Gestion des utilisateurs

Nom du bloc de construction :

fr.scsmagazine.back.userManagement

Fonctionnalité fournie :

Service backend répondant au besoin de gestion des utilisateurs de la GED. Ce service est composé de des méthodes de gestion des utilisateurs : authentification, accès aux données personnelles des utilisateurs, règle de gestion des utilisateurs. Parmi les règles de gestion ont retrouve celles associées à l'accès aux documents et à l'édition de commentaire.

Aussi sur le plan légal ce service répondra aux exigences de gestion des utilisateurs décrit par la RGPD.

Lien vers des exemples d'implémentation ou d'interfaces :

Il existe aussi des gestionnaires d'utilisateurs comme [KeyClock](#) qui peuvent répondre aux exigences de ce bloc.

Travail exceptionnel pour compléter ce bloc de construction :

1. Ajout des fonctionnalités de la gestion des utilisateurs :
 1. ajout
 2. modification
 3. suppression
 4. affectation à un groupe
2. Ajout des règles de sécurité
 1. Cryptage des données de bout en bout
 2. Règles de sécurité associé à l'authentification
3. La gestion des groupes documents et commentaires :
 1. ajout
 2. modification
 3. suppression

Alignement architectural :

1 Objectif Création d'un système d'authentification sécurisé

Principe

- Les organisations deviennent de plus en plus vulnérables aux brèches de données dues à la corruption d'identités.
- Les procédures de gestion des identités inefficaces augmentent les frais associés à l'administration informatique.
- La lassitude liée aux mots de passe et les opérations de réinitialisation de mot de passe font chuter la productivité des utilisateurs.
- Le manque de visibilité sur les accès au cloud ralentit le processus de conformité aux réglementations.

2 Objectif Implémenter la gestion des utilisateurs

Principe

Répondre aux exigences du client pour réalisation du module de gestion.
Les utilisateurs ont des rôles précis dans le système.



3.4.3.2 Gestion des documents

Nom du bloc de construction :

fr.scsmagazine.back.documentManagement

Fonctionnalité fournie :

Ce service répond aux exigences de gestion des documents. Appelé par le front, ce service renvoie les documents. Des méthodes permettent aussi de créer, modifier et supprimer des documents. Il pourra aussi être appelé par le gestionnaire d'utilisateur afin de fournir les listes des documents. Durant les processus d'ingestion d'éléments de documentation, ce service appellera le gestionnaire d'utilisateur afin de confirmer les droits en écriture des utilisateurs. Le backend de la gestion des documents inclura toutes les fonctionnalités d'écriture et de lecture.

Lien vers des exemples d'implémentation ou d'interfaces :

[Seafile](#)

Travail exceptionnel pour compléter ce bloc de construction :

Implémenter les méthodes relatives au besoin fonctionnel.
Interroger le backend de gestion des utilisateurs au sujet des droits d'accès à certaine fonctionnalité.

Alignement architectural :

1	Objectif	Implémenter la gestion des documents
	Principe	Répondre aux exigences du client pour réalisation du module de gestion.
2	Objectif	Rendre les fonctionnalités disponibles en fonction du rôle des utilisateurs
	Principe	Afin de prévenir l'accès par une tierce application au système, l'existence des utilisateurs sera aussi contrôlé dans la couche backend.
3	Objectif	Contrôler les changements de version des documents.
	Principe	Chaque fois qu'un utilisateur modifie un document, le système enregistrera automatiquement la date et l'heure de modification. Aussi, il sera proposé à l'utilisateur d'ajouter un incrément de version au document modifié.



3.4.3.3 Gestion des commentaires

Nom du bloc de construction :

fr.scsmagazine.back.commentManagement

Fonctionnalité fournie :

Ce service répond aux exigences de gestion des commentaires. Appelé par le front, ce service renvoie les commentaires associés à un document. Des méthodes permettent aussi de créer, modifier et supprimer des commentaires dans un document.

Il pourra aussi être appelé par le gestionnaire d'utilisateur afin de fournir les listes des documents. Durant les processus d'ingestion d'éléments de commentaire, ce service appellera le gestionnaire d'utilisateur afin de confirmer les droits en écriture des utilisateurs.

Lien vers des exemples d'implémentation ou d'interfaces :

Travail exceptionnel pour compléter ce bloc de construction :

Implémenter les méthodes relatives au besoin fonctionnel.

Interroger le backend de gestion des utilisateurs au sujet des droits d'accès à certaine fonctionnalité.

Alignement architectural :

1	Objectif	Implémenter la gestion des commentaires
	Principe	Répondre aux exigences du client pour réalisation du module de gestion.
2	Objectif	Rendre les fonctionnalités disponibles en fonction du rôle des utilisateurs
	Principe	Afin de prévenir l'accès par une tierce application au système, l'existence des utilisateurs sera aussi contrôlé dans la couche backend.



3.4.4 Couche données

3.4.4.1 Répertoire de données

Nom du bloc de construction :

fr.scsmagazine.data.repositoryAccess

Fonctionnalité fournie :

Le répertoire de données est le cœur de la GED, il contient les listes des utilisateurs, les documents ainsi que leurs commentaires associés.

- Un répertoire concatène l'accès aux données : si plusieurs sources de données fournissent des données à l'application, le répertoire a la responsabilité principale de les cacher aux couches supérieures pour faciliter l'accès à ces données, indépendamment de la provenance de ces données.
- Si l'application implémente une stratégie de mise en cache pour rendre les données disponibles pour différentes requêtes, le répertoire est la couche où ce mécanisme de mise en cache est implémenté.
- Un répertoire ne gère qu'un seul type de données (entité). Cela signifie que si un cas d'utilisation donné doit accéder aux données de trois entités différentes pour réussir, le cas d'utilisation dépend de trois référentiels différents : un par entité - objet métier qui doit interroger.

Lien vers des exemples d'implémentation ou d'interfaces :

[Repository pattern](#)

Travail exceptionnel pour compléter ce bloc de construction :

Possiblement administrer d'autre base de données afin réduire le scope de gestion du répertoire. Ainsi, les utilisateurs ne seraient pas dans la même base de données que celle-ci.

Alignement architectural :

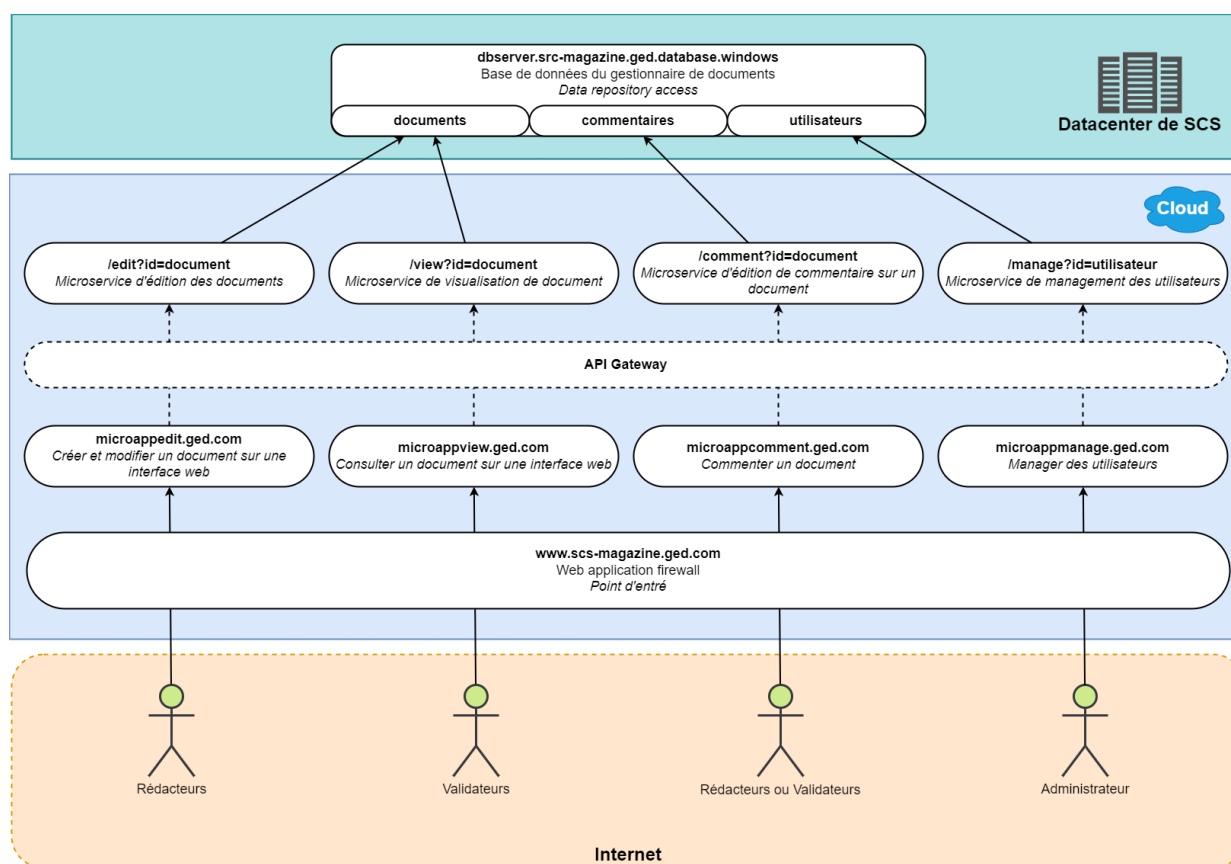
1	Objectif	Auditer les systèmes actuels afin d'évaluer la compatibilité matérielle ce bloc.
	Principe	Green IT
2	Objectif	Implémenter ce bloc
	Principe	Ici l'entité est le document. Les commentaires y seront associés au travers d'une autre table de données.



4 Cartographie

4.1 Cartographie SBB

Le schéma suivant présente la cartographie SBB (cliquer sur l'image pour être renvoyé à l'annexe correspondante).



Afin de limiter les investissements initiaux du client sur le projet, une solution PaaS est recommandé pour la mise en place du projet.

Aussi, le prestataire de solution cloud assurera les problématiques de stockage et de performance du système. Cela devrait permettre un gain de temps en maintenance des services.

Les produits du cloud assure une flexibilité et une évolutivité de l'infrastructure et donc de l'architecture. Si les besoins du client évoluent, il sera possible, grâce aux solutions cloud, d'adapter l'offre rapidement et simplement.

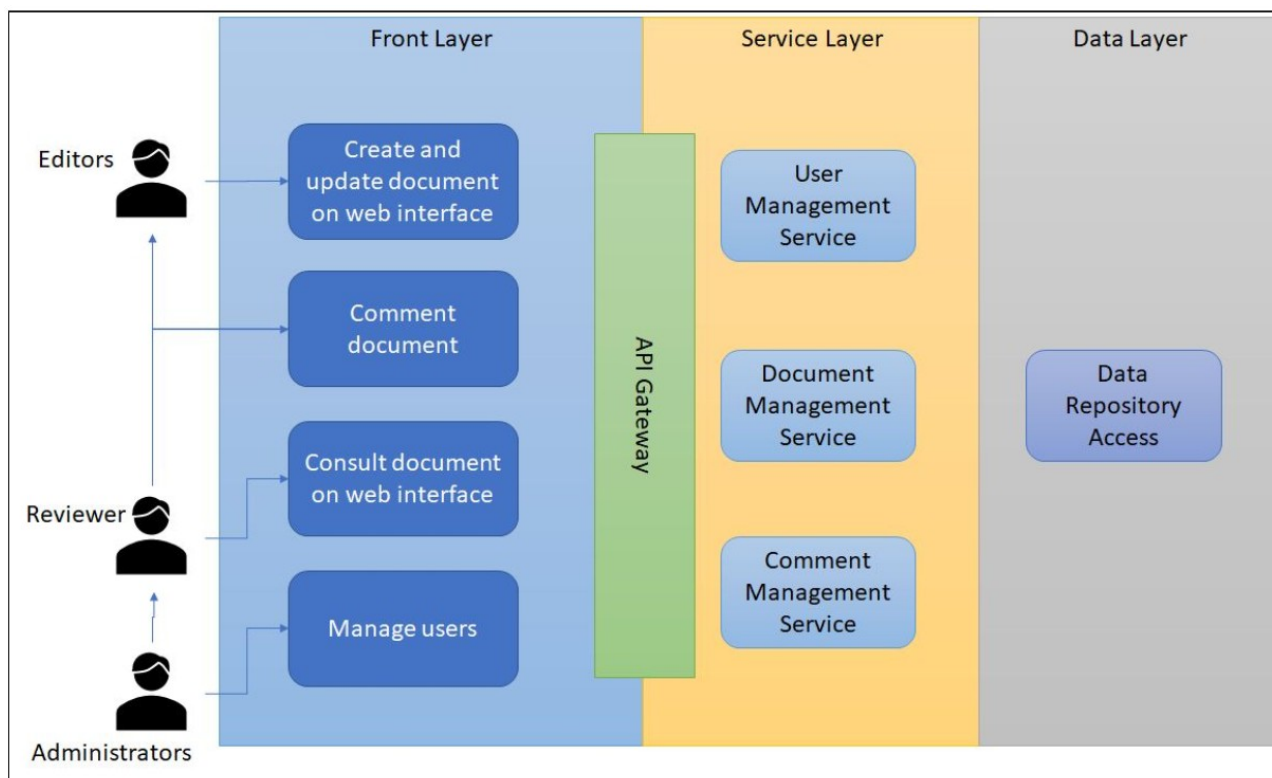
Pour palier à certain problème des solutions cloud, le client hébergera chez lui la couche donnée du nouveau système. Aussi, la sauvegarde des données chez le client est privilégié aux vues de la protection intellectuel et de l'activité de l'entreprise.

Une étude financière permettra de confirmer ce modèle.



4.2 Relations entre SBB et ABB

Pour rappel, le ABB issue du document d'architecture technique est le suivant :



Par rapport au ABB, les micros applications visualisation et d'édition de document sont des agrégations du service de gestion de document.



A1 – SBB du GED de recherche scientifique

