

DAT

CRISTAL

REFONTE

VERSION DU

12/05/2025

AUTEURS

Boi Chanh HUYNH + Scalian
Olivier ROUSVAL
Christophe VALADE

REFERENCE

\\ad.0efg\partages\SI\Pôle_SIM\Unité_archi\DAT\Cristal\ DAT_Cristal-
Refonte__ABM.docx

APPROBATION

Instance	Date	Remarques
----------	------	-----------

HISTORIQUE DES VERSIONS

Version	Date	Auteur(s)	Modifications
1.0	08/11/2024	B. HUYNH, Scalian	Version initiale, intégration des éléments livrés par Scalian, validés par BHU.
1.0.1	27/12/2024	P.JARD	Description fonctionnelle + typologie des acteurs
1.1	12/05/2025	C. VALADE	Prise en compte remarques O. ROUSVAL
1.2			

1. TABLE DES MATIERES

1	INTRODUCTION.....	5
1.6.	RACI : qui fait quoi ?.....	5
1.7.	Documents de référence.....	5
2	PRESENTATION FONCTIONNELLE DE LA SOLUTION.....	6
2.2.	Principales fonctionnalités de l'application.....	7
2.3.	Schéma de l'architecture fonctionnelle.....	7
2.4.	Acteurs et utilisateurs.....	8
2.5.	Cycle de vie des données et volumétrie.....	10
2.6.	Exigences et contraintes particulières exprimées par le métier.....	11
3	SECURITE.....	12
2.7.	Organisation de la sécurité autour du SI.....	12
	Couche réseau.....	12
	Couche système d'exploitation (OS).....	12
	Couche de virtualisation (si présente).....	12
	Couche middleware.....	12
	Couche applicative.....	12
	Couche de gestion des données.....	13
	Couche de sécurité.....	13
	Couche utilisateur (Interface utilisateur).....	13
	Couche de gestion et de monitoring.....	13
	Couche de gouvernance, conformité et gestion des risques.....	13
2.8.	RGPD et données sensibles.....	14
	Principe de sensibilité des données.....	14
	Principe du "besoin de savoir" (Need to Know).....	14
	Principe de la minimisation des données.....	14
	Principe de l'intégrité des données.....	14
	Principe de la séparation des données sensibles.....	14
	Principes de gestion du cycle de vie des données.....	14
	Mise en œuvre pratique de la classification des données.....	14
2.9.	Disponibilité.....	15
2.10.	Intégrité.....	15
2.11.	Confidentialité.....	15
2.12.	Traçabilité et piste d'audit.....	16
2.13.	Gestion des incidents.....	17
	Couche réseau	17
	Couche système d'exploitation	17
	Couche de virtualisation	18
	Couche middleware	18
	Couche applicative	18
	Couche de gestion des données	18
	Couche de sécurité	19
2.14.	Continuité d'activité.....	19
4	ARCHITECTURE APPLICATIVE.....	20
3.1.	Schéma de l'architecture applicative.....	20
3.2.	Architecture détaillée de l'application.....	20
	Le socle applicatif.....	20
	Les webservices / API.....	21
	Les applications <i>frontend</i>	23

Spécificités de l'application.....	24
3.3. Interfaces.....	26
Interfaces internes.....	26
Interfaces externes.....	26
3.4. Traitements batch	27
3.5. Liste des composants applicatifs et <i>middlewares</i>	27
5 GESTION DES ACCES A L'APPLICATION	28
4.1. Authentification	28
4.2. Les appels d'API par les <i>backends</i>	28
4.3. Sécurisation des flux d'API	29
Appels vers le SI CRISTAL par les applications externes	29
4.4. Création et purge des comptes	29
4.5. Gestion des droits / profils applicatifs	29
Droits d'accès aux fonctionnalités.....	30
Droits d'accès aux ressources	30
4.6. Administration de l'application.....	31

1 | INTRODUCTION

1.6. RACI : qui fait quoi ?

Le tableau suivant présente les responsabilités en ce qui concerne la rédaction du DAT :

Paragraphe / Rôle	Chef de projet MOE	Prestataire	Architecte	Pôle STS	RSSI
1. Introduction	R	I	A	I	A
2. Présentation fonctionnelle de la solution	R/A	C	I	I	I
3. Sécurité	A	I	I	C	R
4. Architecture applicative	A	C	R	I	I
5. Gestion des accès à l'application	I	I	R	I	A
6. Architecture technique	I	I	C/A	R	I
7. Liste des environnements de l'application	A	I	R	C	I

Où :

- ☐ **R** : Réalisateur (rédacteur)
- ☐ **A** : Approbateur (vérification de l'exactitude des informations)
- ☐ **C** : Contributeur (apporte de informations au rédacteur)
- ☐ **I** : Informé.

1.7. Documents de référence

Information (supprimer l'encart dans la version validée du DAT)

Lister tous les documents référencés par ce présent DAT, techniques ou autres.

Code	Titre du document	Version	Lieu de stockage
REF_DAT-01	AbM - Externe - DAT socle technique WSO2 Keycloak Portail SIPG - rev_Archi.docx	final	\\ad.0efg\partages\SI\Pôle_SIM\Organisation\Référentiels\Architecture\DAT
REF_DAT-02	ABM-REFONTE_CRISTAL-DAT-v2.1.docx	v2.0 (002)	\\ad.0efg\partages\SI\Pôle_SIM\Unité_archi\DAT

2 | PRESENTATION FONCTIONNELLE DE LA SOLUTION

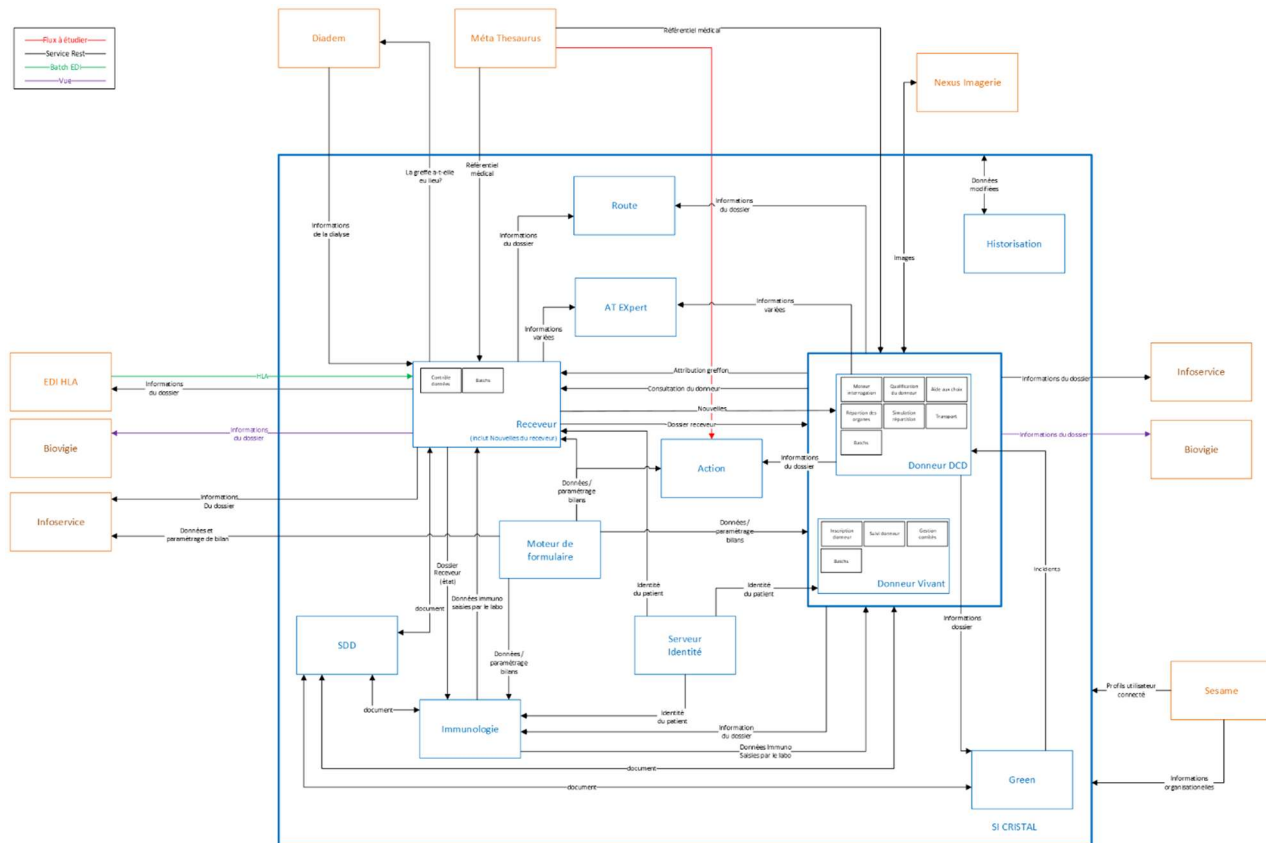
2.2. Principales fonctionnalités de l'application

Cristal est le système d'information développé par l'Agence de la biomédecine pour la gestion des activités de prélèvement et de greffe d'organes et de tissus. Il fonctionne H24 7 jours sur 7.

Les fonctionnalités principales sont :

- Donneur Décédé : Qualification du donneur, répartition des organes, aide au choix (outil automatique d'appariement donneur/receveur), traçabilité des transports de greffon
- Receveur : Inscription en Liste Nationale D'attente, déclarations de greffe, suivi du receveur après-greffe
- Donneur Vivant : Inscription du donneur vivant, planification et o des comités, suivi du donneur, gestion du don croisé.
- Stockage et mise à disposition de documents PDFs (via l'API "SDD", Serveur De Documents)
- Recueil d'incidents Donneurs et alertes par mail des équipes Receveurs ("Green"). Un batch envoie des mails de relance d'alerte chaque nuit.
- Gestion commune des identités des Receveurs, Dialysés (Diadem) et Donneurs Vivants (IU, Identité Unique)
- Saisie et consultation des données immunologiques des Receveurs et Donneurs
- Saisie et consultation de données dans des formulaires dynamiques gérées par un "Moteur de Formulaires"
- Echanges automatiques de données avec des laboratoires HLA ("EDI HLA")
- Mise à disposition en consultations de certaines informations Receveur/Donneur à d'autres applicatifs du SI de l'Agence : Biovigie, Infoservice, Diadem
- Interactions avec un serveur d'imagerie médicale

2.3. Schéma de l'architecture fonctionnelle



2.4. Acteurs et utilisateurs

Type acteur	Effectif approximatif	Rôle
Coordination Hospitalière de prélèvement	1600	Personnels externes à l'ABM, sur site de prélèvement. En charge de l'organisation du prélèvement et de la constitution du dossier Donneur Décédé
Equipe de greffe	3500	Personnels externes à l'ABM, sur site de greffe. Composées d'un coordinateur de greffe (référént du patient, en charge de le tenir informé, organise son accueil ainsi que la réception du greffon) et d'une équipe médicale et paramédicale en charge des activités allant de l'indication au suivi post-greffe. Inscrit les receveurs d'organes sur la LNA, recrute et inscrit les Donneurs Vivants, assure le suivi post-greffe des patients
Equipe de greffe de tissus	1000 (?)	Principalement : ophtalmologues en charge de l'inscription et de la greffe des patients en attente de cornée
Membre Comités Donneur Vivant	200	Experts (internes ou externes) habilités à approuver une candidature de donneur vivant.

Banque de Tissus	100	Réceptionnent et cèdent pour greffe les greffons de tissus.
Laboratoires HLA	450	Effectuent le typage HLA des donneurs et receveurs
Secrétaire de Service Régional	12	Contrôlent et cloturent les dossiers des donneurs DCD, alertent les équipes de greffe via Green
Cadre infirmier Animateur de Réseau	12	Contrôlent/valident les dossiers DCD,
Médecin Régulateur de niveau 2	4	Assurent le rôle de "régulateurs de niveau 2" avec le niveau 1 CIAR (pendant 24h) + validation/vérification des dossiers donneurs vivants
Médecin Expert organes / tissus	3	Expertise des priorités (super-urgence) et analyse des incidents de sécurité sanitaire
Gestion Liste Nationale Attente	3	Gestion des dossiers de patients en attente de greffon
Gestion Registre National des Refus		Gestion du registre national des refus
Supervision Pôle de répartition	3	Supervisent la régulation/répartition du don décédé et don vivant
Gestion répartition	15	IDR/ADR : gestion répartition + LNA (priorités, indisponibilités des équipes)
Evaluation	15	Analyse de la donnée, data scientist, biostat, en vue de l'évaluation de la greffe.
Secrétaire Donneur Vivant	4	Instruction dossiers DVI + organisation comités
Chargés de Recherche en Données de Santé	5	Agents du PQD. Dédoublonnage et contrôle de la qualité de la saisie. Assistance aux utilisateurs de receveur, formation, guide des scores
Chef de service Pole Qualité des Données	1	Idem CRDS
Unité Cristal	3	Agents DSI en charge des évolutions et du MCO Cristal. Installations techniques, développements, migration de données, analyse/corrections anomalies, support applicatif de second niveau.
Support applicatif	1	Support applicatif de premier niveau, création de comptes utilisateurs
Agents DSI en astreintes	6	Support technique hors horaires de bureau.

TOM	20	Equipes de greffe de Nouvelle Calédonie et Polynésie. Gèrent l'ensemble de la chaîne régulation/répartition O/T sur leur territoire.
Unités de thérapie cellulaires	30	Gèrent les greffes d'îlots de Langerhans

2.5. Cycle de vie des données et volumétrie

Type de données	Durée de conservation	Volumétrie
PATIENT / RECEVEUR : - Données d'identification ; - Données de santé. - Nom, prénom, date de naissance, sexe, - Données immunologiques (groupe sanguin, typage HLA, etc.)	Conservation sans limite des données relatives aux receveurs (dans le cadre du suivi des patients greffés) ;	Environ 350000 dossiers receveurs en bdd, en croissance d'environ 15 000 par an
DONNEUR : - Données d'identification ; - Données de santé. - Nom, prénom, sexe, date de naissance, - Dossier médical du patient, Code d'identification	Conservation sans limite des données relatives aux donneurs vivants ou décédés prélevés ; Conservation pendant 3 ans (à partir de la déclaration de non-prélèvement) des données relatives aux donneurs (vivants ou décédés) non prélevés puis anonymisation des données (sans possibilité de réidentification de la personne).	Environ 200 000 dossiers donneurs en bdd, en croissance d'environ 10 000 par an

(source : PIA Cristal)

2.6. Exigences et contraintes particulières exprimées par le métier

Information (supprimer l'encart dans la version validée du DAT)

Rappeler ici les exigences et contraintes particulières (juridiques, légales, ...) exprimées par le métier dans le cahier des charges ou dans d'autres documents et qui ont un impact sur l'architecture de la solution.

Le SI Cristal doit être disponible H24, sans perte de données en cas d'incident d'exploitation.

3 | SECURITE

2.7. Organisation de la sécurité autour du SI

Information (supprimer l'encart dans la version validée du DAT)

A partir des éléments mis à disposition dans la note d'orientation, les besoins de sécurisation devront être couverts un à un.

- ✓ La disponibilité du service et de l'application
- ✓ L'intégrité des informations
- ✓ La confidentialité des informations
- ✓ La traçabilité des actions

Décrire ici comment l'application s'intègre dans un contexte global d'organisation de la sécurité, s'il existe des dispositifs de surveillance, de supervision, par exemple, dans lesquels des alertes émises par l'application seraient remontées (et réciproquement).

Tableau de synthèse de l'organisation de la sécurité au sein du Système d'information Cristal

N°	Périmètre	Spécificités Système d'Information Cristal
	<p>Couche physique (Infrastructure matérielle)</p> <ul style="list-style-type: none"> Cette couche regroupe tous les équipements matériels sur lesquels repose le SI. Cela inclut les serveurs, les ordinateurs, les terminaux (PC, smartphones, tablettes), les réseaux (câblage, routeurs, commutateurs), les dispositifs de stockage, ainsi que les centres de données. C'est la base sur laquelle toutes les autres couches sont construites. 	<p>L'application CRISTAL se situe à la fois sur le site de Lognes (site de production) et à la fois sur le site d'Aubervilliers (site de reprise).</p> <p>Les deux sites en question sont des sites distincts disposant de norme de sécurité associés à des sites industriels.</p>
	<p>Couche réseau</p> <ul style="list-style-type: none"> Elle est responsable de la communication entre les différents composants du système. Cela inclut les protocoles de communication (TCP/IP, HTTP, etc.), les dispositifs réseau (pare-feu, commutateurs, routeurs), ainsi que la connectivité à Internet et à d'autres réseaux externes. Cette couche garantit que les données puissent circuler entre les différents équipements du SI et que les utilisateurs puissent y accéder à distance ou localement. 	
	<p>Couche système d'exploitation (OS)</p> <ul style="list-style-type: none"> Cette couche regroupe les systèmes d'exploitation (Windows, Linux, macOS, etc.) qui permettent de gérer les ressources matérielles du SI (processeur, mémoire, périphériques) et d'assurer l'exécution des applications. Elle sert de passerelle entre le matériel et les applications logicielles. C'est au niveau du système d'exploitation que des fonctions de sécurité, de gestion des ressources et des utilisateurs sont mises en place. 	
	<p>Couche de virtualisation (si présente)</p> <ul style="list-style-type: none"> Dans certains SI, la virtualisation permet de créer des environnements isolés appelés machines virtuelles (VM) qui tournent sur un même matériel physique. Cette couche permet d'optimiser l'utilisation des ressources matérielles et d'assurer une gestion plus flexible des serveurs. Elle se situe entre le système d'exploitation et l'infrastructure physique. 	
	<p>Couche middleware</p> <ul style="list-style-type: none"> Le middleware est un logiciel qui facilite la communication et la gestion des données entre les applications logicielles et les systèmes sous-jacents. Il s'agit d'un ensemble de services permettant d'assurer l'interopérabilité entre des applications hétérogènes, de gérer les transactions, d'assurer la communication entre différents composants du SI, ou encore de fournir des services de sécurité. Exemple de middleware : serveurs d'applications (Apache Tomcat, JBoss), systèmes de gestion de files d'attente, etc. 	
	<p>Couche applicative</p> <ul style="list-style-type: none"> Cette couche contient les applications métiers qui traitent directement les données et offrent des services spécifiques aux utilisateurs. 	

	<ul style="list-style-type: none"> Les applications peuvent être déployées localement ou sur le cloud, et leur rôle est de répondre aux besoins fonctionnels de l'organisation 	
	<p>Couche de gestion des données</p> <ul style="list-style-type: none"> Cette couche regroupe les systèmes de gestion de bases de données (SGBD), qui sont responsables de l'organisation, de la gestion et du stockage des données utilisées par les applications. Elle inclut également la gestion des données dans des entrepôts de données (Data Warehouses), des bases de données distribuées, des solutions de Big Data et de stockage en cloud. 	
	<p>Couche de sécurité</p> <ul style="list-style-type: none"> La couche de sécurité s'intercale à tous les niveaux du SI et s'assure de la protection des données, des accès et des communications. Elle comprend des dispositifs de contrôle d'accès, de chiffrement des données, d'authentification (multi-facteurs, gestion des identités), de pare-feu, de détection des intrusions, ainsi que des solutions de gestion des incidents. Elle vise à protéger l'intégrité, la confidentialité et la disponibilité des données et des services du SI. 	<p>La couche de sécurité est portée par :</p> <ol style="list-style-type: none"> 1 - L'architecture portail d'entreprise, l'infrastructure SSO (Keycloak/ LDAP / SESAME / ...) et permet : L'identification et l'authentification des métiers aux différents modules de l'application CRISTAL. 2 - Par le cloisonnement réseau et télécoms (VLAN, Firewall, DMZ, sonde IA, etc.) 3 - par les services de sécurité associés : SOC/SIEM Externalisé 4 - Réplication des données entre le site nominal et le site de secours.
	<p>Couche utilisateur (Interface utilisateur)</p> <ul style="list-style-type: none"> Cette couche représente l'interaction entre les utilisateurs (internes ou externes) et le Système d'Information à travers des interfaces graphiques (applications desktop, sites web, applications mobiles). Elle comprend également les outils de communication (email, messagerie instantanée, etc.) qui permettent aux utilisateurs de dialoguer avec le SI. 	
	<p>Couche de gestion et de monitoring</p> <ul style="list-style-type: none"> Cette couche est responsable de la supervision du bon fonctionnement du SI. Elle comprend les outils de monitoring, de gestion des performances (gestion des logs, surveillance des serveurs et applications) et d'audit. Cela permet de détecter rapidement les anomalies, de réaliser des analyses de performance et de prendre des mesures correctives lorsque nécessaire. 	
	<p>Couche de gouvernance, conformité et gestion des risques</p> <ul style="list-style-type: none"> Cette couche regroupe les pratiques de gestion de la sécurité, de conformité aux normes et réglementations (RGPD, IM901, ISO 27001, etc.), ainsi que la gestion des risques liés au Système d'Information. Elle veille à ce que les processus internes respectent les standards de sécurité, les bonnes pratiques et la réglementation en vigueur. 	

2.8. RGPD et données sensibles

Information (supprimer l'encart dans la version validée du DAT)

Préciser ici si l'application collecte ou manipule des données personnelles, des données de santé nominatives ou d'autres données sensibles ou confidentielles.

Détailler les mesures particulières prises dans le cadre du RGPD.

Rappel des principes de classification des données

La **classification des données** est un processus fondamental pour gérer et protéger les informations au sein d'un Système d'Information (SI). Elle consiste à attribuer des catégories ou des niveaux de sensibilité aux données en fonction de leur valeur, de leur confidentialité et de leur importance pour l'Agence. L'objectif est de déterminer des mesures de sécurité et des politiques adaptées pour chaque type de données, en fonction de leur classification.

N°	Périmètre	Spécificités Cristal
	<p>Principe de sensibilité des données</p> <ul style="list-style-type: none"> Les données sont classifiées en fonction de leur degré de sensibilité. Les informations plus sensibles, comme des données personnelles ou des informations financières, nécessitent un niveau de protection plus élevé que des données moins sensibles. Exemple de catégories : public, interne, confidentiel, secret. 	<p>L'ensemble des données métiers sont classifiées comme confidentiel.</p> <p>L'ensemble des éléments de sécurité : mot de passe, certificats, etc. sont classifiés comme secret.</p>
	<p>Principe du "besoin de savoir" (Need to Know)</p> <ul style="list-style-type: none"> Ce principe stipule que l'accès aux données doit être restreint uniquement à ceux qui en ont réellement besoin dans le cadre de leur fonction. Les personnes autorisées à accéder à des données classifiées doivent avoir un besoin spécifique pour l'exploiter. Cela signifie que chaque donnée doit être protégée en fonction de l'utilisateur et du rôle qui y a accès. 	<p>Une politique d'habilitation a été formalisée au plus près des besoins utilisateur avec le principe de tout ce qui n'est pas explicitement autorisé est interdit.</p>
	<p>Principe de la minimisation des données</p> <ul style="list-style-type: none"> Les données doivent être collectées et stockées uniquement dans la mesure nécessaire. Ce principe est particulièrement important dans le cadre du respect des réglementations comme le RGPD (Règlement Général sur la Protection des Données) en Europe. Cela implique une classification qui assure qu'aucune donnée sensible n'est collectée ou conservée de manière excessive. 	<p>Le cycle de vie des données a été défini et les données devant être archivées de manière à être force probante en cas de litige devant la justice.</p> <p>></p>
	<p>Principe de l'intégrité des données</p> <ul style="list-style-type: none"> Les données classifiées doivent être protégées contre toute modification ou altération non autorisée. Le principe de l'intégrité assure que les données restent exactes et fiables tout au long de leur cycle de vie. Pour les données sensibles, des mécanismes comme des contrôles de version, des systèmes de validation et des journaux d'audit doivent être utilisés pour en garantir l'intégrité. 	<p>Des mécanismes spécifiques ont été mis en œuvre afin d'assurer l'intégrité des échanges de données (flux HTTPS). Les habilitations définies permettent de s'assurer que les droits en écriture sur les données sont maîtrisés sachant qu'une politique de traçabilité au moins sur les données les plus sensibles permet de s'assurer des pistes d'audit en cas de perte d'intégrité détectée.</p>
	<p>Principe de la séparation des données sensibles</p> <ul style="list-style-type: none"> Les données doivent être séparées selon leur niveau de sensibilité pour limiter les risques d'exposition. Cela inclut la segmentation du réseau, la mise en place de zones de sécurité (ex. : zones démilitarisées, partitions sécurisées), et l'utilisation de cryptage pour les données les plus sensibles. 	
	<p>Principes de gestion du cycle de vie des données</p> <ul style="list-style-type: none"> La classification des données permet de gérer leur cycle de vie de manière optimale. Cela inclut la collecte, le stockage, l'utilisation, l'archivage et la destruction des données selon leur niveau de sensibilité. Par exemple, les données classifiées comme secret peuvent avoir des durées de rétention plus courtes, et doivent être effacées de manière sécurisée après leur utilisation. 	
	<p>Mise en œuvre pratique de la classification des données</p> <ul style="list-style-type: none"> Étiquetage des données : Les données peuvent être étiquetées ou marquées avec leur niveau de classification afin que les utilisateurs sachent quel type de protection elles nécessitent. 	

	<ul style="list-style-type: none"> ▪ Contrôles d'accès : En fonction de la classification, des règles strictes d'accès peuvent être définies, avec l'utilisation de solutions de gestion des identités et des accès (IAM). ▪ Chiffrement : Les données sensibles (confidentielles, très confidentielles) doivent être chiffrées, tant lorsqu'elles sont stockées que lorsqu'elles transitent sur le réseau. ▪ Formation des employés : Sensibiliser les employés à la classification des données et à la gestion de leur accès est essentiel pour assurer une bonne mise en œuvre de la politique de sécurité. 	
--	--	--

2.9. Disponibilité

Information (supprimer l'encart dans la version validée du DAT)

Pour les besoins de disponibilité de l'application, la description doit permettre de savoir quelles sont les plages de service, quelle est la durée maximale d'interruption qui peut être supportée par les métiers.

- ✓ Le système doit permettre de garantir une continuité de service,
- ✓ Le télé service doit être accessible aux utilisateurs 24/7,
- ✓ Le traitement doit être accessible aux heures de bureau,
- ✓ La durée d'indisponibilité doit être inférieure à 24 heures
- ✓ Les opérations de maintenance et de sauvegardes doivent se faire en dehors des heures de bureau.

Etc.

A partir de la durée maximale d'interruption, et des éventuels sinistres à couvrir, la description indiquera si un site de secours est prévu dans le projet, comment sont opérées les répliques, et à défaut s'il existe des dispositifs de sauvegarde/restauration.

- Temps d'indisponibilité accepté : 2 heures
- Pas de perte de données.
- Plan de continuité en cas d'incident majeur.
- Période de sensibilité critique : 24h sur 24 et 7 jours sur 7.

2.10. Intégrité

Information (supprimer l'encart dans la version validée du DAT)

A partir des besoins identifiés dans la note d'orientation concernant l'intégrité des informations gérées dans l'application, la description indiquera les dispositifs qui sont mis en œuvre pour couvrir les risques d'altération ou de dégradation des informations.

- Aucune perte d'intégrité sur les données de production
- Les habilitations doivent permettre d'assurer la maîtrise des modifications faites sur les données pendant l'ensemble du cycle de vie des données. A ce titre, une donnée peut être modifiable à un instant t et ne plus l'être à un instant t+1.

2.11. Confidentialité

Information (supprimer l'encart dans la version validée du DAT)

A partir des besoins identifiés dans la note d'orientation concernant la confidentialité des informations gérées dans l'application, la description indiquera les dispositifs qui sont mis en œuvre pour couvrir les risques de divulgation des informations.

- L'identification et l'authentification doivent être fort (double facteur par exemple) pour l'ensemble des personnes ayant potentiellement accès aux données de production (ou quelques soient l'environnement où se trouve les données de production comme par exemple les environnements de qualification, de préproduction, etc.).
- L'ensemble des données métiers sont des données classifiées comme confidentiel (ou diffusion restreinte).

- L'ensemble des données critiques comme les mots de passe, les clés de chiffrement, les éléments d'identification sont classifiés comme secret (et doivent être chiffrés)
- L'anonymisation des données doit être envisagée lorsque les données venant de l'extérieur de l'Agence sont stockés dans nos environnements (et que le patient ou le donneurs n'ont pas besoin d'être connus) afin de garantir à nos interlocuteurs que les données de citoyens sont protégés.

2.12. Traçabilité et piste d'audit

Information (supprimer l'encart dans la version validée du DAT)

(Traçabilité)

Est-ce que l'application peut stocker les traces de connexions (et/ou d'activité dans l'application) des utilisateurs ? Cette fonctionnalité est-elle souhaitée ? A moins que la traçabilité soient assurées à minima par les logs techniques.

(Piste d'audit)

Est-ce que l'application met à disposition des fonctions permettant de retracer le cycle de vie des informations gérées par l'application (de la création de l'information jusqu'à la dernière mise à jour/suppression).

Tableau de synthèse

Besoins en terme de traçabilité ou prouvabilité :	
P1 : Aucun besoin de traces	
P2 : Besoin de traces à des fins d'amélioration de la qualité, de statistiques ou de tableau de bord	
P3 : Besoin de traces tangibles pour un contrôle interne ou réglementaire pour éviter tous incidents ou contestations dû à un manque d'éléments tracés de manière fiable. Les preuves et contrôles font partis des fonctionnalités de l'application.	P3
P4 : Besoin de traces opposables (besoin de non-répudiation), notamment pour répondre à un contentieux juridique. L'absence, la perte ou la falsification des éléments de preuve est inacceptable. Les preuves et contrôles font partis des fonctionnalités essentielles de l'application ou relèvent de contraintes légales.	
Pas de besoin de trace	Besoin de trace (piste d'audit).
Besoin de logs de connexion (personnels concernés, date et heure)	oui
Besoin de logs sur les flux de données entrants	oui
Besoins de logs sur l'octroi d'habilitation (auteur, personnels concernés, ...)	oui
Besoins de logs sur des évolutions de paramétrage (seuils, etc.)	oui
Besoin de logs sur une catégorie d'accès (droits étendus, droits critiques, ...)	oui
Besoin de logs sur des actions particulières sur les données	oui, sachant que les données concernées doivent être répertoriées.
Besoin de logs sur des fonctions particulières de l'application	oui
Besoins de logs réglementaire et contractuels (horodatage, non répudiation)	oui
Besoin de logs sur les transferts de données (date, heure, données, ...)	oui
Besoin de logs sur des modifications de données critiques (données médicales, ...)	oui
Besoin de détecter la désactivation de la fonction d'audit au sein de l'application.	non
Besoin d'historique de données et sur quels types de données	oui, pour les données de santé et les données nominatives. Les dossiers doivent être conservés de 10 à 30 ans.
Besoin d'archivage des données (nombre de mois, d'année)	Hors projet.
Besoin d'archivage des traces (nombre de mois, d'année)	

2.13. Gestion des incidents

Information (supprimer l'encart dans la version validée du DAT)

Existe-t-il un dispositif ou une organisation permettant de gérer les incidents, remontés manuellement ou automatique par l'application ?

Rappel des principes de sécurité dans la gestion des incidents

La gestion des incidents de sécurité dans un **Système d'Information (SI)** doit être effectuée de manière structurée et coordonnée afin de minimiser l'impact sur l'Agence et de protéger ses actifs. Le traitement des incidents de sécurité doit couvrir l'ensemble des couches du SI, car chaque couche peut être impactée par un incident (que ce soit au niveau matériel, logiciel, réseau, données ou utilisateur).

Tableau de synthèse des différentes couches du Système d'Information et des mesures pouvant être mise en œuvre en cas d'incident et niveau de couverture pour l'application Cristal

N°	Périmètre	Spécificités pour l'application CRISTAL
	Couche physique (Infrastructure matérielle) <ul style="list-style-type: none"> ■ Identification de l'incident : La première étape consiste à identifier les incidents qui affectent l'infrastructure matérielle, tels que la défaillance des équipements, les pannes des serveurs, la perte ou le vol de matériel, ou encore la compromission physique de centres de données (incendies, inondations, etc.). ■ Mesures immédiates : <ul style="list-style-type: none"> ○ Isolement du matériel compromis : Si un matériel est suspecté d'être compromis (par exemple, un serveur contenant des données sensibles), il doit être immédiatement isolé du réseau. ○ Vérification physique : S'assurer que le matériel n'a pas été manipulé de manière malveillante, que les câblages sont en ordre et que les dispositifs de sécurité physique (verrouillage, surveillance vidéo) sont fonctionnels. ○ Remplacement du matériel : En cas de panne ou de défaillance, le matériel défectueux doit être remplacé ou réparé, selon le cas. ■ Suivi : Effectuer une analyse post-incident pour déterminer les causes de la défaillance ou de l'attaque physique (ex. : vol de matériel), et renforcer les mesures de sécurité physique si nécessaire (ex. : amélioration des contrôles d'accès physiques). 	Traiter par la solution Nagios et Dynatrace
	Couche réseau <ul style="list-style-type: none"> ■ Identification de l'incident : Un incident réseau peut inclure une attaque par déni de service (DoS), une intrusion via un pare-feu ou un routeur compromis, ou une fuite de données. ■ Mesures immédiates : <ul style="list-style-type: none"> ○ Isolement du réseau : Si un composant réseau est compromis (par exemple, un serveur compromis ou un dispositif de pare-feu), il doit être immédiatement isolé pour éviter la propagation de l'incident. ○ Analyse du trafic réseau : Utiliser des outils de surveillance réseau (SIEM, IDS/IPS) pour identifier les sources de l'attaque, analyser le trafic suspect et bloquer l'accès non autorisé. ○ Blocage des adresses IP malveillantes : Les attaquants peuvent utiliser des adresses IP spécifiques pour lancer des attaques. Il est important de bloquer ces adresses pour limiter les dommages. ○ Réduction des vecteurs d'attaque : S'assurer que les dispositifs réseau sont correctement configurés et que les protocoles non sécurisés sont désactivés. ■ Suivi : Réaliser une analyse approfondie pour comprendre comment l'incident a pu se produire (par exemple, vulnérabilité d'un dispositif réseau mal configuré) et apporter les corrections nécessaires (mise à jour des pare-feu, changement des configurations réseau, etc.). 	Traiter par le SOC/SIEM externalisé et les sondes IA mise en œuvre au sein du Système d'Information.
	Couche système d'exploitation <ul style="list-style-type: none"> ■ Identification de l'incident : Les incidents au niveau du système d'exploitation peuvent inclure des exploits de vulnérabilités (par exemple, une attaque par élévation de privilèges), des malwares, ou des attaques de type ransomware. ■ Mesures immédiates : <ul style="list-style-type: none"> ○ Identification de l'attaque : Utiliser des outils de monitoring du système pour détecter les signes d'une intrusion ou d'une compromission (ex. : activités suspectes dans les logs du système). ○ Isolement de la machine infectée : Si un système d'exploitation est compromis, il doit être isolé du réseau pour éviter la propagation de l'attaque. 	<p>Le suivi des vulnérabilités au niveau de la couche exploitation n'est pas mise en œuvre</p> <p>Les logs des points de cloisonnement sont traités (SOC/SIEM Externalisé)</p> <p>Les sondes IA permettent d'identifier l'incident.</p>

	<ul style="list-style-type: none"> ○ Application des patches de sécurité : Mettre à jour les systèmes d'exploitation avec les derniers correctifs de sécurité pour éliminer les vulnérabilités exploitées par l'attaque. ○ Analyse des logs : Examiner les journaux du système pour détecter des traces d'attaques (connexions non autorisées, modifications anormales de fichiers système, etc.). ■ Suivi : Après avoir nettoyé l'infection, mener un audit approfondi pour identifier la cause de l'incident et ajuster les politiques de sécurité du système (renforcement de l'accès administrateur, installation de solutions antivirus, etc.). 	
	<p>Couche de virtualisation</p> <ul style="list-style-type: none"> ■ Identification de l'incident : Les incidents peuvent survenir dans des environnements virtualisés, tels que des attaques visant à compromettre des machines virtuelles ou à exploiter des failles dans l'hyperviseur. ■ Mesures immédiates : <ul style="list-style-type: none"> ○ Isolement des VM compromises : Si une machine virtuelle est infectée, elle doit être immédiatement désactivée ou isolée pour éviter que l'attaque ne se propage aux autres VMs. ○ Analyse de l'hyperviseur : S'assurer que l'hyperviseur (la couche de virtualisation) n'a pas été compromis. En cas de doute, il peut être nécessaire de redémarrer l'hyperviseur en mode sécurisé et d'analyser les logs de sécurité. ■ Suivi : Mettre à jour les systèmes de virtualisation pour combler les éventuelles failles exploitées, et reconfigurer les politiques de sécurité des machines virtuelles. 	A voir
	<p>Couche middleware</p> <ul style="list-style-type: none"> ■ Identification de l'incident : Les incidents peuvent inclure des attaques visant des serveurs d'applications, des défaillances dans la gestion des transactions ou des erreurs dans l'intégration des services. ■ Mesures immédiates : <ul style="list-style-type: none"> ○ Réinitialisation des services compromis : Si un middleware (par exemple, un serveur d'application ou un gestionnaire de base de données) est compromis, il est essentiel de réinitialiser le service ou de le mettre hors ligne pour l'analyse. ○ Contrôle de l'intégrité des données : Vérifier que les transactions et les données n'ont pas été altérées pendant l'incident. ■ Suivi : Auditer les configurations des serveurs middleware et appliquer des correctifs si nécessaire. Il est aussi important de s'assurer que les mécanismes de journalisation sont activés pour traquer les erreurs et les accès non autorisés. 	A voir
	<p>Couche applicative</p> <ul style="list-style-type: none"> ■ Identification de l'incident : Les incidents au niveau des applications peuvent inclure des failles de sécurité dans le code, des attaques par injection (SQL, XSS, etc.), ou des défaillances de performance. ■ Mesures immédiates : <ul style="list-style-type: none"> ○ Suspension de l'application compromise : Si une application est compromise, il peut être nécessaire de la suspendre temporairement pour éviter la propagation de l'incident. ○ Correction du code : Corriger immédiatement toute vulnérabilité connue dans l'application, appliquer des patches de sécurité, et vérifier les logs d'erreurs pour détecter toute activité suspecte. ■ Suivi : Effectuer un test de régression pour vérifier que les correctifs n'ont pas introduit de nouveaux problèmes, et mettre en place des contrôles de sécurité plus stricts pour prévenir les attaques futures. 	Il existe un protocole d'audit de code qui permet de détecter les failles de sécurité au niveau de la couche application et un suivi dans le temps des nécessités de correctifs (librairie, etc.).
	<p>Couche de gestion des données</p> <ul style="list-style-type: none"> ■ Identification de l'incident : L'incident peut impliquer la fuite, la corruption ou la suppression non autorisée de données sensibles. ■ Mesures immédiates : <ul style="list-style-type: none"> ○ Protection des données sensibles : Si des données sensibles sont compromises (par exemple, données personnelles), il est important de prendre des mesures immédiates, comme le chiffrement des données en transit et leur stockage sécurisé. ○ Restaurer les données à partir de sauvegardes : Si des données ont été corrompues ou perdues, restaurer les données à partir des sauvegardes les plus récentes. ■ Suivi : Examiner les politiques de sauvegarde et de gestion des données pour éviter que l'incident ne se reproduise, et renforcer les contrôles d'accès aux bases de données. 	

	<p>Couche de sécurité</p> <ul style="list-style-type: none"> ■ Identification de l'incident : Cela inclut les incidents touchant les contrôles de sécurité comme les violations de la gestion des identités, la compromission des accès, ou l'exploitation de vulnérabilités dans les dispositifs de sécurité. ■ Mesures immédiates : <ul style="list-style-type: none"> ○ Réinitialiser les identifiants compromis : Si des identifiants (mots de passe, clés d'API) sont compromis, réinitialiser les accès et renforcer l'authentification (par exemple, avec une authentification multifactorielle). ○ Mise en place de blocages ou de restrictions d'accès : Appliquer immédiatement des restrictions sur les comptes utilisateurs et les accès réseau pour contenir la menace. ■ Suivi : Mener une analyse approfondie de l'incident de sécurité pour comprendre les causes et mettre en œuvre des mesures correctives et préventives (mise à jour des politiques de sécurité, renforcement de la gestion des accès). 	
--	---	--

2.14. Continuité d'activité

Information (supprimer l'encart dans la version validée du DAT)

A partir des besoins identifiés dans la note d'orientation, la description fournira les informations permettant de comprendre comment la Durée Maximale D'Interruption Acceptable (DMIA) est respectée par l'application.

- Préciser si un Plan de Continuité de l'Activité est mis en œuvre (PCA-PRA), il conviendra d'indiquer si des exercices ont déjà été réalisés
- Préciser si l'architecture est redondée (serveurs/data center), s'il existe un plan de traitement des sinistres globaux (incident, inondation, attaques etc.)?

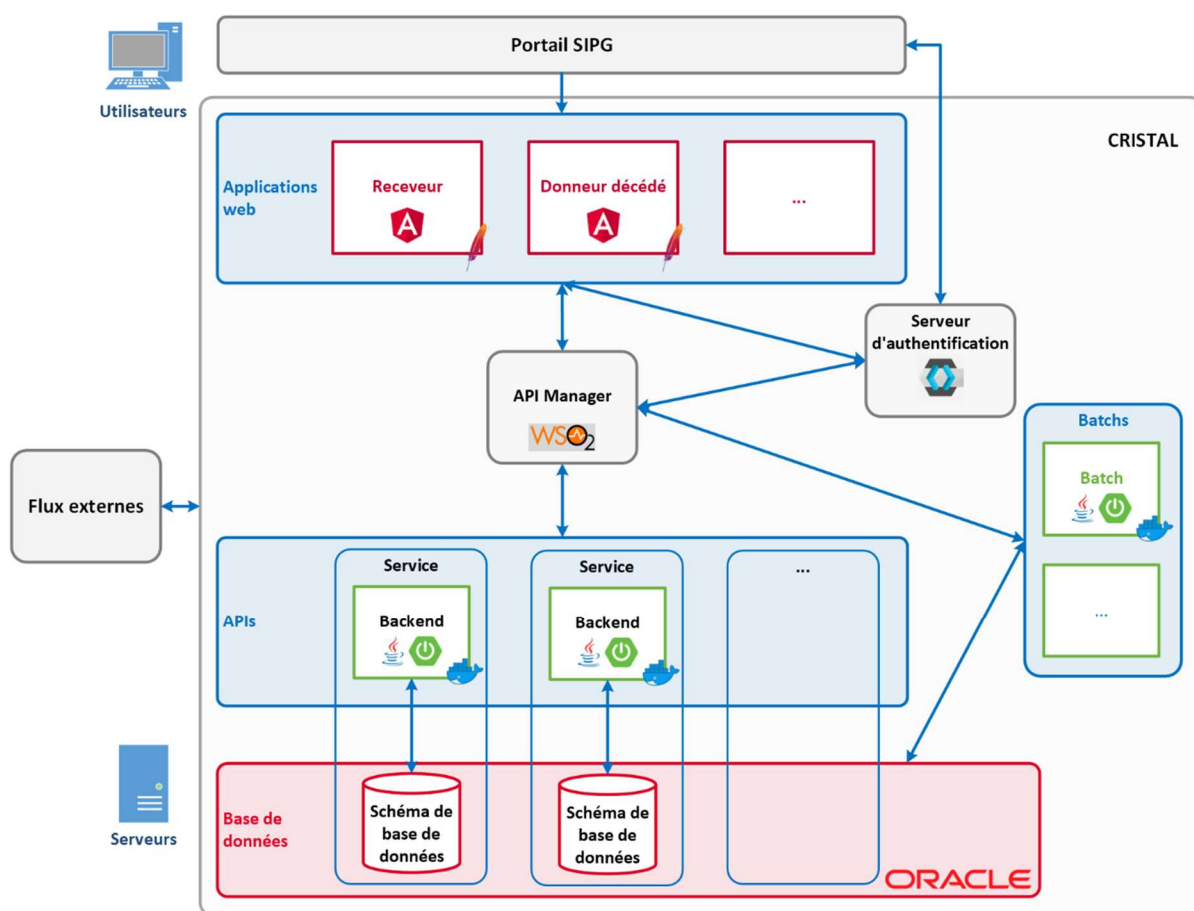
Préciser si des exercices (sinistres graves) sont mis en œuvre.

4 | ARCHITECTURE APPLICATIVE

3.1. Schéma de l'architecture applicative

Information (supprimer l'encart dans la version validée du DAT)

Représenter visuellement l'application dans son écosystème en mettant en exergue ses interactions avec les briques environnantes : les différents composants de l'application (de façon détaillée), les *middlewares* et briques techniques (de façon simplifiée), les autres applications et logiciels avec lesquels l'application dialogue.



L'architecture choisie pour le projet de refonte du SI CRISTAL est une architecture orientée API.

Chaque application refondue du SI CRISTAL est découpée en blocs logiques :

- Une base de données contenant les données propriétés du backend.
- Un backend implémentant les services métiers et les exposant à travers des API.
- Un frontend Web permettant de consulter et manipuler les données exposées par les API.

L'accès aux applications se fait à travers le Portail SIPG qui permet une authentification SSO.

La gestion de l'authentification et des autorisations est déléguée à un module transverse.

3.2. Architecture détaillée de l'application

Le socle applicatif

Information (supprimer l'encart dans la version validée du DAT)

Décrire succinctement si c'est une application Angular/Spring, Kasper, Outsystems, etc.
 Pour les détails relatifs au socle renvoyer le lecteur au document de référence précisé dans le §1.2.

Les applications de Cristal Refonte sont *full* Angular/Spring (Java). Les applications *frontend* sont des applications Angular. Le *backend* implémente et expose des services métier en APIs à travers l'*api-manager*, dans le style RESTful afin de favoriser l'interopérabilité via internet. Les applications *frontend* et les APIs sont conteneurisées, *dockerisées*, s'exécutant sur un même *cluster Swarm* comportant plusieurs nœuds, cf. §4.2.2. La conteneurisation a l'intérêt de simplifier considérablement les déploiements, et permet aussi de se passer des serveurs d'application « lourds » comme JBoss ou Weblogic au profit des serveurs web légers comme Tomcat embarqués dans les images docker des APIs. Celles-ci, ainsi que les images Docker des applications *frontend*, intègrent toute la pile technologique de sorte que les applications soient complètement autonomes et portables pour le « *run* ».

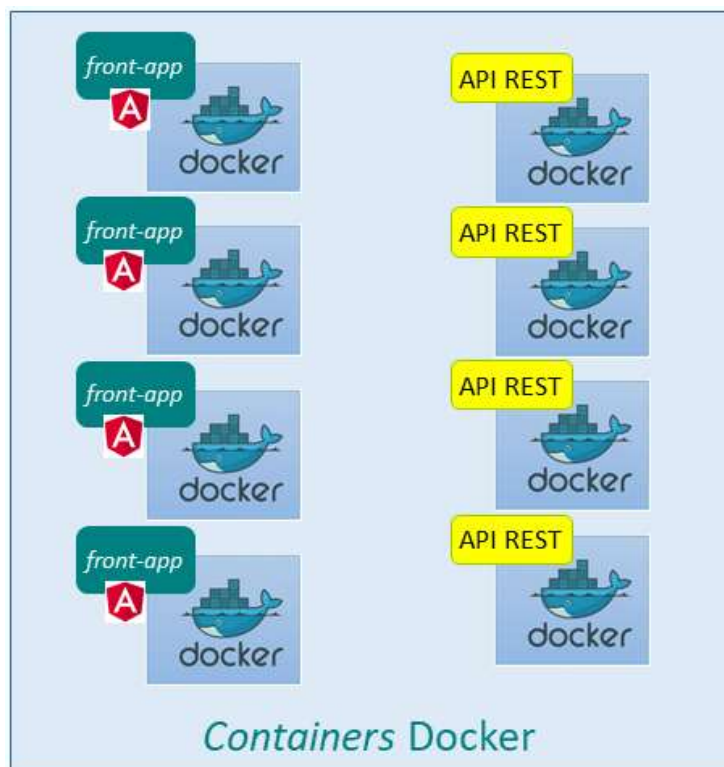


Figure 4.2.1 : Les containers Docker des applications frontend et des APIs.

Les webservices / API

Information (supprimer l'encart dans la version validée du DAT)

Lister les webservices et API exposés par l'application et les fonctionnalités associées. Décrire comment ces webservices sont appelés (API manager, directement) et sécurisés (OAuth2, apikey, etc).
Si le webservice apparaît dans le schéma d'architecture applicative avec un code, le préciser.

Le *backend* de Cristal Refonte comporte un ensemble d'APIs, chacune correspondant à un périmètre fonctionnel bien défini. Ces APIs implémentent la logique métier, sont listées dans le tableau ci-dessous.

Code	Webservice/API	Fonctionnalité	Format	Appel via	Sécurisation
	cristal-api-consultation	Gestion des requêtes transversales (select seulement)	json	<i>api-manager</i>	OIDC / Keycloak + gestion de droits par le <i>backend</i>
	cristal-api-document	Gestion des appels vers SDD (ajout d'une couche de gestion des droits notamment)	json	<i>api-manager</i>	OIDC / Keycloak + gestion de droits par le <i>backend</i>
	cristal-api-donneur	Gestion des tables du domaine donneur (DCD et vivant)	json	<i>api-manager</i>	OIDC / Keycloak + gestion de droits par le <i>backend</i>
	cristal-api-droit	Gestion des tables liées aux droits des utilisateurs	json	<i>api-manager</i>	OIDC / Keycloak +

					gestion de droits par le <i>backend</i>
	cristal-api-formulaire	Gestion des tables du moteur de bilan	json	<i>api-manager</i>	OIDC / Keycloak + gestion de droits par le <i>backend</i>
	cristal-api-identité	Gestion des tables du domaine identité (IU)	json	<i>api-manager</i>	OIDC / Keycloak + gestion de droits par le <i>backend</i>
	cristal-api-paramètre	Gestions des tables liées aux paramétrages	json	<i>api-manager</i>	OIDC / Keycloak + gestion de droits par le <i>backend</i>
	cristal-api-préférence	Gestion des tables liées aux préférences des utilisateurs	json	<i>api-manager</i>	OIDC / Keycloak + gestion de droits par le <i>backend</i>
	cristal-api-receveur	Gestion des tables du domaine receveur	json	<i>api-manager</i>	OIDC / Keycloak + gestion de droits par le <i>backend</i>

Tableau 4.2.2 : Les différentes APIs implémentées par le backend métier de CRISTAL REFONTE.

Compte tenu du (très) grand nombre de services offerts par chacune de ces APIs REST, il n'est pas possible de les lister exhaustivement. Le meilleur moyen d'explorer ces services est de le faire sur l'interface *Developer Portal* d'un *api-manager* tel que Gravitee. Cette console IHM n'est pas uniquement réservée aux architectes, mais ouverte à un public plus large : chefs de projet, MOA, prestataires, autres partenaires. Chaque chef de projet ou chaque utilisateur n'y a accès qu'aux APIs qui correspondent à son périmètre.

L'appel de ces APIs par les applications *frontend* Angular passe par l'*api-manager* (Gravitee) dont le rôle principal est de les sécuriser selon le protocole *OpenID Connect* implémenté avec Keycloak comme serveur d'autorisation.

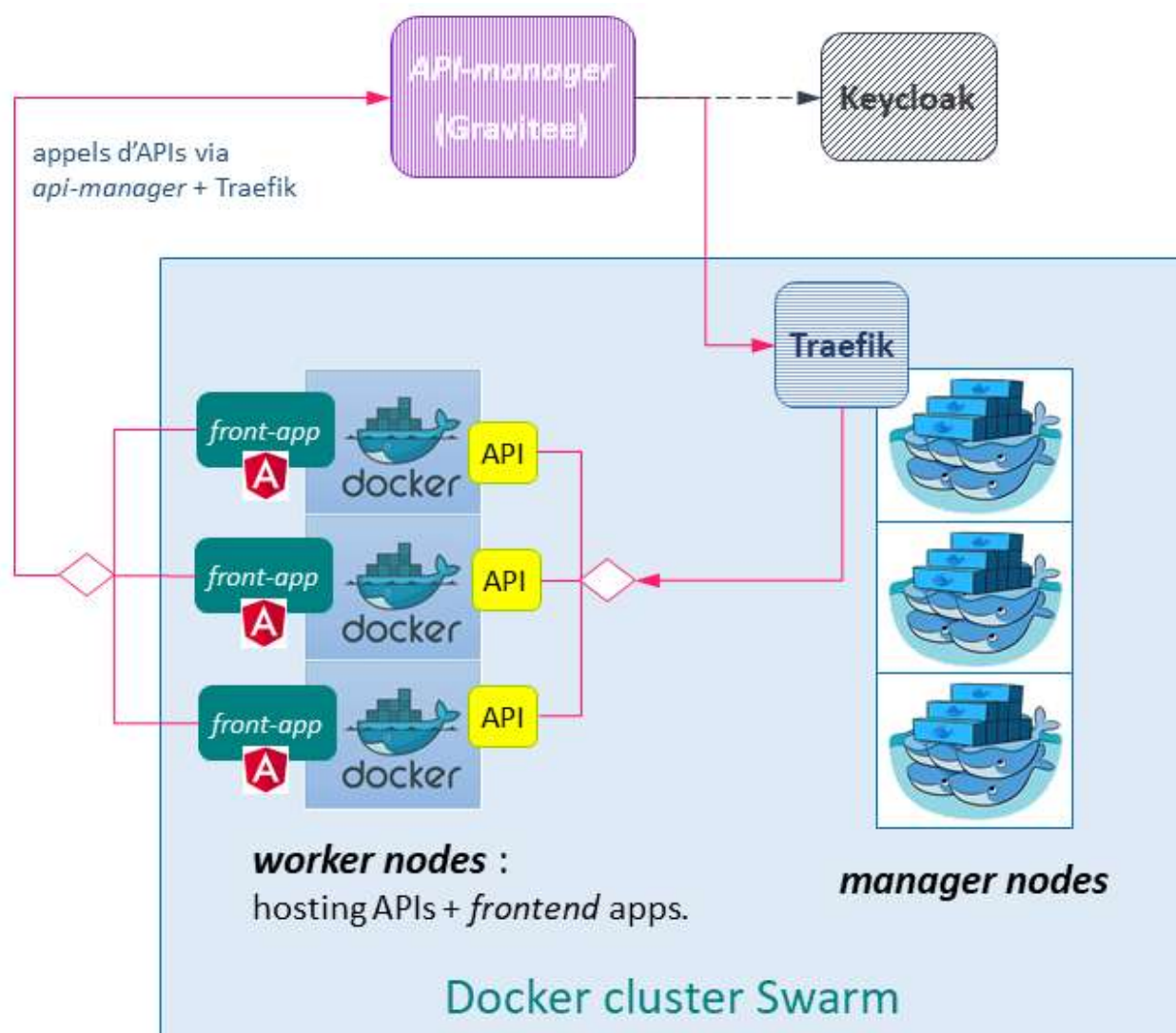


Figure 4.2.2 : Les appels d'API passant par l'api-manager.

Certaines de ces APIs appellent entre elles, en passant également par l'*api-manager* (Gravitee) qui effectue le contrôle d'accès, vérifiant la validité de l'*access_token*. Ces appels ne sont pas représentés sur le schéma ci-dessus. Tous les appels d'API entrants dans le *cluster* Swarm passent par le composant Traefik qui les route vers les containers d'API en tout équilibrant la charge (fonction de loadbalancing) sur ces derniers et selon les règles de routage définies. Les accès sont loggés de façon centralisée sur un syslog

Les applications *frontend*

Nom de l'application	Périmètre
Receveur-front	Gestion des receveurs
Immunologie-front	Gestion des données immunologiques
Donneur-Vivant-front	Gestion des donneurs vivants
Donneur-Decede-front	Gestion des donneurs décédés
Atexpert-Front	Gestion des compatibilités Donneur - Receveur
Administration-front	Gestion de l'administration (substitution de profil)
Ngx-cristal	Librairie transverse utilisée par toutes les applications front

Tableau 4.2.3 : Les applications *frontend*.

Le développement des applications *frontend* s'appuie sur la librairie **PrimeNG**, une librairie de composants Angular, permettant la création rapide d'éléments comme les tableaux, formulaires, formatage des messages d'erreur, etc. PrimeNG s'interface avec la librairie CSS **PrimeFlex** qui facilite la mise en œuvre d'un design réactif. Elle fournit des classes pour un grand nombre de fonctionnalités (disposition, tailles, couleurs, etc.) pour simplifier voire éviter les développements CSS.

Spécificités de l'application

Information (supprimer l'encart dans la version validée du DAT)

Décrire avec un maximum de précisions ce que l'application a de particulier.
Exemple : invocation de l'API d'envoi de SMS, appel à MS-Santé, etc.

3.2.1.1. Utilisation des données de Sesame

Les données issues de SESAME (fonctions, réseaux, équipes, etc.) sont nécessaires à chaque appel à une API CRISTAL afin de vérifier le profil de l'utilisateur et le périmètre auquel il a accès.

Afin d'éviter de surcharger les webservices SESAME, deux mécanismes sont mis en œuvre pour conserver les données SESAME dans les API CRISTAL.

3.2.1.2. Mise en cache des données

Les appels aux webservices SESAME sont faits par l'API Droit. Afin de réduire les appels à SESAME, une couche de cache est mise en place. L'annotation Spring `@EnableCaching` (disponible dans la dépendance Spring Boot Starter Cache) permet d'activer le cache dans l'application et fournit une abstraction pour utiliser différentes options de cache. La solution privilégiée est d'utiliser Ehcache comme implémentation.

La durée de mise en cache des données SESAME est paramétrable *via* une variable d'environnement.

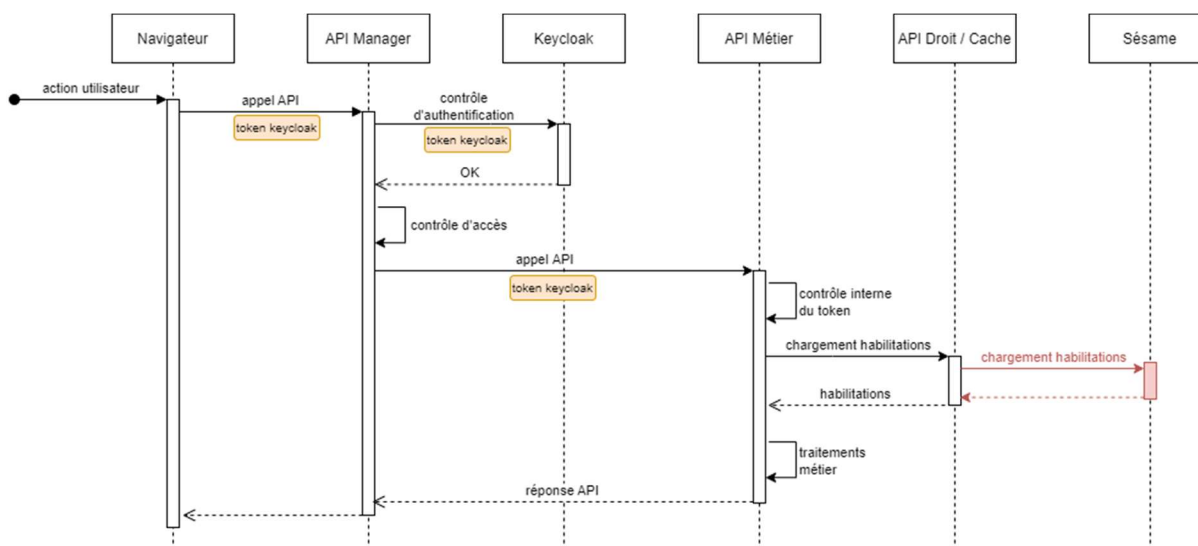


Figure ?? : utilisation du cache dans l'API Droit

3.2.1.3. Utilisation de Spring Security Context

Le deuxième mécanisme mis en place est l'utilisation du contexte de sécurité Spring. Il contient le profil sélectionné par l'utilisateur, les informations nécessaires à la gestion des habilitations et les droits de l'utilisateur. Il est disponible après la réception des requêtes http dans les applications *back*.

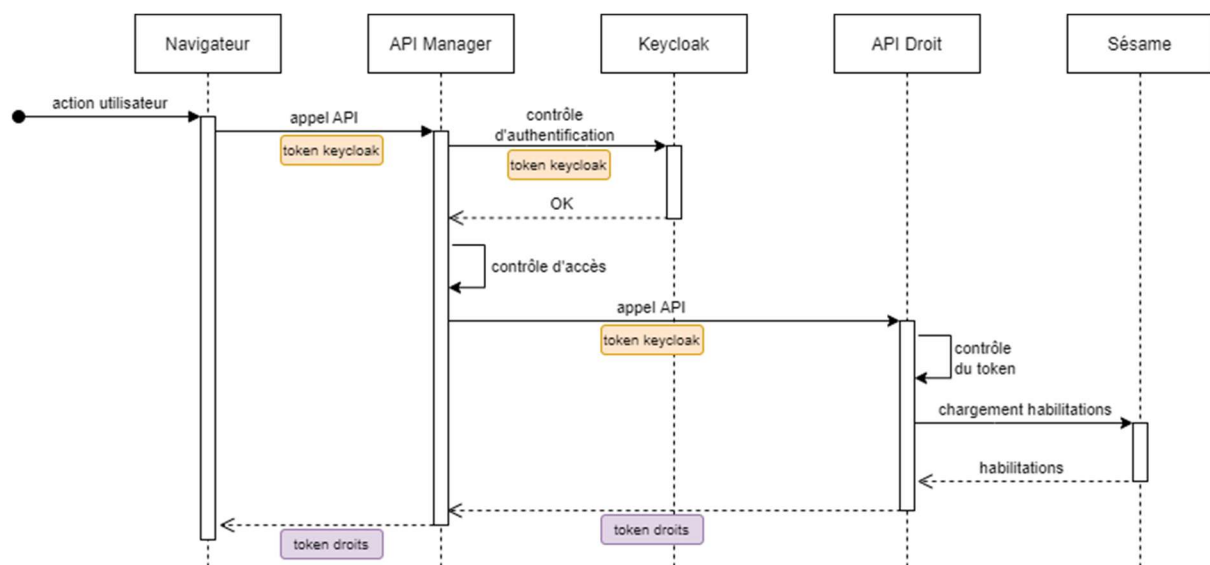


Figure ?? : Chargement des habilitations dans le token

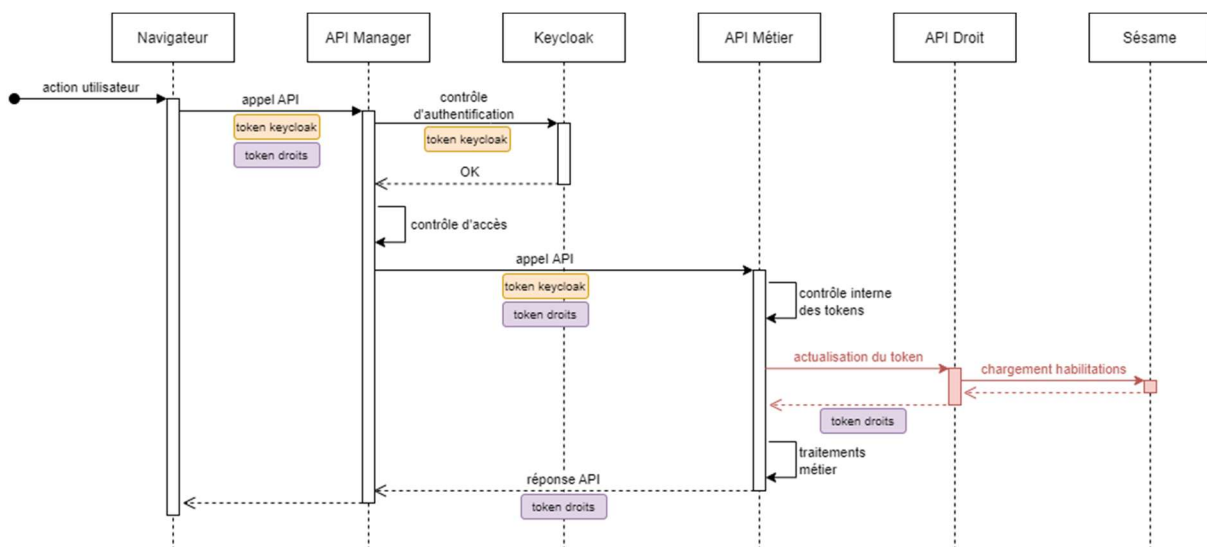
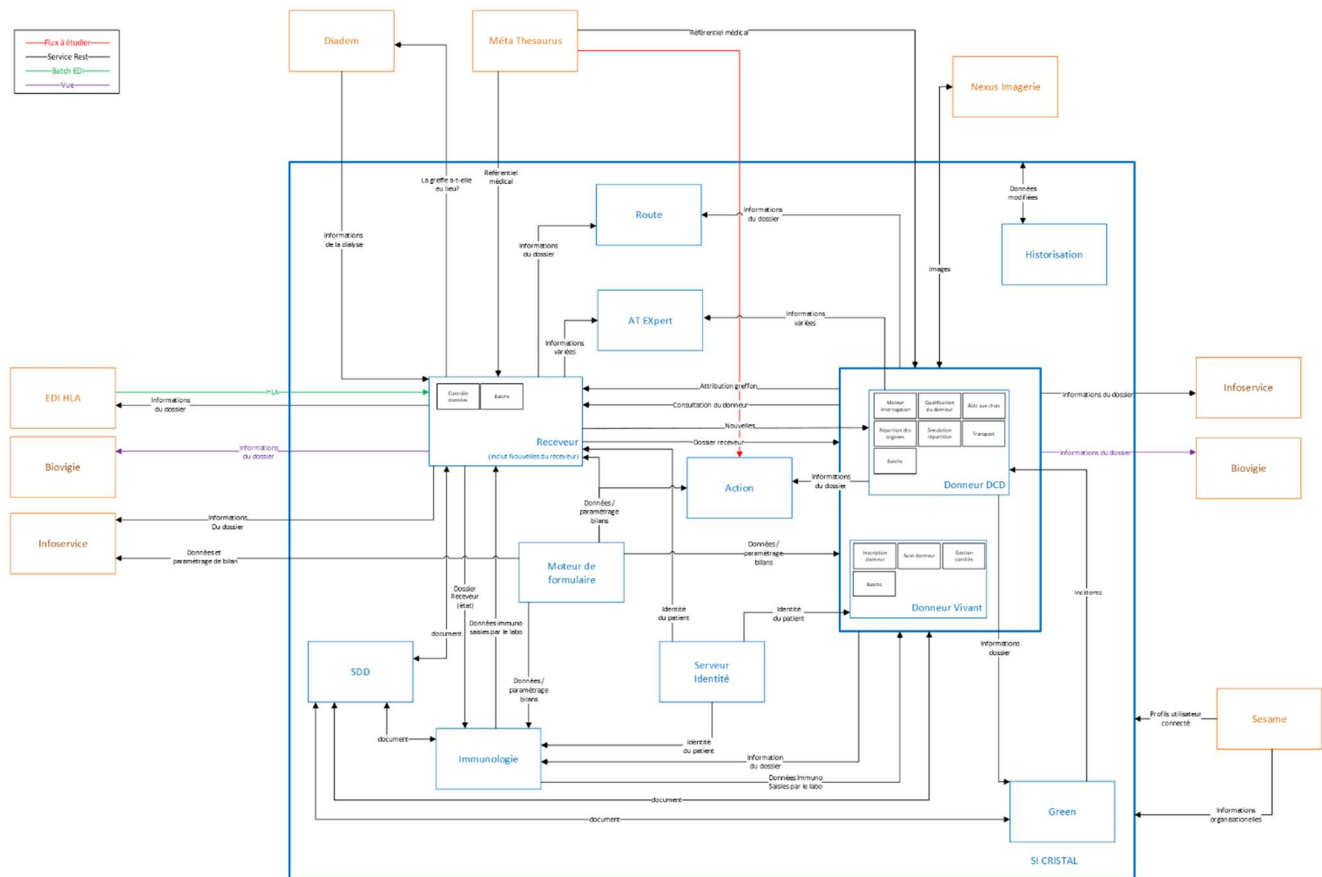


Figure 15 : Utilisation du token d'habilitations

3.3. Interfaces



Interfaces internes

Les applications du SI CRISTAL communiquent entre elles par des appels d'APIs REST. Le protocole utilisé est HTTPS et tous les appels passent par l'*api-manager*.

Interfaces externes

Description générale de l'interfaçage avec les applications hors SI CRISTAL. Les interfaces externes sont ajoutées à ce document lors de leurs mises en place au fur et à mesure de l'avancement de la Refonte.

Application Source	Application Cible	Protocole	Format de données échangées	Lecture / Ecriture	Document d'interface	Description
Applications CRISTAL	SESAME	REST / HTTPS	JSON	Lecture		Référentiel ABM : récupération des acteurs, réseaux, équipes, établissements, etc.
Applications CRISTAL	IU	REST / HTTPS	JSON	Lecture / Ecriture		Référentiel des informations identitaires des patients.
Donneur	Nexus Imagerie	REST / HTTPS	JSON	Lecture / Ecriture		Récupération des images et bordereaux reins
Donneur	Méta Thésaurus	REST / HTTPS	JSON	Lecture		Récupération des données de nomenclature
Receveur	Méta Thésaurus	REST / HTTPS	JSON	Lecture		Récupération des données de nomenclature
Receveur	EDI HLA	Batch	Fichier	Lecture		Intégration des données d'analyses médicales
Infoservice	Donneur / Receveur / Moteur de formulaire		Copie BDD	Lecture		Extraction de données CRISTAL pour le pôle infoservice
Biovigie	Donneur / Receveur		Vues BDD	Lecture		Récupération des données de biovigilance

EDI HLA	Receveur	REST / HTTPS	JSON	Lecture		Intégration des données d'analyses médicales
DIADEM	IU	REST / HTTPS	JSON	Lecture / Ecriture		Référentiel des informations identitaires des patients.
DIADEM	Receveur		Vues BDD	Lecture		Réplication des dossiers patients dans IU

Tableau 4.3.2 : Les flux applicatifs.

3.4. Traitements batch

Information (supprimer l'encart dans la version validée du DAT)

Décrire les batchs de l'application et tous les traitements non interactifs autour de l'application.

3.5. Liste des composants applicatifs et *middlewares*

Information (supprimer l'encart dans la version validée du DAT)

Lister l'ensemble des composants constituant l'application, mais aussi les briques logicielles (base de données, autres applications, . . .) avec lesquelles l'application dialogue. Lister également les logiciels *middlewares* (APIM, Keycloak, Apache, etc.) avec lesquels l'application a des interactions.

Si le composant apparaît dans le schéma d'architecture applicative avec un code, le préciser.

Code	Composant	Technologie	Version	Fin de support
	<i>api-manager</i> : Gravitee	Java	4.2.20	
	Keycloak	Java	19.0.3	
	OpenJDK	Java	17.0.2	30/09/2031
	SpringBoot	Java	3.1.2	18/05/2024 18/08/2025 (+)
	Spring-core	Jaba	6.0.11	31/08/2024 31/12/2025 (+)
	Hibernate	Java	6.3.1	
	Angular		17.3.11	15-05-2025
	PrimeNG		17.18.1	Aligné sur Angular
	Angular-OAuth2-OIDC		17.0.2	
	Oracle			
	Docker Engine		> 24	

Tableau 4.5 : Les composants applicatifs et transverses.

(+) *Support commercial*

L'ensemble des versions ci-dessus dont la date de fin de vie n'est pas précisée dispose actuellement d'un support actif ou a minima d'un support pour des correctifs de sécurité.

5 | GESTION DES ACCES A L'APPLICATION

4.1. Authentification

Information (supprimer l'encart dans la version validée du DAT)

Décrire comment les utilisateurs ou les différentes populations d'utilisateurs s'authentifient pour accéder à l'application : via le Portail ou directement, auprès de Keycloak ou le *backend* qui traite les requêtes d'authentification d'une façon ou d'une autre, etc.

L'accès aux applications de CRISTAL est sécurisé à l'aide du protocole *OpenID Connect* (authorization code flow with PKCE). Dans l'architecture du socle technique de l'Agence le serveur d'autorisation Keycloak réalise l'authentification centralisée des utilisateurs *en mode SSO*, via le Portail SIPG qui les redirige vers l'application demandée en cas d'une authentification avec succès. Le jeton d'accès (*access token*) aux APIs est délivré en conséquence. Ce jeton d'accès est ensuite vérifié par l'API *Manager* lors de chaque appel d'API. Une fois l'utilisateur entré dans le périmètre d'une application le jeton d'accès ayant une courte durée de validité est renouvelé par l'application *frontend* Angular qui s'interface directement avec Keycloak.

Ce processus est détaillé dans le document de référence REF_DAT-01.

4.2. Les appels d'API par les *backends*

Notamment dans le cadre des tâches planifiées, des applications back du SI Cristal s'appellent entre elles. Il est nécessaire dans ce cadre de permettre la récupération du jeton d'accès sans l'action d'un utilisateur final. En effet, toutes les requêtes http(s) doivent être authentifiées.

Pour cela, un client Keycloak dédié, de type *confidential*, a été créé. Il est configuré pour fournir un jeton d'accès (*access_token*) à partir de l'id client Keycloak et le secret associé.

Le jeton d'accès contient l'attribut *code_acteur*. La valeur de cet attribut est paramétrée en dur dans Keycloak. Elle doit correspondre à un code acteur d'un utilisateur 'système' présent dans Sésame.

Lorsque dans le cadre d'une tâche planifiée, une application *back* exécute par exemple une requête SQL. Dans la table d'audit, l'auteur de la modification correspond au code acteur contenu dans le jeton d'accès ou a la valeur de la variable d'environnement : SCHEDULED_TASKS_ID_ACTEUR.

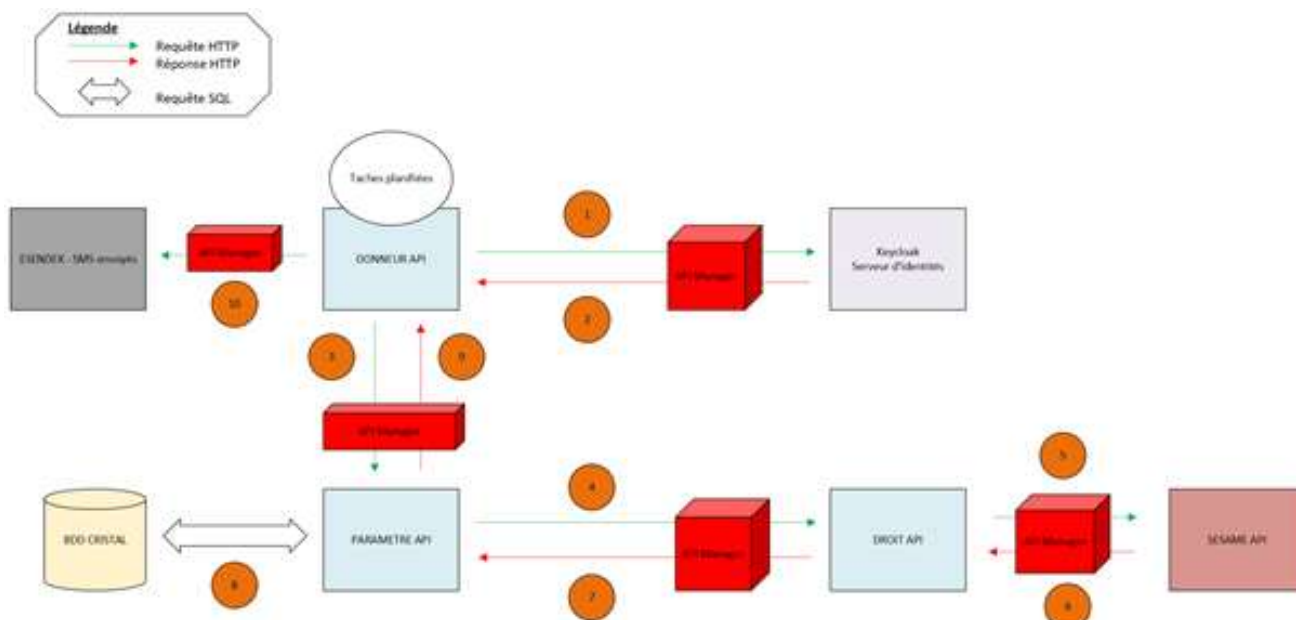


Figure 5.2 : Les appels entre APIs.

4.3. Sécurisation des flux d'API

Tous les appels d'API CRISTAL passent par l'*api-manager* dont le rôle principal est de sécuriser ces appels. Le protocole HTTPS est utilisé pour les appels entrants comme pour les appels sortants vers les API d'applications hors SI CRISTAL. Les appels sont chiffrés à l'aide du protocole TLS.

Les applications *back* (API) de CRISTAL vérifient si le jeton d'accès (*access_token*) est présent dans chaque appel HTTP, si sa date est valide et enfin s'il n'a pas été altéré.

L'*api-manager* se charge de vérifier si le jeton d'accès est légitime auprès de Keycloak, fournisseur des jetons d'accès.

Les entêtes HTTP CORS sont renvoyés dans les entêtes de réponses HTTP pour protéger les services des appels inter-domaines malveillants.

Pour les appels entre des applications du SI CRISTAL, le jeton d'accès fourni par Keycloak sert à l'authentification des requêtes HTTP (ce jeton d'accès contient le code Sésame spécifique à l'utilisateur final). C'est une configuration dans Keycloak qui permet d'ajouter le code sésame dans le jeton d'accès.

Les applications *front* du SI Cristal demandent un jeton d'accès à Keycloak. Ce jeton d'accès est ensuite placé dans le *header* de toutes les requêtes HTTP envoyées vers les applications *back* (API). Keycloak fournit un jeton d'accès seulement si l'authentification de l'utilisateur final est en succès. Lorsqu'une application *back* reçoit une requête HTTP d'une application *front* ou d'une application *back*, le jeton d'accès est propagé aux appels HTTP sous-jacents.

Appels vers le SI CRISTAL par les applications externes

L'API Identité (IU) du SI CRISTAL peut être appelée par des applications externes au SI CRISTAL, tel que Diadem.

L'API Identité est appelée par les applications *front* et *back* du SI CRISTAL. Dans ce cadre, l'API Identité applique des contrôles et vérifications métier liés au SI CRISTAL.

Lorsque l'API Identité est appelée par des applications externes au SI CRISTAL, tel que Diadem, elle ne doit pas appliquer les contrôles et vérifications métier liés au SI CRISTAL.

Pour prouver que c'est un appel HTTP légitime, l'application externe, Diadem par exemple, doit fournir dans les requêtes HTTP vers l'API Identité deux *headers* :

- ☐ Un *header* (nom : Authorization) contenant un jeton d'accès généré par Keycloak lors de la connexion de l'utilisateur final sur l'application *front*. L'*Access Type* du client Keycloak utilisé pour générer le jeton d'accès est *public*.
- ☐ Un *header* (nom : App-Token) contenant un jeton d'accès généré par Keycloak lors de la construction de la requête HTTP dans l'application *back* appelante (Diadem). Ce jeton d'accès est généré via un client Keycloak dont l'*Access Type* est *confidential*. Il est important d'utiliser un client Keycloak spécial « application *back* ». Ce client Keycloak ne doit jamais être utilisé par une application *front*. Ce jeton d'accès permet de prouver que la requête HTTP ne vient pas d'une application *front*, notamment.

Lorsque l'API Identité reçoit une requête http, elle vérifie la présence d'un header appelé : App-Token. Si un *header* existe avec ce nom :

- ☐ L'API Identité vérifie que les jetons d'accès dans le *header* Authorization et App-Token n'ont pas été modifiés et si leurs dates de validités sont valides. (L'*api-manager* doit vérifier auprès de Keycloak que ces jetons d'accès sont légitimes).
- ☐ L'API identité vérifie que la valeur AZP contenu dans le jeton d'accès fourni par Keycloak est présente dans une variable d'environnement positionnée dans l'application *back* API Identité.

Si ces éléments sont corrects, l'API Identité traite la requête HTTP sans contrôle métier. C'est à l'application *back* appelante de vérifier en amont que la demande est légitime. C'est pourquoi il est important que les appels HTTP passent par une application *back* qui vérifie en amont les droits métier.

Sinon, l'API Identité vérifie que le jeton d'accès dans le *header* Authorization n'a pas été modifié et si sa date de validité est valide. (L'*api-manager* doit vérifier auprès de Keycloak que ce jeton d'accès est légitime). Si ces éléments sont corrects, l'API Identité traite la requête HTTP en appliquant les contrôles et vérifications métiers.

4.4. Création et purge des comptes

Information (supprimer l'encart dans la version validée du DAT)

Décrire comment les accès à l'application sont demandés (formulaire du Portail, mail, ...) et traités. Préciser s'il existe une procédure de revue des accès et de purge des comptes inutilisés.

4.5. Gestion des droits / profils applicatifs

Information (supprimer l'encart dans la version validée du DAT)

Décrire la gestion des permissions applicatives : quels sont les profils et à quelles fonctions spécifiques ils permettent d'accéder (Exemple : création, modification de dossiers, etc.).
Si les profils sont dédiés d'un référentiel tel que Sésame, préciser les règles de correspondance.

Lorsqu'un utilisateur accède au SI CRISTAL depuis le portail SIPG, ce dernier exécute une requête HTTP vers l'URL */api-preference/redirect*, qui redirige l'utilisateur (via une réponse HTTP 302) vers une application frontend du SI CRISTAL (application définie dans ses préférences CRISTAL ou application par défaut).

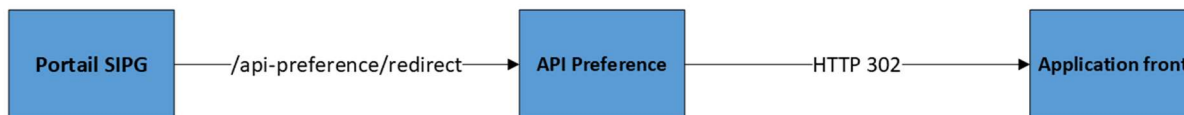


Figure 5.5 : Accès au SI CRISTAL depuis le portail SIPG

Droits d'accès aux fonctionnalités

L'utilisateur accède à une application en étant connecté avec un profil prédéfini (profil défini dans ses préférences ou profil par défaut). À l'initialisation d'une application *frontend*, une requête est envoyée à l'API Droit pour récupérer la liste des droits de l'utilisateur (liste des applications et liste des fonctionnalités de l'application courante auxquelles il a accès).

L'application *frontend* restreint ensuite l'accès aux fonctionnalités en fonction de ses droits :

- Page d'erreur si l'utilisateur n'a pas accès à l'application.
- Redirection vers la page d'accès et/ou message d'erreur si l'utilisateur n'a pas accès à la fonctionnalité.

Droits d'accès aux ressources

Lors de la navigation à l'intérieur d'une application, chaque requête HTTP à une des API nécessite la vérification des droits et habilitations de l'utilisateur connecté.

Le jeton d'accès fourni par Keycloak, ainsi que le profil sélectionné (le profil peut être changé dans le *frontend* par l'utilisateur), sont stockés dans les headers des requêtes HTTP.

Une requête HTTP à une des API entraîne les actions suivantes :

- Appel à l'API Droit :
 - Appel à SESAME : récupération des fonctions de l'utilisateur connecté
 - Vérification que l'utilisateur connecté est bien associé au profil défini dans la requête (à une fonction SESAME correspond un profil CRISTAL)
- Appel à l'API Droit : récupération du type de périmètre associé au profil de l'utilisateur
- En fonction du périmètre :
 - Si la requête concerne une ressource unitaire (cas des requêtes POST, PUT, DELETE, et de certaines requêtes GET) :
 - Vérification que le périmètre de l'utilisateur a bien accès à cette ressource
 - Retour de la ressource ou d'une erreur HTTP 403 si l'utilisateur n'a pas les droits d'accès à cette ressource
 - Si la requête concerne une liste de ressource (requêtes GET) :
 - Ajout d'une clause dans la requête de récupération des ressources
 - Retour de la liste des ressources filtrée selon le périmètre de l'utilisateur.

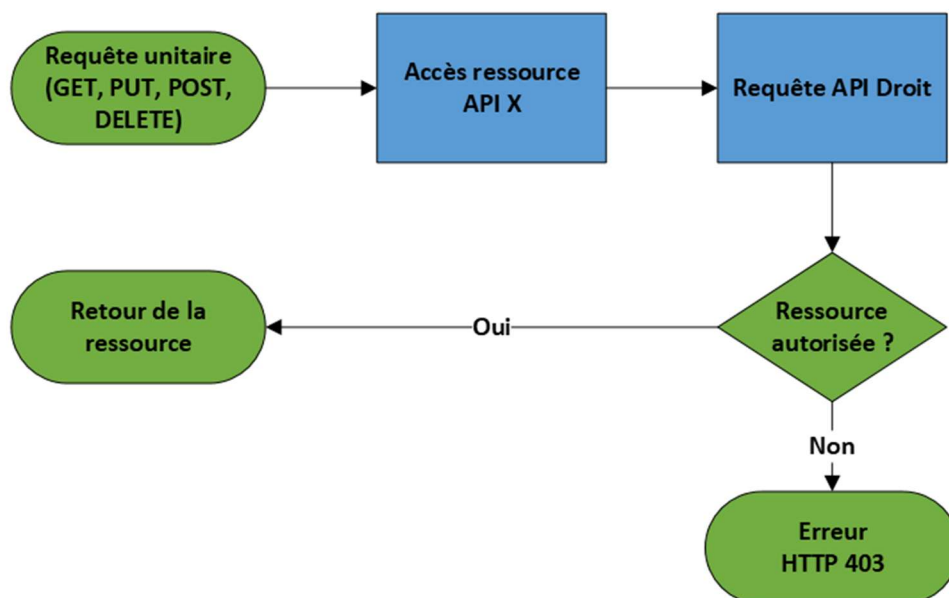


Figure 5.5.2 (a) : Gestion des habilitations, cas d'une requête unitaire.

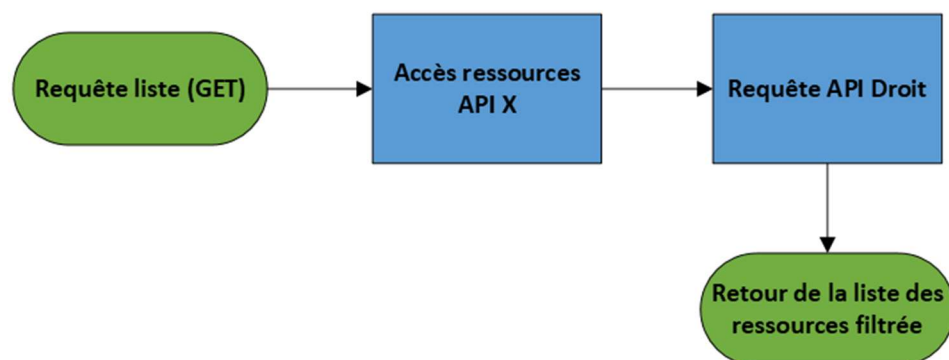


Figure 5.5.2 (b) : Gestion des habilitations, cas d'une requête de liste.

4.6. Administration de l'application

Information (supprimer l'encart dans la version validée du DAT)

Si l'application présente des fonctionnalités pour son administration, les décrire explicitement et indiquer quelle sont les utilisateurs qui y ont accès