Rapport de stage : Développement de tests de services

Ryan RAMASSAMY, 2024-2025

Promotion	:	BUT3FI 2024 – 2025 |USPN Villetaneuse
Maitre de stage	:	Jean-Christian PENNETIER
Tuteur enseignant	:	Pierre GERARD

IUT Villetaneuse
Université Sorbonne Paris Nord

Rapport de stage

Stage chez AIBSS de Orange

Version du document :	1.5	Validé par :	Jean-Christian PENNETIER
Date du document :	13/06/2025	Soumis le :	13/06/2025
		Type de diffusion :	Document électronique (.pdf)
Auteur :	Ryan RAMASSAMY	Confidentialité :	Réservé au corps enseignant de USPN Villetaneuse & Orange

Maître de stage :	Jean-Christian PENNETIER	Tuteur enseignant :	Pierre GERARD

Remerciements

Je voudrais avant tout remercier toutes les personnes qui m'ont conseillé et participé à ma recherche de stage, et plus particulièrement le corps enseignant, qui m'a aidé dans la réalisation de mon CV, ainsi que mes proches, qui ont majoritairement contribué à la trouvaille de cette expérience.

Je souhaite également remercier l'ensemble de l'équipe Architecture et Ingénierie Briques Serveur/ Stockage de m'avoir réservé un tel accueil ainsi que de m'avoir offert l'opportunité de travailler avec eux. J'ai non seulement pu y développer de nouvelles compétences, mais j'ai également pu observer plus attentivement comment se déroule la vie en entreprise.

Et enfin, je souhaite remercier toute la communauté de l'informatique qui m'a aidé dans mon apprentissage ainsi que dans la résolution de problèmes notamment au travers de différents sites tels que OpenClassroom et StackOverflow.

Sommaire

I. INTRODUCTION **5**

II. CONTEXTE **6**

1. Présentation de l'entreprise **6**
A. Orange SA 7
1. Carte d'identité 8
B. Orange France 10
1. Architecture des marques Orange 11

2. Mon environnement **12**
A. Compute Stockage Sauvegarde Archivage (CSSA) - Infrastructures Automatisées (INFRAS) 12
1. OS Factory 12
2. Déploiement et Exploitation Stockage, Sauvegarde, Serveur et Archivage 12
3. Hardware, Server et Stockage 13
4. Centre de Service Déploiement Système Serveurs Châssis 13
B. Architecture et Ingénierie Briques Serveur/ Stockage (AIBSS) - CSSA 15

III. MON STAGE **16**

1.Mes attentes **16**
2.Mon rôle dans l'équipe **16**
3.Ma Mission **17**
A.Contexte 17
B.Objectif 17
C.Ma contribution 20
1. Autoformation 20
2. Prototype 20
3. Refactorisation & Lancement de sauvegarde 21
4. Suivi de statut 24
5. Lancement d'une restauration 27
6. Élaboration des logs & Suite ELK 28
D.Mes attentes ont-elles été satisfaites ? 32

IV.CONCLUSION **33**

V.GLOSSAIRE **34**

ANNEXES **35**

I. INTRODUCTION

Ce rapport de stage présente mon rapport d'activités en tant que stagiaire développeur au sein de l'équipe Architecture et Ingénierie Briques Serveur Stockage (AIBSS) de l'entreprise Orange. Ce stage est effectué dans le cadre de la dernière année de mon Bachelor Universitaire de Technologie informatique à l'Université Sorbonne Paris Nord de Villetaneuse et a pour objectif de me permettre de développer mes compétences en développement informatique au travers de la mission qui m'a été attribuée, afin que je puisse découvrir les différents aspects du métier de développeur et plus précisément centré mes efforts sur l'intégration de tests, ainsi que les différents outils et méthodologies utilisés au sein d'une équipe professionnelle.

Dans ce rapport je vais dans un premier temps, présenter la société Orange ainsi que mon environnement de travail et celui de mon équipe. Dans un second temps, je donnerai des détails à propos de la mission qui m'a été attribuée. J'expliquerai son objectif, son fonctionnement, ma contribution, les différents problèmes rencontrés ainsi que les solutions trouvées pour ces derniers et les éventuels résultats. J'évoquerai également les diverses compétences acquises au cours de cette période de stage et mon ressenti vis-à-vis de cette mission.

II. Contexte

1. Présentation de l'entreprise

Orange est une entreprise de télécommunications française. Elle est aujourd'hui l'un des principaux opérateurs de téléphonie mobile et fixe en France et à l'International. Orange propose une large gamme de services, tels que l'accès à internet, la télévision et la téléphonie. L'entreprise se distingue par son réseau de qualité, sa couverture nationale et son engagement dans l'innovation technologique. Orange est également impliqué dans des initiatives de responsabilité sociale et environnementale, contribuant ainsi au développement durable.

A. Orange SA

Orange est un groupe international de télécommunications qui figure parmi les leaders du secteur au niveau mondial, présent dans 26 pays. À l'origine, Orange est une entreprise de télécommunications Britannique, devenue filiale en 1999 de Mannesmann (conglomérat industriel allemand à la fin du XXe siècle). En 2000, Orange devient la filiale de France Télécom, entreprise publique Française. Progressivement Orange est devenu la marque principale de France Télécom et s'est construit en tant que leader en étendant sa présence en Europe, en Afrique et Moyen- Orient et dans les Antilles.

Orange se positionne sur plusieurs enjeux stratégiques, tels que le déploiement de la fibre optique, des réseaux 4G et 5G ou satellite, et le développement de services numériques innovants (intelligence artificelle, cyberdéfense, services financiers, énergies...).

Sur le marché des télécommunications, Orange opère dans plusieurs segments, offrant des services de téléphonie fixe et mobile, d'internet haut débit, de télévision, ainsi que des solutions pour les entreprises et sert plus de 291 millions de clients à travers le monde. En Afrique et au Moyen-Orient, Orange connaît une croissance rapide, capitalisant sur l'augmentation de la demande en services de communication et d'accès à internet.

Les offres d'Orange sont diversifiées pour répondre aux besoins variés de ses clients. Pour les particuliers, elle propose des forfaits mobiles, des abonnements internet fibre et ADSL, ainsi que des services de divertissement comme Orange TV. Pour les professionnels et les entreprises, Orange Business offre des solutions de connectivité, de cloud computing, de sécurité et de collaboration numérique.

La clientèle d'Oraet ses salariés (4,64 %). Orange possède de nombreuses filiales comme Orange Store, Orange Cyberdéfense, Orange Events...).nge est large, allant des particuliers aux grandes entreprises, en passant par les PME et les collectivités. L'entreprise s'efforce de fournir un service client de qualité et de développer des offres personnalisées pour répondre aux attentes spécifiques de chaque segment de marché. Avec l'évolution constante des technologies et des usages numériques, Orange continue d'innover pour rester à la pointe de son secteur et maintenir sa position de leader.

L'entreprise s'engage également dans les critères E.S.G. : Environnementaux, Sociaux et de Gouvernance (anciennement RSE), avec des initiatives pour réduire son empreinte carbone et promouvoir l'inclusion numérique. Orange développe par exemple sur l'ensemble de son territoire des Digital Center, un programme gratuit et accessible à tous pour se familiariser au monde numérique, développer ses compétences ou se faire accompagner dans un parcours professionnel. Elle vise la neutralité carbone d'ici 2030.

Orange est la première société à avoir inscrit dans ses statuts sa raison d'être : ''Orange est l'acteur de confiance qui donne à chacune et chacun les clefs d'un monde numérique responsable''.

Orange est capitalisé à 26,5 milliards d'€ en 2023, sa Directrice Générale est Christel Heydemann et son Président du Conseil d'Administration est Jacques Aschenbroich. Orange est coté en bourse à EuroNext (Principale bourse Européenne), au CAC40 et NYSE (Bourse de Wall Street, New York), son action vaut 10,60€ (EPA - 04/2024) / 11,52USD (NYSE - 04/2024) et ses principaux actionnaires sont institutionnels (64,3%), l'État français (23 %), particuliers (5 %)

Carte d'identité Orange SA

Chiffre d'affaires 2024

40,3 Mds €

Salariés

127 000

Clients

291 M

Marque mondiale des télécoms

8e

Investissements en R&D

643 M €

26 pays

et une présence mondiale avec Orange Business

Europe

Belgique, Espagne, France, Luxembourg, Moldavie, Pologne, Roumanie, Slovaquie

Afrique et Moyen-Orient

Botswana, Burkina Faso, Cameroun, Côte d'Ivoire, Égypte, Guinée, Guinée-Bissau, Jordanie, Liberia, Madagascar, Mali, Maroc, Maurice, République centrafricaine, République démocratique du Congo, Sénégal, Sierra Leone, Tunisie

Carte d'identité Orange France

Chiffre d'affaires 2023

17,7 Mds €

Collaborateurs

60 487

Années en tant que meilleur réseau mobile

13

Marque Française

5e

Clients abonnés à un forfait mobile

20,8 M

Une année Olympique !

En 2024, Orange met son expertise et sa capacité d'innovation au service du plus grand événement sportif au monde : les Jeux Olympiques et Paralympiques de Paris 2024 !

Évènements sportifs

878

Salariés mobilisés

+1 000

Terminaux "Push to Talk"

13 000

Sites officiels connectés

+120

Kilomètres de fibre optique

400 000

B. Orange France

En France, Orange a remplacé la Direction générale des télécommunications. L'administration est fondée en 1988 sous le nom de France Télécom, l'entreprise n'a été privatisée qu'en 1997 et a adopté la marque Orange seulement après le rachat de celle-ci en 2000. L'ensemble des offres et services ont adopté le nom commercial Orange progressivement jusqu'en 2012 et ce n'est qu'en 2013 que France Télécom a laissé place au nom Orange.

Orange France est le leader incontesté du marché des télécommunications en France, offrant une gamme étendue de services couvrant la téléphonie fixe et mobile, l'internet haut débit et très haut débit, ainsi que des services de télévision et de divertissement numérique. Son réseau de fibre optique est l'un des plus avancés du pays, permettant une couverture large et une qualité de service supérieure, tant pour les particuliers que pour les entreprises.

Orange France s'engage sur plusieurs enjeux stratégiques clés, notamment le déploiement massif de la fibre optique, des réseaux mobiles 4G et 5G et l'internet par satellite. L'entreprise vise à offrir une connectivité de haute qualité partout sur le territoire français, y compris dans les zones rurales et isolées, contribuant ainsi à réduire la fracture numérique. Elle joue également un rôle actif dans l'innovation technologique avec des initiatives en intelligence artificielle, cybersécurité et services cloud.

En termes de concurrence, Orange France se positionne face à des acteurs majeurs comme SFR, Bouygues Telecom et Free. Pour maintenir sa position de leader, Orange investit continuellement dans l'amélioration de son infrastructure réseau et l'optimisation de son offre de services. L'entreprise s'efforce de proposer des offres compétitives et attractives, telles que des forfaits convergents (internet, mobile, TV) et des services personnalisés pour les clients particuliers, professionnels et entreprises.

Orange France est également engagé dans des missions d'intérêt public, collaborant étroitement avec l'État pour assurer une couverture réseau optimale sur l'ensemble du territoire (Orange s'est engagé avec l'Etat dans l'article L. 33-13 sur le déploiement de la fibre). L'entreprise participe à des initiatives nationales pour renforcer la cybersécurité, promouvoir l'inclusion numérique et encourager l'innovation technologique. Elle est un acteur clé dans la mise en œuvre des politiques publiques visant à améliorer l'accès aux services numériques pour tous les citoyens, en particulier les populations vulnérables et les zones sous-desservies (avec son offre solidaire "Coup de pouce").

Comme à l'internationnal, Orange s'est fixé les mêmes objectifs ambitieux en France concernant les enjeux environnementaux, sociaux et climatiques.

Le saviez-vous ?

Orange en 2001 a rendu possible (pour la première fois dans le monde) une opération chirurgicale réalisée à distance entre un chirurgien situé à New York et une patiente à Strasbourg !

1. Architecture des marques Orange

Le plan stratégique Lead the Future, présenté en février 2023 par Christel Heydemann prévoit l'évolution de notre modèle d'entreprise vers plus de simplicité et d'efficacité.

Cela se traduit notamment par la simplification de l'architecture des marques.

Concrètement, Orange a réduit ses différentes typologies de marques. Il en existe désormais 4: la Master Brand, la Descriptor, les Sub Brand et enfin les B Brand.

Elles illustrent le positionnement d'Orange sur ses différents marchés.

Master Brand

Liée à notre coeur de business, la stratégie de marque Orange nécessite l'exclusivité d'utilisation de son logo Orange comme signal principal pour maximiser sa valeur.

Descriptor Brand

Liée à notre plan stratégique, elle illustre une activité particulièrement stratégique pour l'entreprise.

Sub Brand

Liée aux télécommunications ou dans des pays où Orange n'est pas présent pour aborder de nouveaux clients et toujours porter la marque. Elle n'utilise pas le logo mais utilise le nom Orange et adhère aux mêmes éléments d'identification.

B Brand

La marque B est séparée d'Orange, elle représente une valeur bouclier ou d'adressage à des cibles précises.

2. MON ENVIRONNEMENT

A. Compute Stockage Sauvegarde Archivage (CSSA) - Infrastructures Automatisées (INFRAS)

1. OS Factory

L'équipe OS Factory (**OSFY**) est en charge de l'ingénierie des **systèmes d'exploitation** utilisés dans les infrastructures du Groupe Orange. Concrètement, OSFY **construit** et **maintient** les versions officielles des systèmes d'exploitation Red Hat Enterprise Linux, Obuntu (version interne d'Ubuntu) et Windows Server. Elle est également responsable de la **maîtrise** d'œuvre de l'application **APT Platon**, qui fait office de référentiel centralisé pour la gestion des paquets et mises à jour sur les serveurs RHEL du groupe.

En lien avec les impératifs de cybersécurité, l'équipe joue un rôle clé dans le traitement des avis CERT, en évaluant et en intégrant les correctifs nécessaires dans les versions distribuées.

2. Déploiement et Exploitation Stockage, Sauvegarde, Serveur et Archivage

L'équipe A3S-STS est une entité spécialisée dans le **déploiement** et **l'exploitation** des produits CSSA. Elle intervient sur l'ensemble du cycle de vie des solutions techniques, depuis leur mise en place jusqu'à leur **maintien** en condition opérationnelle. Elle est également responsable du bon fonctionnement des environnements existants, ce qui implique la réalisation de mises à jour régulières, la maintenance **proactive** des équipements, ainsi qu'une **veille** permanente sur les aspects de sécurité. Lorsqu'un incident survient ou qu'une situation à risque est identifiée, l'équipe intervient rapidement pour en **limiter** l'impact et **restaurer** le service dans les meilleurs délais.

En complément de ses missions techniques, A3S-STS joue un rôle **d'accompagnement** auprès des clients **internes**, notamment pour la réalisation de plans de reprise d'activité (**PRA**), de **migrations** ou d'intégrations complexes. Elle travaille en étroite **collaboration** avec les mainteneurs, qu'elle pilote opérationnellement, ainsi qu'avec les équipes projets et **agiles**, dans une logique d'amélioration continue des services. Son expertise technique couvre plusieurs domaines clés, tels que le **stockage** sous différentes formes (SAN, NAS, SDS objet et bloc), ainsi que la **sauvegarde**, avec des outils reconnus comme CommVault, Avamar ou NetBackup.

3. Hardware, Server et Stockage

L'équipe **HSS** (Hardware Serveur et Stockage) est en charge de l'architecture et de l'ingénierie des produits d'infrastructure **SBM** (Site Sophia Antipolis, Bagnolet et Montsouris) et **CSSA**.

L'une des principales missions de HSS est de concevoir, construire et industrialiser les produits et solutions CSSA, en tenant compte des besoins fonctionnels et des contraintes techniques. Elle est également garante de leur **évolution** dans le temps, en intégrant régulièrement des améliorations issues de la **veille technologique** et de retours d'expérience.

En parallèle, elle assure un **support** aux équipes opérationnelles pour l'utilisation et la maintenance des solutions CSSA, ainsi qu'un **accompagnement** des clients sur des sujets plus spécifiques, tels que les migrations.

4. Centre de Service Déploiement Système Serveurs Châssis

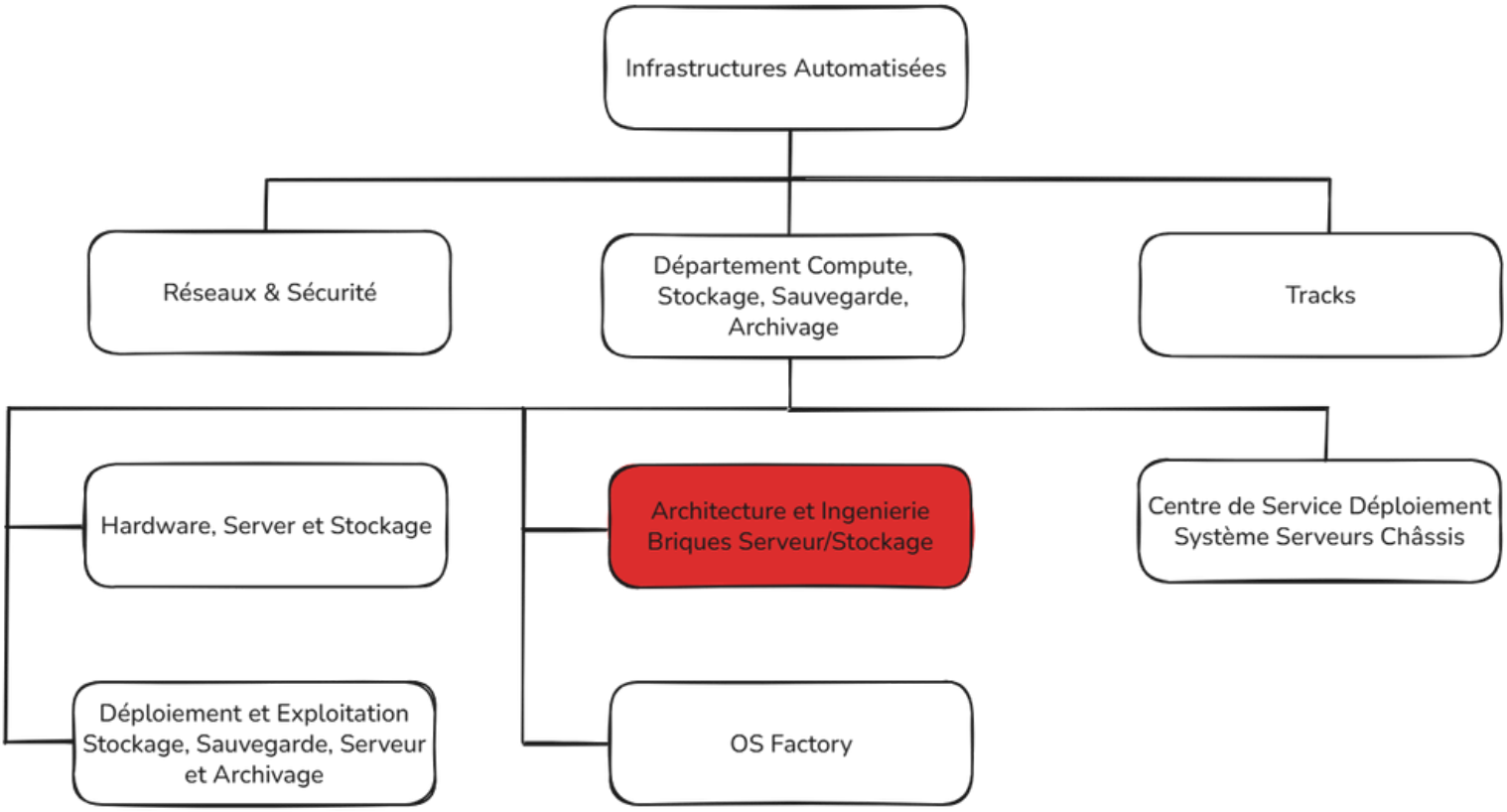
L'équipe **D2SC** (Déploiement Systèmes Serveurs et Châssis) intervient sur toute la chaîne de mise en service des **infrastructures matérielles**, en particulier les serveurs **physiques** et **virtuels**, ainsi que les châssis et blocs intégrés dans les environnements d'hébergement.

Le rôle de D2SC peut être structuré en **trois** grandes phases : Think, Build et Run. Dans la phase **Think**, l'équipe accompagne les projets dès la phase d'ingénierie, en apportant son expertise pour faciliter la réalisation technique des besoins exprimés.

Lors de la phase **Build**, l'équipe est en charge du déploiement système des différents composants matériels : blocs d'infrastructure, châssis, et serveurs, qu'ils soient physiques ou virtuels.

Enfin, dans la phase **Run**, D2SC assure la customisation finale des serveurs au moment de leur mise en production.

ORGANIGRAMME DE INFRASTRUCTURES AUTOMATISÉES



B. Architecture et Ingénierie Briques Serveur/ Stockage (AIBSS) - CSSA

L'équipe dans laquelle jai effectué mon stage est l'équipe **AIBSS** (Architecture et Ingénierie des Briques Serveur et Stockage) quj est responsable de la conception, de l'industrialisation et de l'évolution des produits d'infrastructure CSSA déployés au sein des environnements **PFC** (Plateformes Cloud).

Cette équipe contribue notamment aux appels d'offre pour le choix de **nouveaux composants** ou solutions. Elle assure également un **support technique** aux équipes opérationnelles et aux chefs de projet, notamment lors de l'utilisation et la maintenance des solutions CSSA tout comme l'équipe HSS.

En parallèle de ses responsabilités techniques, AIBSS porte une **vision stratégique** sur l'ensemble de son périmètre. Elle est impliquée dans la planification à **long terme** des évolutions d'infrastructure, dans la définition et le suivi budgétaire, ainsi que dans la gestion de l'obsolescence des composants matériels et logiciels. L'équipe contribue également activement à la **veille technologique**, en identifiant les innovations pertinentes et en évaluant leur intégration potentielle dans le catalogue de solutions de l'entreprise.

Enfin, elle est responsable de la **mise à jour** du catalogue des offres PFC sur ses briques d'infrastructure, en assurant une description **complète** incluant les dimensions techniques, financières et de service.

III. MON STAGE

1. Mes attentes

Avant de commencer ce stage, j'espérais pouvoir en apprendre plus sur **GoLang**, un langage de programmation inspiré notamment du C et du Pascal et également renforcer mes compétences en tant que développeur. J'espérais également pouvoir en apprendre plus sur **la vie d'entreprise**, notamment sur comment se déroulait une journée en open space avec toute une équipe, comment se déroulait les différentes **réunions** ou encore comment s'établissait la **communication entre les membres** d'un groupe. Durant ce stage, j'espère pouvoir acquérir les différentes compétences techniques visées, et en développer de nouvelles, notamment de part le fait que Go est un langage de programmation qui m'est **inconnu** jusqu'à maintenant. Je suis donc évidemment très **fier** de pouvoir travailler avec cette équipe et j'espère atteindre les objectifs que je me suis fixé autant sur le plan technique qu'humain.

2. Mon rôle dans l'équipe

Je fais partie de l'équipe AIBSS en tant que développeur **de tests de services**, ma principale responsabilité était de m'assurer de la qualité et de la fiabilité des API utilisées par le portail du service nommé IASO, en vérifiant leur bon fonctionnement et leur capacité à renvoyer les résultats attendus.

Il est possible de simplifier tout cela en le comparant à la gestion d'un restaurant :

- Le portail IASO correspond à l'aspect visuel esthétique, à la présentation et à l'ambiance de l'application, de manière similaire à la décoration d'un restaurant.
- Le backend, comparable à la cuisine, opère en coulisses pour garantir le bon fonctionnement.
- Les données jouent le rôle des ingrédients. Elles sont préparées en coulisses par le backend avant d'être servies sur le frontend.
- Les API, similaires aux menus d'un restaurant, fournissent une interface pour accéder aux diverses fonctionnalités (plats) proposées par le site.

Ainsi, un développeur de test a pour rôle principal de s'assurer que les services communiquent correctement entre eux, répondent aux attentes fonctionnelles et restent performants sous diverses conditions et est chargé de signaler les anomalies rencontrées afin de permettre au reste de l'équipe d'intervenir le plus rapidement et efficacement.

Et pour continuer dans cette analogie de la restauration, après ces présentations, nous pouvons entamer le plat de résistance, la mission que j'ai effectué.

3. Ma mission

A. Contexte

L'entreprise Orange compte un énorme réseau d'infrastructures réparties dans ces différents **datacenters.** L'organisation a donc mis en place un système de sauvegarde pour ces différentes infrastructures car les données au sein d'un système d'informations sont **vitales** et il est donc nécessaire de protéger ses infrastructures des différents **risques** en sauvegardant ses données, chose souvent oubliée. On compte aujourd'hui plus de 50 000 machines virtuelles dans ces infrastructures. Il est donc primordial de s'assurer que, lorsque l'on souhaite effectuer une sauvegarde sur l'une d'elles, cela se fasse sans qu'il y ai de soucis.

A l'heure actuelle, il existe 3 types de sauvegarde : la sauvegarde **système**, la sauvegarde **applicative** et la sauvegarde **DATA**. Il est possible d'effectuer la sauvegarde de 3 types d'infrastructures différentes : les **machines virtuelles**, les **machines physiques** (serveurs Standalone) et les **filers de stockage**. Les offres de sauvegardes peuvent différer en fonction de l'équipement (cf **Offres de sauvegarde**).

L'équipe a mis en place l'offre convergente **Backup as a Service** (BaaS) qui vise à protéger et restaurer ses données en toute **autonomie** et de manière **intuitive** (as a Service). La solution **Commvault** est alors présentée comme une solution qui permet de couvrir l'ensemble des besoins et d'apporter de la **flexibilité** (cf **Commvault**). Le portail IASO confère l'accès à un panel d'API. Cette interface graphique a été développé pour la solution Commvault. Le portail permet notamment de lancer des sauvegardes ainsi que des restaurations.

B. Objectif

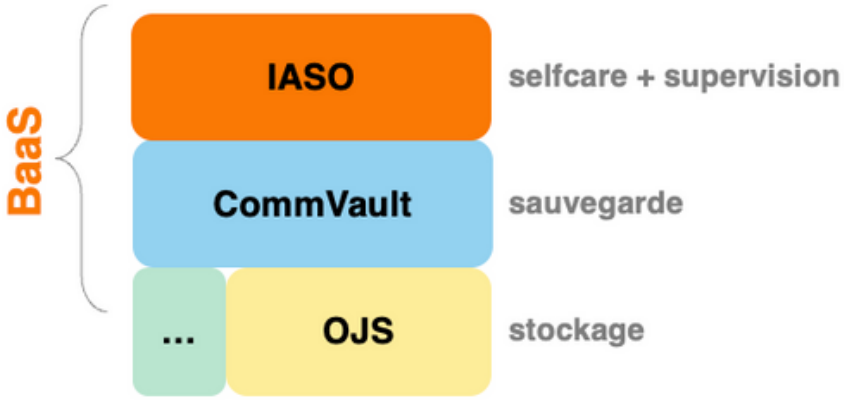
Le problème qui se pose est que le nombre de machines sauvegardées étant particulièrement important, il va de soit qu'il puisse parfois y avoir des problèmes lors des sauvegardes de certaines machines ou que les différents services puissent momentanément être indisponible ou qu'ils n'effectuent pas l'action demandée.

Ma part de travail dans tout ce processus est de m'**assurer** que les API proposées par IASO "fonctionnent bien". Je dois m'assurer que ces dernières renvoient bien les résultats **attendus** en effectuant des **tests de services** de bout en bout. Cela revient à demander à l'API si pour telle machine, est-ce que cette action là s'est bien passée ou s'il y a un problème ou encore de demander à l'API si tel service peut être rendus ou non.

Cela signifie donc qu'il est à la fois question de **superviser** le bon fonctionnement des API, mais également de superviser le bon fonctionnement le l'application qui permet la supervision de ces API.

L'ensemble de ces objectifs vise à **maintenir** une qualité optimale des services fournis par les API IASO, garantissant ainsi une expérience fiable et sans interruption pour les utilisateurs. L'ensemble de mon travail permettra alors d'avertir l'équipe en cas de problèmes avec l'une des API et ainsi permettre une intervention plus rapide et efficace en fonction de l'incident.

Ci-dessous, on retrouve un schéma représentant l'offre BaaS et ses composants.



Commvault est un **logiciel** de sauvegarde et de restauration qui a été développé pour les **entreprises**. Ce dernier repose sur une architecture à plusieurs briques.

Tout d'abord, le **Commserve**. Il est le point d'entrée, la tête pensante de toute l'architecture. Il s'agit d'un **serveur** sur lequel est installé Commvault, mais également mis en place un **web serveur** qui permet d'accéder à la console Commvault sur le **web**. Enfin il contient également la base de données **MicrosoftSQL**. Toute action du client passe par le Commserve

Ce Commserve est rattaché plusieurs grids de 2 ou 4 **Media Agents**. Un Média Agent est un serveur qui sert de **passerelle** entre les clients et les équipements de stockage. Il contient également une base d'index, qui fonctionne un peu comme les index d'une base de données, elle permet d'indiquer au Commserve où se trouve la data dans les équipements de stockage en cas de demande de restauration. Il contient une base de **déduplication**, qui permet notamment l'**optimisation** de l'espace disque. Prenons pour un exemple la sauvegarde de deux fichiers. Le premier fichier est composé de plusieurs blocs que nous appellerons "1, 2, 3 et 4" et il est sauvegardé, la base de déduplication contiendra alors ces 4 blocs qui seront ensuite envoyés vers les équipements de stockage. Le deuxième fichier lui contient des blocs "2, 3, 4 et 5". La base de déduplication contenant déjà les blocs 2, 3 et 4, seul le bloc 5 sera sauvegardé et envoyé vers le stockage, ce qui entraîne une **réduction** du nombre de données transférées sur le réseau. Cela permet également de ne compter que 5 blocs sur le stockage au lieu de 8, on **économise** donc de l'espace disque.

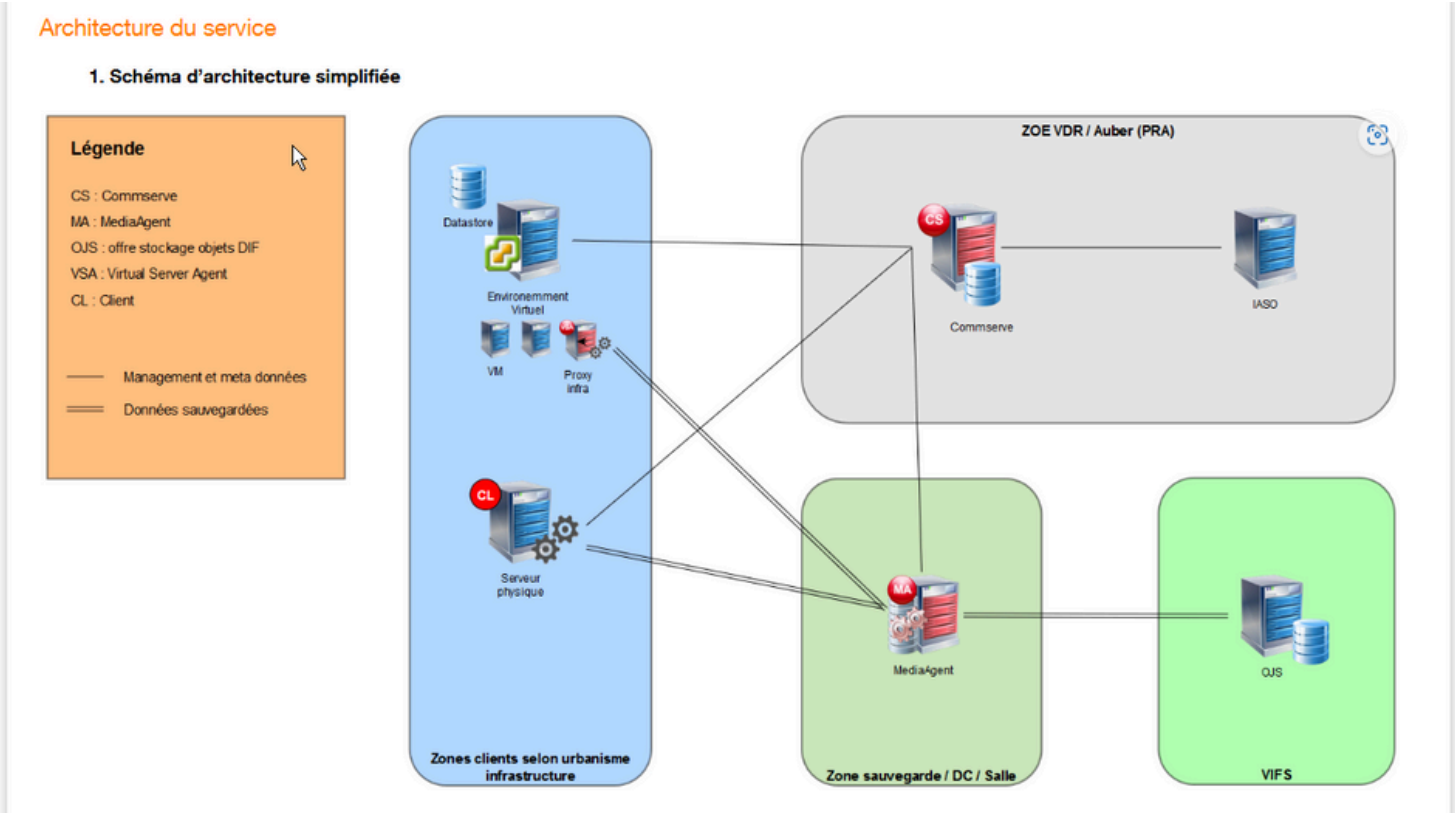
Les équipements de stockage reposent sur l'offre **OJS** qui est l'offre de Stockage **Objet** de PFC qui consiste à stocker la data sur un ensemble de serveurs. On compte plusieurs **Peta** de Sotckage.

Afin de lancer une sauvegarde pour une machine **virtuelle**, le Commserve communique avec le **VCenter**, l'orchestrateur **VMWare**. Un agent virtuel Commvault (machine virtuelle) installé sur l'environnement virtuel **communique** les données a sauvegarder avec le Media Agent qui fait ensuite la **liaison** avec les équipements de stockage.

Dans le cas d'un serveur **physique**, il y a un autre type de client Commvault qui est installé sur le serveur, qui communique egalement avec le Media qui fait la liaison avec les équipements de stockage.

Le role de IASO dans tout cela est de permettre à un utilisateur **externe** de pouvoir lancer une sauvegarde ou une restauration en **autonomie** en interrogeant le Commserve a l'aide d'**API** via une interface **web**. Par rapport à la console web et aux API natives de Commvault, IASO permet donc aux utilisateurs de bénéficier d'une interface **simplifiée** ainsi que de de s'**intégrer** avec le SI d'Orange afin de récupérer les informations sur l'utilisateur et sur la machine à sauvegarder/restaurer, entre autres pour gérer les droits, pour aiguiller les alarmes de supervision et pour générer des rapports.

Ci-dessous, on retrouve un schéma simplifié de l'architecture de Commvault.



C. Ma contribution

Le portail IASO avait **déjà** été **développé** et mis en production. Il permet de lancer la sauvegarde d'un fichier dans une machine virtuelle, ou de **sauvegarder** une VM complète. Il est également possible de lancer une **restauration**. Durant cette mission, qui est un projet durant lequel j'ai, pour la plupart du temps, travaillé sous les conseils et consignes de mon tuteur, mon travail est de développer une application de **supervision** des API IASO. Mon rôle est de développer cette application de bout en bout, du "prototype" à la version opérationnelle.

1. Autoformation

Au préalable, j'ai entrepris une **autoformation** sur le langage Go, également connu sous le nom de Golang. Go est un langage de programmation open source développé par Google, conçu pour être simple, efficace et performant. Là où certains langages, tel que Python par exemple, sont des langages **interprétés**, Go est un langage **compilé**, ce qui lui permet d'offrir des performances supérieures et un temps d'exécution plus rapide. Grâce à des ressources en ligne, des tutoriels et des projets pratiques, j'ai pu mettre en application mes connaissances et cela a énormément enrichi mes compétences techniques.

2. Prototype

Ensuite vient le début du développement de l'application, j'ai commencé par apprendre à comprendre le fonctionnement de l'infrastructure de IASO, ce qui inclut donc la découverte des différents **environnements** de travail : le "LAB", la "Qualif" et la "Prod" qui ont chacun leur utilité : la Prod est l'environnement de **production**, le LAB permet d'effectuer des tests et d'essayer de nouvelles choses et la Qualif elle permet de **fusionner** l'ensemble des fonctionnalités développées par l'ensemble de l'équipe afin de vérifier que tout fonctionne avant de mettre en production. Parmi ces trois environnements, j'ai développé mon prototype en utilisant le LAB. Il a ensuite fallu mettre en pratique les notions de Go que j'ai appris durant mon autoformation sur ce langage de programmation. J'ai alors commencé par développer un premier test appelé "Test de vie" qui consiste tout simplement à vérifier qu'une API est bien **opérationnelle,** c'est-à-dire qu'on interroge l'API et on s'assure qu'elle nous "répond". Pour cela j'ai utilisé le package "http" de Go afin de pouvoir effectuer une **requête** ainsi que le package "io" afin de lire le fichier JSON contenant tout les url des différents environnements et ensuite appeler l'API IASO "test de vie".

L'objectif étant ensuite de vérifier la réponse de l'API et s'assurer qu'elle renvoie bien le résultat attendu en l'occurrence pour ce test là cette dernière est censé renvoyer un "**statusCode**" de 200. Pour cela, j'utilise le package "io" pour **décoder** la réponse qui est également au format JSON et en extraire le code afin de pouvoir ensuite l'interpréter. Afin de savoir ce qu'il se passe et pouvoir afficher cette réponse, j'ai décidé d'afficher les opérations en cours ainsi que les réponses obtenues, ce qui permet ensuite de faciliter le **deboggage**.

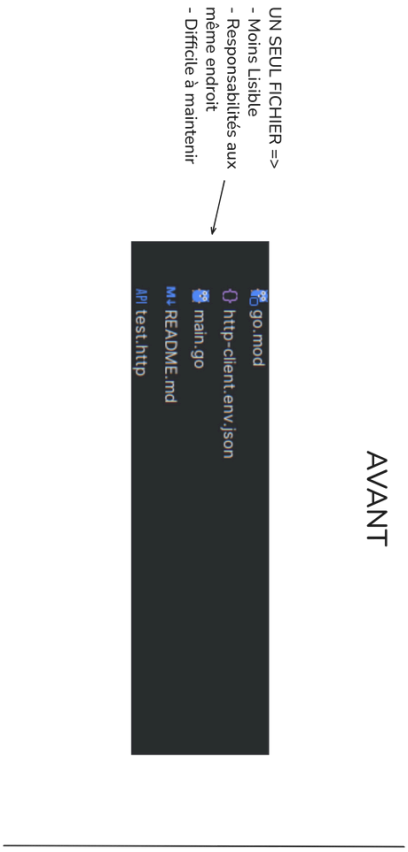
3. Refactorisation & lancement de sauvegarde

Cependant cette première version du prototype a plusieurs **problèmes** notamment au niveau de l'organisation du code. Une **refactorisation** du code étant nécessaire, je me suis penché vers le code **existant** du **broker** IASO qui m'a été fourni par les membres de l'équipe ainsi qu'un **autre** projet contenant un ensemble de requêtes HTTP qui interrogeait les API IASO. De là est alors partie la prochaine étape de mon projet où je vais devoir **analyser**, comprendre et m'inspirer des fonctions déjà mises en place. Premièrement, qui dit refactorisation dit que je dois "alléger" ma fonction main et séparer les responsabilités en redistribuant les différentes fonctions dans d'autres package.

Le premier code est bien fonctionnel mais il ne couvre qu'un seul test et est donc encore trop **spécifique** et donc inutilisable pour d'autres cas. L'objectif est alors de pouvoir couvrir les autres cas. Pour cela, il faut commencer par remplacer la requête très ciblée par un moyen plus **général** afin de pouvoir couvrir l'ensemble des cas de tests. Cela m'a ensuite permis de développer quelques tests supplémentaires et de pouvoir **séparer** les **responsabilités** dans des packages distincts. Cette manœuvre a donc permis d'alléger la fonction main de mon code et donc d'apprendre les bonnes pratiques de développement. J'ai également pu développer deux autres tests de base qui sont le "test de santé" ainsi que le "response time test". Le premier permet d'avoir un aperçu plus détaillé et précis de l'état et la disponibilité des composants de IASO et des interconnexions avec les services tiers consommés par IASO. Il est possible d'obtenir un JSON détaillé de tout les services accompagné de la réponse "healthy" en cas de réussite ou "unhealthy" dans le cas contraire. Le second permet d'exiger une réponse de l'API après un délai d'attente paramétrable.

Remarque : Le test de temps de réponse n'est pas forcément un test intéressant pour le projet mais il a été développé afin que je puisse mettre en pratique les nouvelles notions apprise

Le résultat de cette refactorisation est visible dans la comparaison ci-dessous.



Après avoir développé ces deux tests, je suis alors passé au vif du sujet, les **sauvegardes**. Cependant il a fallu que je développe un autre test au préalable, le test d'**authentification**. Car le but de ce projet est de **simuler un processus** dans lequel l'utilisateur doit donc s'authentifier auprès de IASO avant d'accéder aux services de sauvegarde et de restauration. Il est important de préciser qu'une requête HTTP doit respecter un **protocole** précis, comprenant une **méthode**, une URL, des en-têtes appropriés, et éventuellement un corps.

Dans les cas des tests précédents (Test de vie, Test de santé, Test de temps de réponse), on utilisait la méthode **GET**, qui est utilisée afin de **récupérer** ou **obtenir** des informations, tandis que l'authentification et la sauvegarde nécessitent l'utilisation de la méthode **POST**, qui permet d'**envoyer** des données au serveur pour démarrer une action.

Cette étape d'authentification à l'API IASO, permet d'assurer la sécurité et la validité des opérations. Cette fonction d'authentification a pour rôle la recherche d'un **token** que l'on pourrait considérer comme un **clé** de porte. Lorsqu'une personne veut entrer dans une maison, elle doit présenter cette clé. Si la clé est **correcte**, la porte s'ouvre et la personne peut **accéder** à la maison. De même, dans le monde numérique, le token permet de vérifier que la personne ou l'application a bien le **droit** d'accéder à un service ou à des données. Ce token garantit donc une authentification fiable. Le cas où les identifiants sont incorrect ou que l'authentification échoue pour une quelconque raison, aucun token n'est obtenu et il est donc **impossible** d'effectuer une sauvegarde ou d'utiliser un autre service (cf **backup**).

Dans le cas où le token est obtenu, le processus de sauvegarde peut être lancé grâce à une fonction que j'ai développé qui permet d'effectuer une requête POST en y incorporant un **JSON** contenant le nom de l'hôte et le niveau de sauvegarde afin de lancer l'opération. La réponse a cette requête nous fournit un statut permettant de savoir si la sauvegarde a bien démarrée ou non ainsi qu'un identifiant qui servira par la suite à suivre l'avancement de la sauvegarde.

Après cela, l'étape suivante a donc été de pouvoir lancer **plusieurs** sauvegardes à la fois. Pour cela, l'idée a été d'utiliser un fichier de configuration en JSON afin d'y renseigner les **hostnames** des machines ainsi que le **niveau de sauvegarde** souhaité pour chaque machines. L'objectif étant par la suite de pouvoir effectuer ce test soit sur une machine **spécifique**, soit sur un **ensemble** de machines choisies et donc renseignées dans le fichier de configuration.

En définissant un fichier JSON de test avec différentes machines, il a évidemment fallu que je fasse le necessaire afin de pouvoir lire, **charger** et **utiliser** les données présentes dans ce fichier afin de les **exploiter** pour mes cas de tests. Il a également fallu **séparer** correctement les **responsabilités** entre la fonction permettant le chargement de la configuration et la définition de la fonction permettant de lancer une sauvegarde.

Cela permet alors l'amélioration de la **flexibilité** du processus de sauvegarde permettant de définir **dynamiquement** la liste des hostnames à tester si jamais on souhaite cibler une machine spécifique par exemple. Cette approche permet donc de **faciliter** la gestion et la mise à jour des machines sans pour autant devoir effectuer de modification dans le **code source**, ce qui peut être notamment bénéfique si l'on a un nombre plutôt **conséquent** de machines à traiter ou que l'on doit sélectionner des machines différentes assez **fréquemment**.

4. Suivi de statut

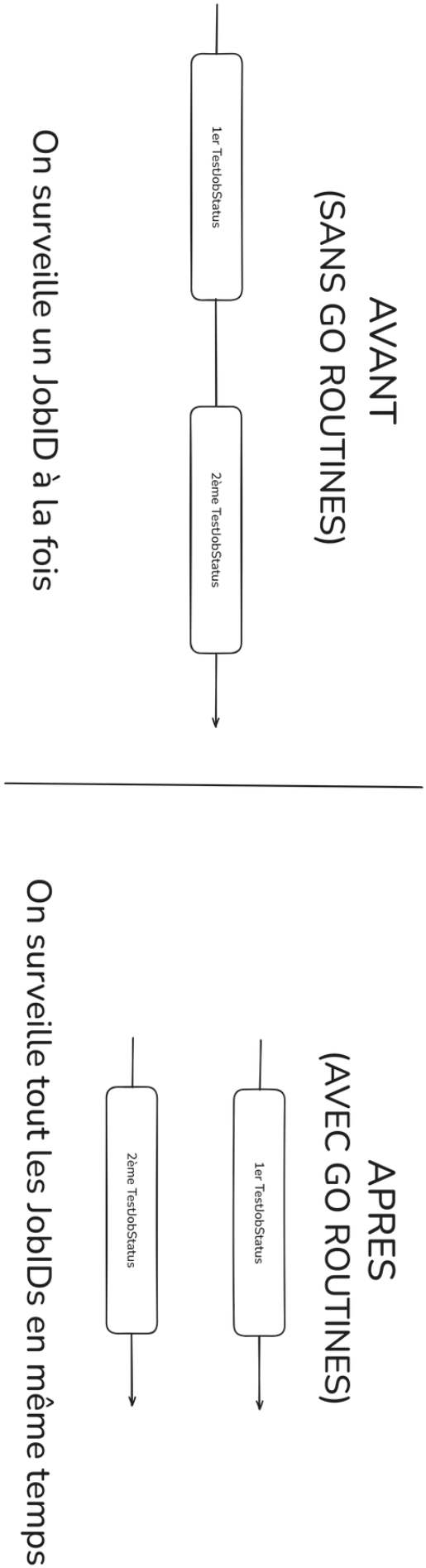
Une fois la sauvegarde lancée, il est essentiel de suivre l'**avancement** de cette sauvegarde et lorsque l'on en effectue une, on obtient un identifiant pour cette action que l'on appelle un "**JobID**". Le suivi de l'état du JobID associé est nécessaire pour garantir la **réussite** de l'opération ou **intervenir** en cas de problème. J'ai alors développé une fonction permettant de **récupérer** le statut d'un job spécifique en envoyant une requête POST à l'API IASO, en renseignant donc le JobID que l'on souhaite surveiller. Le corps JSON est construit dynamiquement pour cibler le bon job et contient donc le nom de la machine également.

La réponse obtenue nous permet d'avoir un **récapitulatif** de l'opération contenant notamment le statut, le pourcentage d'avancement, le temps écoulé depuis le début de l'opération et encore plein d'autres informations. Le premier **problème** concernant ce cas de test est que certes, j'obtiens un résultat qui montre le statut du job en cours, mais il ne renvoie le job qu'**une seule fois** à son démarrage et donne donc des informations **incomplètes**, ce qui est logique car une requête donne une réponse. Le défi est alors de trouver une alternative afin de surveiller le job **jusqu'à la fin**.

J'ai alors choisi de **répéter** les appels à la fonction avec le même jobID et le même hostname et donc effectuer plusieurs requêtes à **intervalle de temps régulier** jusqu'à ce que le job soit complété à 100% ou alors le stopper si une erreur est détectée (cf **suivi du job**).

Cependant un autre problème est alors apparu. En effet, dans le cas où nous souhaitons surveiller **plusieurs** jobs, nous sommes contraints d'attendre que la surveillance du premier job soit terminée afin de pouvoir démarrer celle du job qui suit. Afin de régler ce problème, j'ai choisis d'utiliser les **goroutines**, qui permettent d'exécuter plusieurs fonctions **en parallèle** (on parle ici de parallélisme). La solution apportée consiste à utiliser une goroutine afin de surveiller **chaque** job créé.

Cette approche permet de surveiller plusieurs jobs en parallèle, **optimisant** ainsi le temps de traitement et évitant que l'attente d'un seul job ne bloque la surveillance des autres. Cette méthode permet certes d'améliorer la performance du processus de surveillance, mais il est important de garder en tête que cela peut aussi causer une **surcharge** car cela peut causer un très grand nombre de requête en même temps selon le nombre de jobs. Cela peut être comparé à un supermarché. Il faut plusieurs caisses afin de pouvoir gérer plusieurs clients en même temps.


AVANT
(SANS GO ROUTINES)
1er TestJobStatus
2ème TestJobStatus
On surveille un JobID à la fois
APRES
(AVEC GO ROUTINES)
1er TestJobStatus
2ème TestJobStatus
On surveille tout les JobIDs en même temps

5. Lancement d'une restauration

Enfin, la dernière étape du processus de test concerne la **restauration**. Il existe différents types de restauration, il est possible de restaurer une machine **complète** ou alors de **cibler** un fichier en particulier. Dans le cas de mon stage, je me suis concentré sur la restauration d'un fichier cible.

Prenons pour exemple le cas où vous avez un fichier de configuration pour une application, ou même un fichier quelconque et que vous souhaitez y apportez des modifications, le cas de sauvegarde que nous avons évoqué précédemment vous permet alors par la suite de restaurer **rapidement** et **efficacement** la version précédente de ce fichier ou de cette configuration en cas de besoin.

Grâce à la sauvegarde régulière des fichiers, il devient possible de revenir à un état **antérieur** en cas d'erreur, de corruption ou de modification non souhaitée. La restauration permet alors de **corriger** des erreurs qui peuvent causer le dysfonctionnement d'une application ou même de la machine complète.

Pour cela, j'ai développé une fonction permettant d'effectuer une requête POST vers l'endpoint prévu à cet effet en y renseignant une nouvelle fois les informations requises dans le corps de la requête telles que le nom de l'hôte, le nom du fichier que l'on souhaite restaurer ainsi que le chemin où il se trouve et la destination où l'on souhaite restaurer le fichier.

A l'aide des précédentes fonctions développées, il a été assez simple de développer cette dernière.

6. Élaboration des logs & Suite ELK

Enfin, la dernière étape consiste à récupérer et exploiter l'ensemble des résultats de ces tests sous la forme de **logs** afin de pouvoir garder une **trace** de ce qui a été réalisé et le résultat qui a été obtenu. Cela permet aussi de pouvoir prendre du **recul** par rapport aux fonctionnalités qui ont été développées sur le broker IASO et de pouvoir mettre en avant les services qui rencontrent le plus de problèmes.

Cependant, à l'origine, utilisant un environnement de développement intégré **(IDE)** appelé GoLand, les résultats étaient affichés dans la **console** de l'IDE et étaient donc temporaires car ils étaient perdus lorsque je souhaitais lancer une nouvelle séquence de tests. Il a alors fallu que je passe de cette sortie dans la console à la création d'un **fichier** de log afin d'y mettre tout les résultats des tests.

Ce fichier permettrait alors de pouvoir **garder** les résultats de l'exécution et de ne plus avoir le soucis de perdre ces derniers à chaque nouvelle exécution.

Mais ce n'est pas tout, ce projet étant une application de supervision, il faut prévoir le cas où le problème rencontré provient de l'application de supervision et non pas de IASO. L'idée a alors été de fournir **deux** fichiers de logs **distincts**. Le premier au format texte, constituant un fichier de log applicatif, permettant de surveiller l'application, et un autre fichier de log au format JSON cette fois afin de garder les résultats des tests.

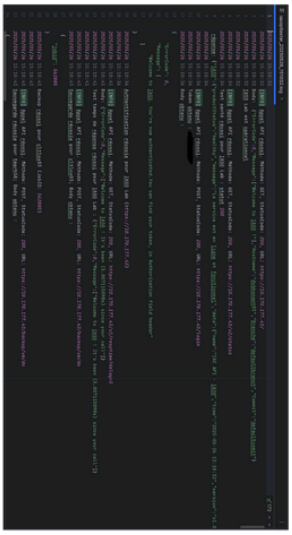
Il a alors fallu revoir la manière de **formater** les logs pour le format JSON car c'est un langage qui nécessite une **structure** bien définie pour assurer une lecture et une analyse efficaces et il est donc important que les résultats des différents tests soient représentés de manière simple afin de ne pas s'éparpiller.

De là a commencé un travail de réflexion afin de décider quelles valeurs allaient être mises en avant, comment on allait faire pour afficher un maximum d'informations tout en évitant d'en perdre un maximum aussi. L'objectif est d'ensuite permettre à l'équipe d'avoir les logs applicatifs, plus adaptés et **compréhensible** pour l'humain, et de convertir les logs au format JSON en un **tableau de bord** à l'aide la suite ELK.

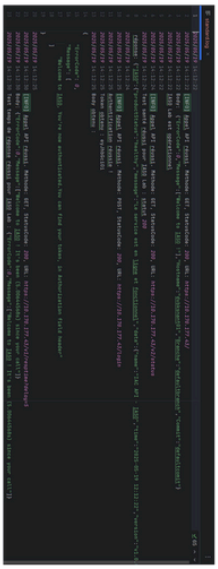
1ère phase

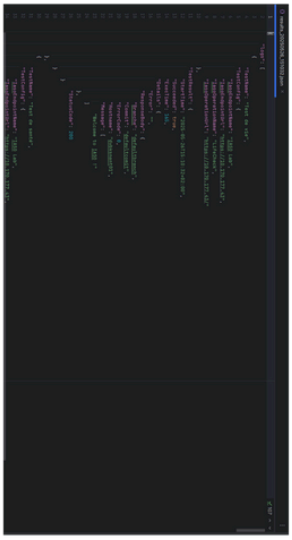
```
25 go setup calls>
Test de l'API : IASO Lab à l'URL https://10.170.177.43
Rapport chargé avec succès

Process finished with the exit code 0
```

2ème phase



3ème phase





La suite ELK (Elasticsearch, Logstash, Kibana) est une solution puissante pour la gestion et l'analyse de logs. Elle permet d'**ingérer**, de **stocker** et de **visualiser** des données en **temps réel**. Lorsqu'il s'agit de fichiers JSON, Logstash peut être configuré pour **parser** ces fichiers en utilisant le filtre `json` et va permettre la collecte, la transformation et l'envoie des logs vers Elastic qui va stocker les logs dans un format souple et scalable. Cela facilite l'extraction des champs et leur **indexation** dans Elastic, rendant les logs facilement consultables et **exploitables** via Kibana qui permet ensuite de **requêter** les données stockées dans Elastic et de construire des **graphes** et des **tableaux de bord**. La flexibilité de cette architecture permet d'adapter l'analyse aux besoins spécifiques de chaque environnement. Dans le cadre de mon stage, je me suis servi d'Elasticsearch ainsi que de Kibana.

Elasticsearch, créé en 2004, est un moteur de recherche et d'analyse distribué et **open source** basé sur la bibliothèque Lucene. Il permet de stocker, rechercher et analyser de **grandes quantités** de données rapidement et en temps réel. L'indexation et la recherche des données s'effectue à partir d'une **API REST**. Les données échangées sont au format JSON. Elasticsearch est souvent utilisé pour la recherche de texte intégral, la surveillance des systèmes, l'analyse de logs et la gestion de métriques. Sa capacité à indexer des documents JSON en fait un composant clé dans la suite ELK, facilitant la recherche en temps réel et la visualisation des données.

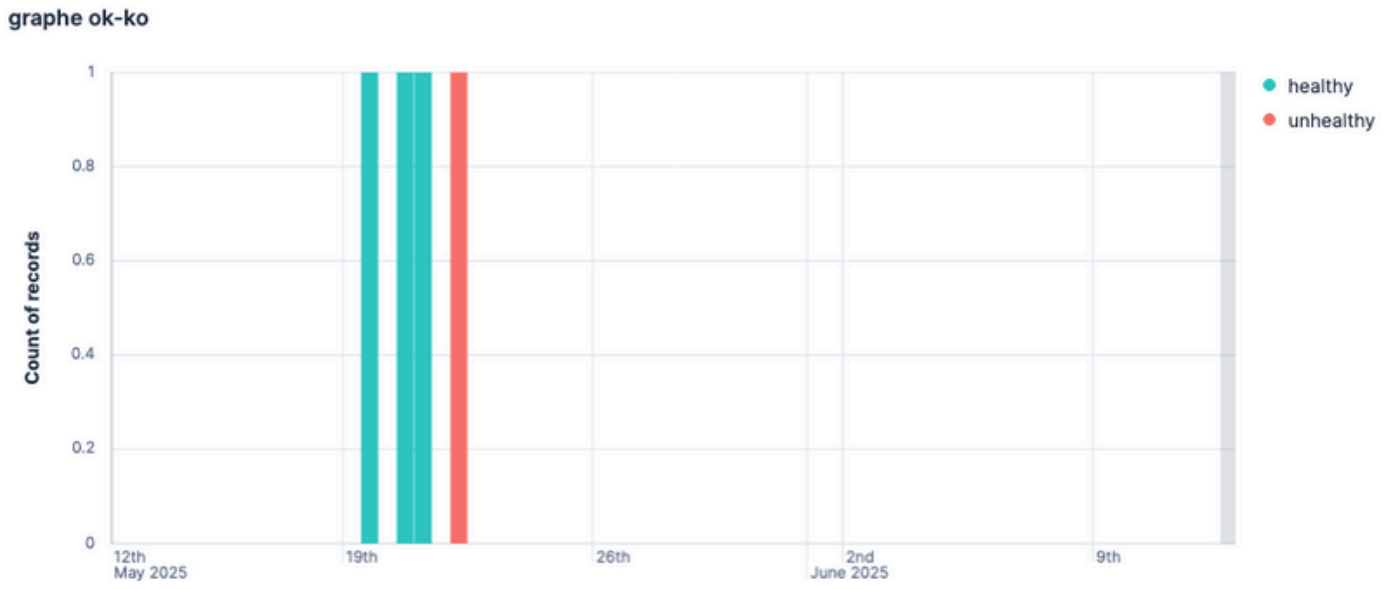
Kibana est une plateforme de **visualisation de données** conçue pour fonctionner en complément d'Elasticsearch et qui a été publié sous la licence libre Apache version 2. Elle permet d'avoir une interface utilisateur **intuitive** permettant de créer des tableaux de bord interactifs, des graphiques, d'autres moyens de visualisations afin d'**analyser** les données stockées dans Elasticsearch. Grâce à ses fonctionnalités, Kibana permet aux utilisateurs de comprendre et analyser plus facilement les logs ou autres types de données et permet donc de **localiser** les anomalies.

J'ai alors été mené à installer ces deux composants. La première étape a consisté à installer Elasticsearch, ce que j'ai réussi à réaliser avec succès. L'installation s'est déroulée sans difficulté majeure en suivant le tutoriel donné sur le site de Elastic, et j'ai pu vérifier le bon fonctionnement du moteur de recherche. Par la suite, j'ai également installé Kibana. Bien que l'installation ait été effectuée correctement, je n'ai pas réussi à établir la communication entre Kibana et Elasticsearch. Malgré plusieurs tentatives de configuration, le dialogue entre les deux composants ne s'établissait pas, ce qui empêchait la visualisation des données dans Kibana.

Il est important de noter que les autres membres de l'équipe ne maitrisaient pas non plus l'outil dans son intégralité, mais possédaient les **bases**, ces derniers ayant fait une **formation** sur la suite ELK. Ils m'ont également conseillé d'opté pour une **solution alternative** en utilisant **WSL** (Windows Subsystem for Linux). J'ai déployé l'environnement ELK dans des **containers**, ce qui m'a permis de contourner les problèmes de compatibilité et de configuration. Cette approche a facilité la mise en place d'un environnement **fonctionnel**, permettant à Elasticsearch et Kibana de communiquer efficacement dans un environnement isolé et contrôlé.

Après cela, il a fallu un peu de temps afin de pouvoir afficher le premier graphique formé à partir des logs qui avaient été produits par l'exécution de mon code. L'idée étant que par la suite, l'équipe puisse se servir de mon application et des logs qu'elle fournit, associé à Elasticsearch et Kibana afin de pouvoir dresser un **tableau de bord** qui leur permettra d'intervenir **efficacement** sur les différents problèmes rencontrés à propos de IASO.

Ci dessous, un exemple de graphe possible formé à partir des logs obtenus .



D. Mes attentes ont-elles été satisfaites ?

Cette mission a été instructive et très enrichissante. Elle s'est déroulée tout au long de ces quatre mois de stage, et est encore d'actualité, car il reste encore quelques modifications a apporter, quelques détails à améliorer.

Ayant reçu l'aide et le soutien du reste de l'équipe, j'ai pu développer de nouvelles connaissances sur une technologie que je n'avais jamais utilisé auparavant et j'ai également pu approfondir celles que j'avais déjà grâce à l'observation et la compréhension de la méthodologie de travail des membres de l'équipe, ce qui m'a permis d'en apprendre encore plus sur les différents outils et leurs utilisations.

La particularité de cette mission, ce qui lui permet de se distinguer des autres projets que j'ai pu réaliser jusqu'à présent, c'est le fait de participer à toutes les étapes de conception du projet, de commencer de zéro pour atteindre l'objectif final souhaité et attendu.

J'ai pu découvrir le quotidien d'un développeur ainsi que de nombreux aspects du métier grâce à ce processus. Au-delà de la simple programmation, j'ai approfondis mes connaissances sur l'aspect méthodologique qui arrive avant même de commencer à coder. L'apport théorique des cours ont également faciliter le processus notamment lors de la découverte de GoLang, langage qui ne nous a pas été enseigné à l'IUT et qui m'était inconnu jusque là. Il va évidemment de soi que j'ai eu des blocages et que j'ai rencontré des difficultés, notamment dans la compréhension du vocabulaire technique spécifique à l'entreprise Orange, l'offre BaaS ou encore sur le développement de certaines fonctionnalités demandées. Mais je retiens avant tout que la mise en pratique de toutes mes connaissances théoriques m'a montré que j'aime ce que je fais, que j'en suis fier et que je souhaite poursuivre dans cette voie.

Parmi les problèmes les plus intéressants que j'ai pu rencontrer, je note avant tout l'installation et la configuration de la suite ELK qui m'a posé pas mal de problème et aussi le fait de devoir me familiariser avec cette technologie que je ne connaissais pas et qui a été très difficile à mettre en place.

Cela m'a cependant permis d'apprendre à trouver des solutions alternative à un problème lorsque certaines méthodes se révèlent inefficace. Cependant, il est important de garder la tête sur les épaules malgré les progrès effectués, car cela ne représente qu'un petit bout de la partie émergée de l'iceberg.

IV. CONCLUSION

À ce jour là, j‘ai pu accomplir le principal objectif de ma mission et j‘estime que le bilan de mon travail est **satisfaisant**. Cette expérience professionnelle m‘aura permis de développer de nouvelles compétences en termes de gestion de projet, notamment grâce à l‘utilisation de la méthode **Scrum**, combinée à des outils tels que **Gitlab** et **Confluence** afin de mieux suivre et contrôler les différentes tâches de ma mission.

De plus, j‘ai pu développer une meilleure connaissance d’une nouvelle technologie via l‘utilisation de GoLang, qui est un langage de programmation open source développé par Google, conçu pour être simple, efficace et performant. C’est également un langage compilé, ce qui lui permet d'offrir des performances supérieures et un temps d'exécution rapide.

Cette mission a également fait l‘objet de diverses améliorations et de différents détails qui devaient être gérés, ce qui m‘a permis de mieux comprendre les différentes étapes de la conception d‘un projet.

Dans l’ensemble, ce stage me laisse **une opinion très positive**, il aura été très enrichissant, que ce soit sur le plan technique, professionnel ou encore personnel. Bien que je me suis mis pas mal de pression au début, l’ensemble de l’équipe aura réussit à me mettre à l’aise et m’aura permis de trouver des repères assez rapidement.

De plus, cette mission requiert diverses compétences, ce qui m’aura permis d’apprendre à leurs côtés à utiliser de nouveaux outils et de nouvelles technologies, telles que **HeidiSQL** entre autres et ils n’hésitaient pas à m’aider lorsque j’en avais besoin.

V. GLOSSAIRE

API : une API (Application Programming Interface), est un ensemble de règles permettant à différents programmes de communiquer entre eux. Par analogie, on peut la comparer au menu d'un restaurant.

Media Agent : Serveur servant de passerelle entre les clients et les équipements de stockage

Confluence : plateforme de collaboration et de gestion de contenu permettant de créer, partager sur des documents ou des notes.

Broker : Intermédiaire ayant pour rôle de faciliter la communication entre les différents services.

Requête HTTP : Demande envoyée par le client à un serveur pour effectuer une opération spécifique. Elle contient généralement des informations telles que la méthode, l'URL, les paramètres et éventuellement des données ou des en-têtes.

Méthode HTTP : Elle définit la nature de la requête, comme la récupération, la création, la mise à jour ou la suppression de données. Les méthodes les plus courantes sont GET, POST, PUT et DELETE.

Token : Chaîne de caractères utilisée pour l'authentification dans un système informatique. Il permet de vérifier l'identité d'un utilisateur lors d'une interaction avec un service, permettant d'accéder à des ressources protégées de manière sécurisée.

HeidiSQL : Logiciel open source permettant de gérer et d'administrer facilement des bases de données.

JSON (JavaScript Object Notation): format d'échange de données souvent utilisé pour transmettre des données entre un serveur et une application web.

GitLab : Plateforme de gestion de code source et de collaboration pour le développement de logiciels. Permet également d'automatiser l'intégration et le déploiement d'une application via des outils de CI/CD.

Logs : Enregistrements chronologiques d'évènements et d'actions qui se produisent dans un systèmes, fournissant un historique pour le suivi et le débogage.

Scrum : Méthode agile de gestion de projets informatiques privilégiant la communication, l'inspection régulière des tâches pour connaître l'avancée de chaque tâche et s'adapter face aux contraintes.

ANNEXES

https://www.orange.com/fr/groupe/nous-connaitre/qui-sommes-nous

https://www.orange.com/fr/orange-dans-le-monde

https://www.orange.com/fr/groupe/nous-connaitre/qui-sommes-nous#mega-level-3-collapse

https://fr.wikipedia.org/wiki/Orange_(entreprise)

https://www.orange.com/fr/la-marque-orange-un-engagement-qui-traverse-les-epoques

https://www.livebox-mag.fr/box/orange-classee-5eme-marque-la-plus-valorisee-de-france/

https://www.clubic.com/actualite-524973-sur-le-fixe-et-sur-mobile-orange-ne-parvient-toujours-pas-a-endiguer-la-perte-de-d-abonnes-en-2024.html

https://newsroom.orange.com/resultats-financiers-2023/

https://newsroom.orange.com/strong-2024-results-2025-organic-cash-flow-target-raised/

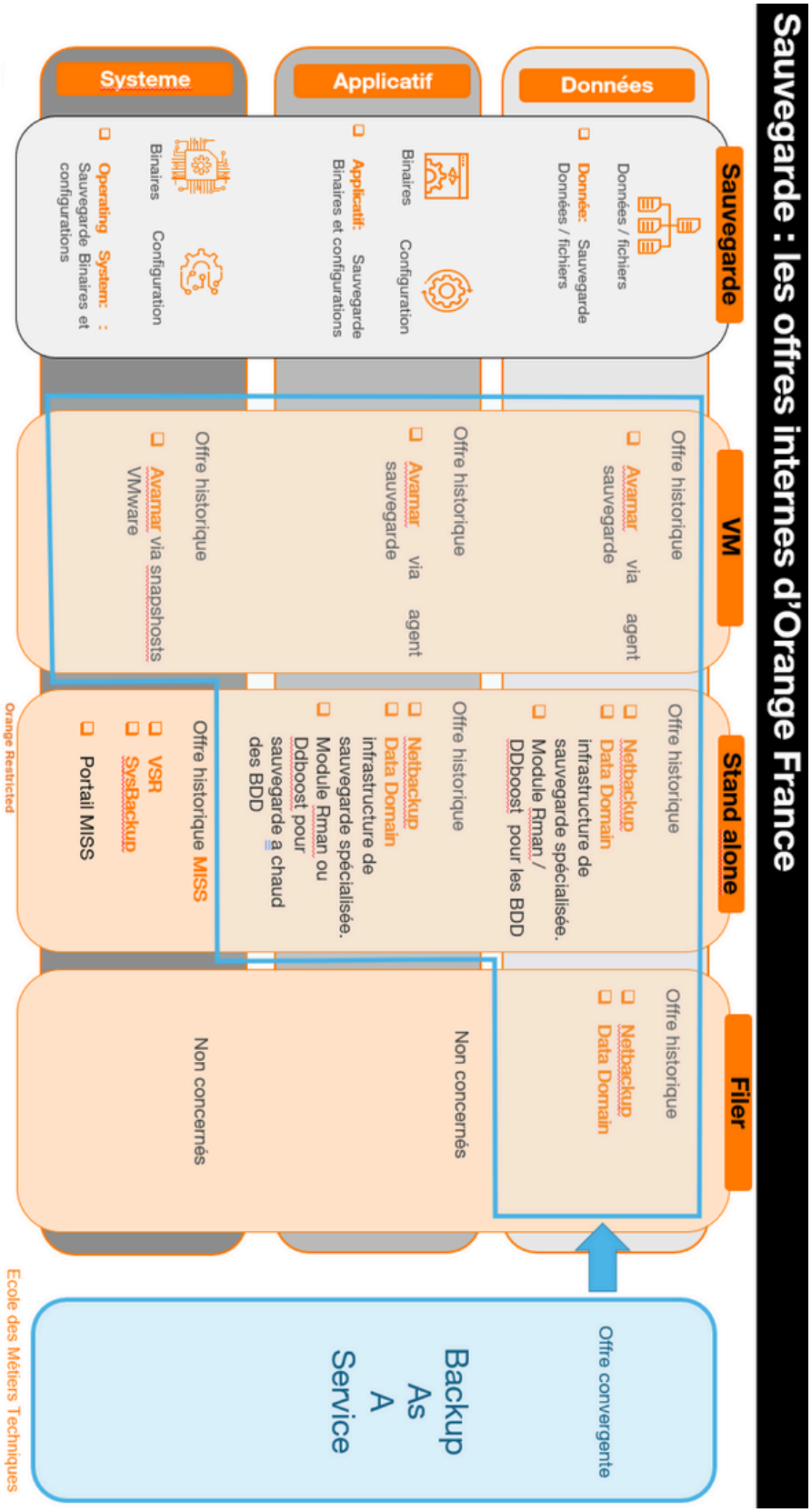
https://fr.wikipedia.org/wiki/Elasticsearch

https://fr.wikipedia.org/wiki/Kibana

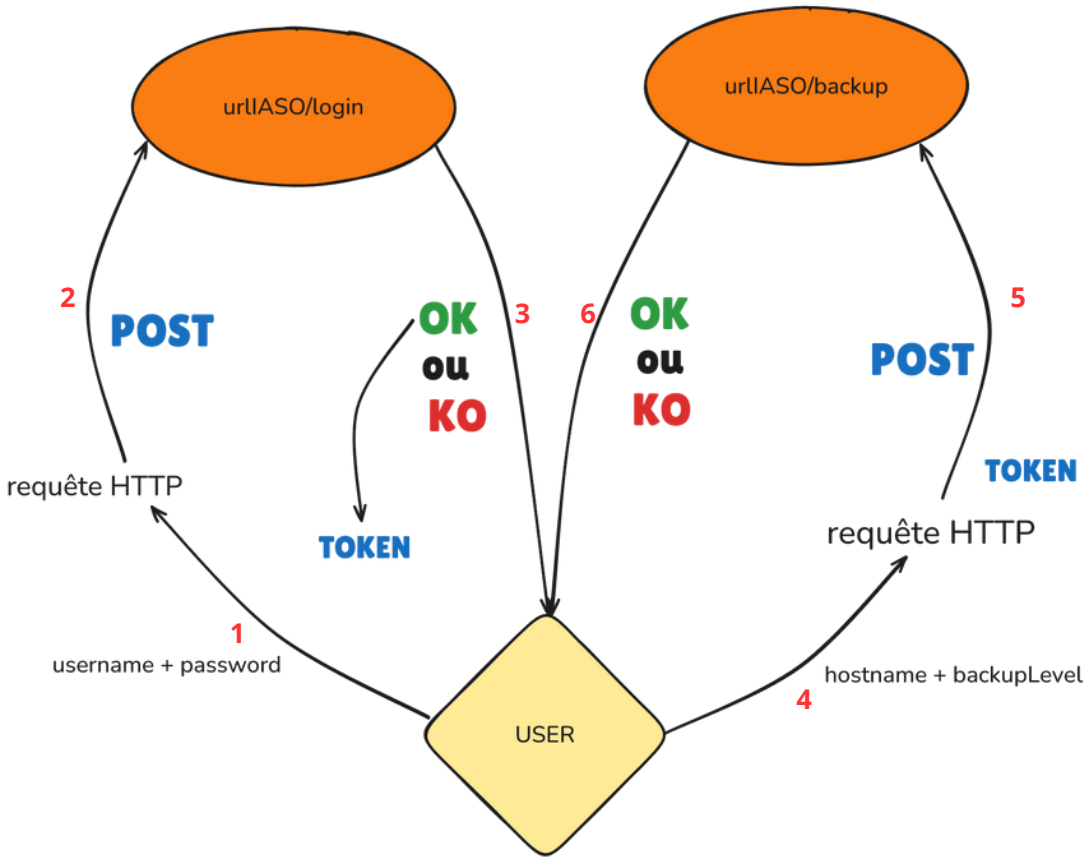
https://www.elastic.co/fr/elasticsearch

https://www.elastic.co/fr/kibana

LES DIFFÉRENTES OFFRES DE SAUVEGARDES



SCHEMA SIMPLIFIE DU LANCEMENT D'UNE SAUVEGARDE



L'utilisateur remplit un formulaire avec ses informations (username & password), qui vont ensuite être envoyées au serveur (requête POST).Si les informations sont valides, le serveur répond "OK" et une "clé d'accès" (TOKEN), sinon il répond "KO".

Lorsque l'utilisateur souhaite effectuer une sauvegarde, il donne les informations de la machine qu'il souhaite sauvegarder et ce qu'il veut sauvegarder (hostname + backupLevel) et ces informations sont ensuite envoyées au serveur (requête POST). Or, le serveur a besoin d'identifier la personne qui lui soumet cette requête et de savoir si elle en a le droit, alors la "clé" est également envoyé lors de la requête et sert d'autorisation, ce qui va permettre à l'utilisateur de lancer sa sauvegarde.

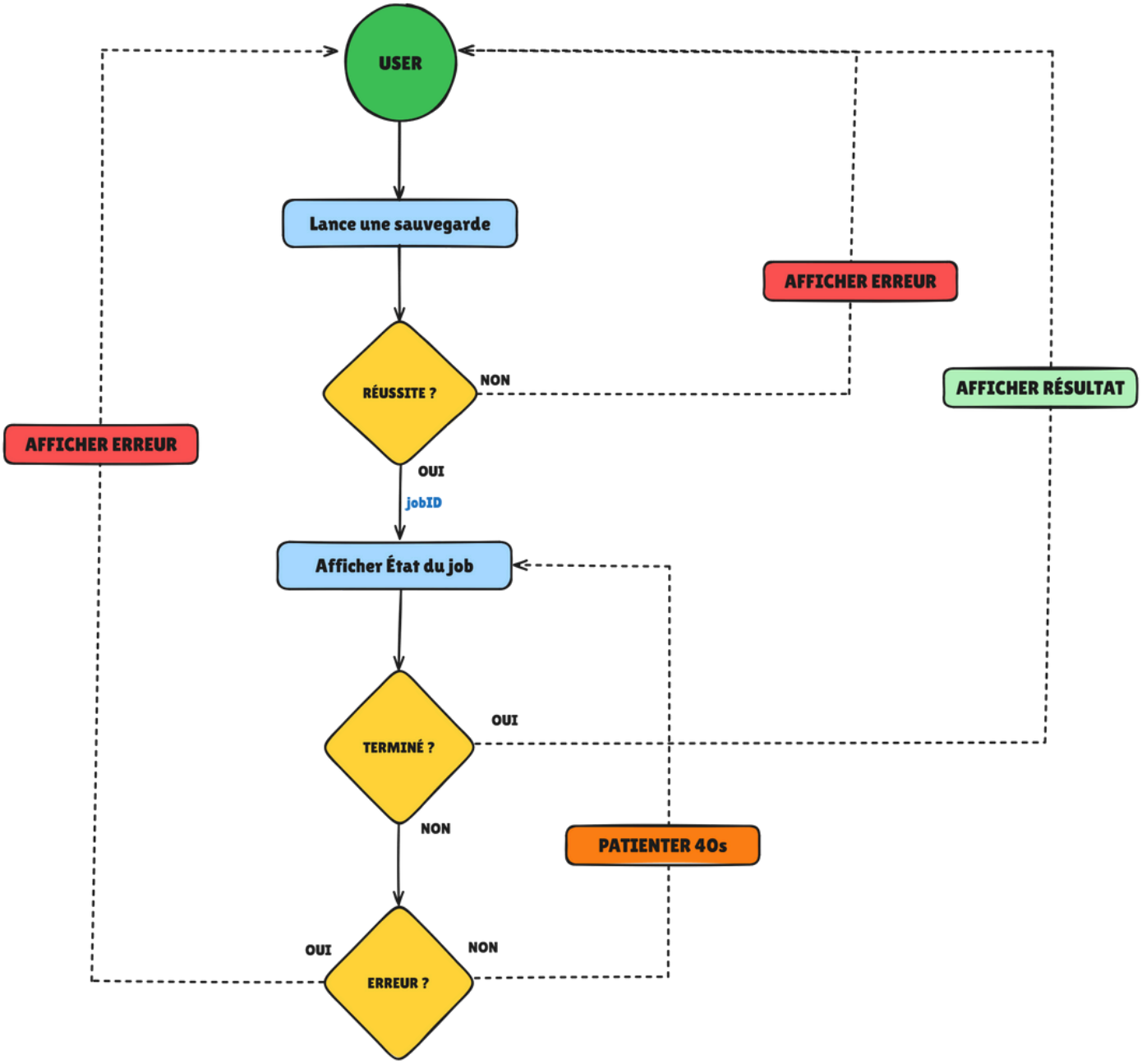



SCHÉMA SIMPLIFIÉ DU SUIVI D'UN JOB