RAPPORT DE STAGE

Stagiaire SecOps

DECARNELLE Samuel
Classe Préparatoire Informatique

Juin 2025 à Août 2025

Tuteur de stage : Monsieur Tavares Bruno
Entreprise d'accueil : KPMG SA – 2 Av. Gambetta, 92400 Courbevoie

École : IPSSI Paris – 25 rue Claude Tillier, 75012 Paris

Remerciements

Cette expérience a été avant tout une aventure humaine. En arrivant chez KPMG, j'ai découvert une équipe SecOps qui m'a accueilli avec bienveillance, simplicité et patience. Très vite, je ne me suis plus senti "stagiaire" ou "nouveau", mais pleinement membre du groupe. Nous avons partagé des cafés, des fous rires en open space, des débriefs honnêtes, et surtout des verres après le travail qui ont transformé de simples collègues en véritables amis. Je suis pleinement conscient de la chance incroyable d'avoir pu réaliser mon stage au sein d'une entreprise de cette envergure et de cette qualité ; évoluer dans un environnement à la fois exigeant, structuré et profondément humain est un privilège dont je mesure la valeur.

Je tiens donc à remercier chaleureusement M. Bruno Tavares pour son accompagnement attentif et son leadership inspirant. Un merci tout particulier à Kevin Guillois, qui m'a accompagné tout au long de mon stage : dès les premiers jours, il m'a mis à l'aise, m'a prodigué des conseils précieux et a partagé une partie de son histoire. Je suis certain que son parcours m'inspirera pour écrire le mien, avec la même exigence, la même humilité et la même envie d'apprendre.

Enfin, un grand merci à l'ensemble de l'équipe SecOps, des alternants aux experts. Chacun, à son niveau, m'a aidé et accompagné avec disponibilité, pédagogie et esprit d'entraide. Cette qualité d'accueil m'a permis d'oser, de demander de l'aide quand il le fallait, d'écouter vraiment les contraintes de chacun et de grandir au contact d'un collectif exigeant mais toujours respectueux.

Je repars surtout avec de belles relations humaines, reconnaissant d'avoir trouvé ici plus qu'une équipe : un collectif dans lequel je me sens bien, et des personnes avec qui j'aurai plaisir à continuer l'aventure, au bureau comme autour d'un verre. Merci à toutes et à tous pour votre confiance et pour la qualité des liens tissés.

Par ailleurs je remercie l'école IPSSI qui m'a permis de réaliser cette expérience par le biais de ce stage de première année.

Table des matières interactive

Introduction

Dans le cadre de mon parcours scolaire, j'ai effectué un stage de première année au sein de KPMG, au cœur de l'équipe SecOps dirigée par M. Bruno Tavares. Cette immersion m'a offert l'opportunité de découvrir le fonctionnement d'une organisation de très grande envergure et de contribuer, aux côtés d'experts, à des sujets concrets de sécurité et d'exploitation.

L'objectif de ce rapport est de présenter le contexte de ce stage, les missions réalisées, les compétences développées et les enseignements que j'en retire pour la suite de mon parcours.

Présentation de l'entreprise

KPMG dans le Monde

KPMG est un réseau de cabinets membres juridiquement indépendants, affiliés à KPMG International Limited (société de droit anglais "limited by guarantee"). Le réseau opère dans environ 145 pays.

Effectifs et activité : KPMG comptait environ 275 000 collaborateurs dans le monde à la fin de l'exercice 2024, pour un chiffre d'affaires combiné de 38,4 Md$ US. Les métiers principaux sont l'audit, le conseil (Advisory) et la fiscalité/droit (Tax & Legal, selon pays).

Positionnement : KPMG est l'un des "Big Four" et se distingue par une organisation fédérale (cabinets membres locaux) coordonnée par des standards et une marque globale.

On appelle "Big Four" les quatre plus grands réseaux mondiaux d'audit et de services professionnels : Deloitte, EY, KPMG et PwC. Ils dominent le marché de l'audit des grandes entreprises cotées et regroupent ensemble plus d'1,5 million de professionnels dans le monde.

KPMG s'appuie sur cinq valeurs communes au réseau mondial. Elles guident les comportements au quotidien, orientent les décisions et structurent la relation avec les clients, les équipes et l'écosystème.

Intégrité — « We do what is right » Nous agissons avec honnêteté, équité et cohérence, même sous pression. Chacun prend la responsabilité de ses actes et tient ses engagements, afin de bâtir la confiance dans la durée.

Excellence — « We never stop learning and improving » Nous visons un niveau élevé de qualité et d'exigence professionnelle. Curiosité, apprentissage continu et amélioration par les données et le feedback sont au cœur de notre manière de travailler.

Courage — « We think and act boldly » Nous faisons preuve d'audace et de franchise intellectuelle : scepticisme professionnel, questions difficiles lorsqu'un doute existe, et capacité à sortir de notre zone de confort pour faire ce qui est juste.

Ensemble — « Together : we respect each other and draw strength from our differences » La collaboration est essentielle : nous valorisons la diversité des profils et des points de vue, créons un environnement inclusif et faisons en sorte que chaque voix soit entendue pour produire le meilleur travail collectif.

For Better — « We do what matters » Nous privilégions l'impact utile et durable : rôle de confiance sur les marchés, contribution positive aux communautés et à la société, et décisions de long terme au service d'un avenir meilleur.

KPMG en France

Périmètre : KPMG France est le cabinet membre pour le marché français au sein du réseau KPMG International. Le siège est à Paris La Défense (Tour EQHO).

Métiers et secteurs : audit, conseil, anciennement expertise comptable (maintenant RIDGE) et, selon les structures, fiscalité et droit (via des entités dédiées). Le cabinet intervient auprès de grands groupes, ETI/PME et secteur public.

Taille et empreinte : KPMG France communique un effectif d'environ 8 300 professionnels et un chiffre d'affaires d'environ 1,5 Md€ et 45 bureaux (exercices récents).

Évolutions récentes : en 2025, la création de Rydge Conseil a été annoncée pour adresser spécifiquement le segment TPE/PME, tout en restant adossé au groupe.

Implantations rattachées à KPMG France (exemples) KPMG France couvre la métropole et plusieurs territoires ultramarins via ses bureaux locaux, intégrés au périmètre "KPMG en France" :

- Polynésie française : bureau KPMG Papeete (audit, conseil, expertise, droit/fiscalité).
- Nouvelle-Calédonie : bureau KPMG Nouméa.
- Antilles-Guyane : bureaux dont Fort de France (Martinique).

Remarque : certaines juridictions proches géographiquement peuvent constituer des cabinets membres distincts (ex. KPMG GLD & Associés à Monaco est un cabinet membre du réseau KPMG International, juridiquement séparé de KPMG France).

Présentation du service

Présentation d'ITS

ITS signifie **Information Technology Services** au sein de KPMG.

Rôle et mission ITS (Information Technology Services) regroupe les fonctions qui conçoivent, opèrent et sécurisent les systèmes d'information de KPMG. Sa mission est de fournir aux équipes métiers un environnement numérique fiable, sécurisé et performant, en alignant la technologie sur les priorités business du cabinet (audit, conseil, tax/legal) et sur les exigences de conformité du réseau KPMG International.

Périmètre fonctionnel –

- Postes de travail et collaboration (EUC End User Computing): gestion du poste Windows/Mac, suite collaborative, messagerie, visioconférence, MDM/MAM pour les mobiles.
- Infrastructures et Cloud : datacenters, serveurs, virtualisation, stockage, sauvegardes, réseaux et interconnexions hybrides ; adoption et gouvernance Cloud (IaaS/PaaS/SaaS).
- Applications et Data : portefeuille applicatif interne (ERP/Finance, RH, outils d'audit), intégrations, qualité et disponibilité des données.
- Cybersecurity/SecOps : gestion des identités et des accès (IAM(Identity Access Management)/PAM(Privilege Access Management)), protection des terminaux, surveillance et réponse aux incidents (SOC(Centre d'Opération de Sécurité)), conformité (normes réseau KPMG, réglementations locales).
- Support et opérations : service desk, gestion des incidents/problèmes/changements, suivi des SLA, amélioration continue.

Organisation (schéma-type) ITS est structuré en pôles complémentaires, pilotés par une gouvernance IT et des comités de changement :

- EUC/Workplace
- Réseaux & Infrastructures
- Cloud & Plateformes
- Applications (ERP/Finance, RH, outils d'audit)
- Cybersecurity / SecOps / GRC (Gouvernance Risque et Conformité)
- Service Management (ITIL(Information Technology Infrastructure Library)/ServiceNow)

Chaque pôle dispose d'équipes run (exploitation) et change (projets/transformations). Les activités transverses (architecture, sécurité, data) assurent la cohérence technique et la conformité.

Gouvernance et processus ITS applique les bonnes pratiques ITIL pour standardiser la qualité de service :

- Gestion des incidents et demandes via un portail ITSM (ex. ServiceNow) avec SLA définis.
- Gestion des changements et des mises en production (CAB/CCT) afin de sécuriser l'exploitation.
- Gestion des vulnérabilités, patching et EDR ; campagnes de remédiation pilotées par indicateurs.
- Continuité d'activité : sauvegardes, PRA/PCA et tests réguliers.
- Conformité : respect des politiques globales KPMG (sécurité, données, records management) et des obligations locales (ex. RGPD).

Technologies et outils –

- Identité et accès : Azure AD/Entra ID, MFA, Conditional Access ; PAM (ex. CyberArk).
- Poste et sécurité : Microsoft 365, Intune/Endpoint Manager, EDR/AV (ex. Defender).
- Observabilité : SIEM/SOAR (ex. Wazuh), supervision, logs centralisés.
- Collaboration : Teams/SharePoint/OneDrive ; salles de réunion intelligentes.
- ITSM : ServiceNow (catalogue de services, CMDB, workflows).

Interactions avec les métiers –

- Les lignes de service (Audit, Advisory, Tax/Legal) pour comprendre les besoins, garantir la disponibilité des outils d'audit et sécuriser les environnements sensibles.
- Les fonctions transverses et support (RH, Finance, Qualité/Risque) pour automatiser les processus, protéger les données et produire les rapports de conformité.

Ancrage global et local –

En tant que cabinet membre, KPMG France aligne ITS sur les standards du réseau global (sécurité, architecture, gouvernance), tout en opérant des solutions et des services adaptés au contexte réglementaire français et aux besoins terrain (multisites, DOM TOM, clients soumis à des exigences sectorielles).

Présentation de SecOps

L'équipe SecOps (**Security Operations**) est chargée d'opérer la sécurité au quotidien, en protégeant les utilisateurs, les postes, les applications et les infrastructures de KPMG. Son objectif : prévenir, détecter et répondre rapidement aux menaces tout en garantissant la continuité d'activité et la conformité aux politiques KPMG International et aux exigences locales (ex. Gestion de Vulnérabilitées, RGPD). SecOps se situe au cœur d'ITS entre la gouvernance sécurité (CISO/GRC), le SOC, les équipes Infrastructures/Cloud et les équipes Workplace/Applications.

Périmètre d'intervention –

- Détection et réponse aux incidents : triage, investigation, remédiation et leçons apprises.
- Gestion des vulnérabilités : scans, priorisation basée sur le risque, coordination du patching (OS, middleware, applicatif).
- Durcissement et protection des endpoints/serveurs : EDR/AV, chiffrement, contrôle des périphériques, contrôle des applications, Credential Guard, protection des secrets.
- Gestion des identités et accès (en lien avec IAM/PAM) : MFA, règles de Conditional Access, recertifications, moindres privilèges.
- Sécurité Cloud et plateformes : posture management, durcissement des images, secrets management, clés et certificats, revue des déploiements.
- Conformité opérationnelle : déploiement des politiques globales KPMG (standards de configuration, classification de données), preuves de contrôle et reporting auditables.
- Sensibilisation ciblée et accompagnement : campagnes anti-phishing, "security champions" côté métiers, guidage pendant les mises en production à risque.
-

Responsabilités clés (Rôle SecOps) –

- Prévenir : appliquer les baselines de sécurité, automatiser les contrôles (CIS/Benchmarks internes), orchestrer le patching régulier et d'urgence.
- Détecter : maintenir les règles, connecteurs et cas d'usage du SIEM/SOAR ; améliorer en continu la couverture de détection.
- Répondre : exécuter les playbooks d'investigation/remédiation, contenir l'incident (isolation poste, reset de secrets), coordonner avec le SOC et ITS.
- Améliorer : piloter les "post incident reviews", mettre à jour les politiques/standards, ajuster les scénarios de détection et de prévention.
- Prouver : produire des tableaux de bord, indicateurs et dossiers de preuves (audits internes/externes, contrôles clients).

Flux opérationnels type –

1. Détection par le SOC ou alerte EDR → qualification par SecOps → ouverture incident P1/P2.
2. Investigation (journaux, télémétrie, timeline, corrélation) → hypothèse de cause racine.
3. Contention/remédiation (isolation, correctifs, révocation tokens, blocage IOC, restauration) + communication utilisateur/métiers.

4. Clôture contrôlée → RCA, actions préventives (MFA renforcé, règle CA, règle EDR, règle SIEM).
5. Retour d'expérience au CAB/Comité Sécu + mise à jour des playbooks.

Gouvernance et processus –

- ITSM/ITIL : incidents, problèmes, changements (CAB/CCT), demandes de service ; CMDB tenue à jour.
- Cybersécurité : politiques globales KPMG, normes de configuration, revue périodique des accès (SoD), gestion des exceptions (délai, justification, compensations).
- Continuité : PRA/PCA, sauvegardes chiffrées, tests de restauration, exercices de crise (table tops).
- Relations externes : interface avec éditeurs, CERT sectoriels, équipes globales KPMG pour les campagnes mondiales.

Interfaces clés –

- SOC : surveillance 24/7, escalade L2/L3 vers SecOps, maintien du catalogue de détection.
- IAM/PAM : politiques MFA, JIT/JEA, comptes à privilèges, recertifications périodiques.
- Workplace/EUC : déploiements Intune/GPO, contrôles EDR, posture Zero Trust sur les postes.
- Infrastructures & Réseaux : durcissement serveurs/réseaux, segmentation, reverse proxy/WAF, VPN/ZTNA.
- Cloud/Apps : durcissement IaC, secrets management, revue sécurité avant mise en prod, tests de sécurité applicative.
- GRC/Qualité-Risque : cartographie des risques, contrôles clés, audits, conformité réglementaire.
- Métiers : accompagnement des équipes Audit/Advisory/Tax pour exigences clients et cycles projets.

Jeux d'outils (quelques exemples) –

- SIEM/SOAR : [Microsoft Sentinel / Splunk / QRadar] + playbooks automatisés.
- Endpoint/Serveur : [Microsoft Defender / CrowdStrike / Trellix], chiffrement [BitLocker/FileVault], contrôle applicatif.
- Gestion des vulnérabilités : [Qualys], patching [WSUS/Intune/SCCM, Ansible/Yum/Apt].
- Identité et accès : Entra ID (Azure AD), MFA, Conditional Access, PAM [CyberArk], SSPR, Identity Protection.
- Cloud Security: [Defender for Cloud / Prisma / Wiz], secret stores [Key Vault], CSPM/CIEM.
- ITSM : [ServiceNow], CMDB, catalogue de services, intégrations.

Valeur pour KPMG et les métiers –

- Réduction du risque opérationnel et financier : moins d'interruptions, moins d'incidents à impact client.

- Confiance marché/réglementaire : capacité à démontrer la maîtrise des contrôles sécurité lors d'audits et de diligences client.
- Accélération des projets : cadrage sécurité anticipé, modèles sécurisés réutilisables (images durcies, configurations baselines).
- Optimisation des coûts : automatisation (SOAR), rationalisation des outils, baisse du volume d'incidents récurrents.

Modèle RACI synthétique –

- Détection (SIEM/SOC) : R=SOC, A=SecOps, C=Infra/Apps, I=GRC.
- Réponse/Remédiation: R=SecOps, A=CISO/Head of SecOps, C=Infra/EUC/IAM, I=Métiers.
- Patching vulnérabilités: R=Infra/EUC, A=Head of Infra, C=SecOps, I=GRC/Métiers.
- Baselines sécurité : R=SecOps, A=CISO, C=Infra/Cloud/Apps, I=Service Management.

Feuille de route (exemples) –

- Zéro confiance renforcée : segmentation, vérifications d'identité/terminal contextuelles.
- "Shift left" sécurité : contrôles dans les pipelines CI/CD, IaC durcie.
- Expansion SOAR : automatisation de x% des tâches répétitives d'ici [date].
- Rationalisation outils : convergence EDR/AV, couverture unique multisystèmes.

Cadre documentaire –

- Politiques et standards sécurité [références internes].
- Playbooks IR, guides de durcissement, procédures d'exploitation.
- Dossiers de preuves et registres de contrôles (audits internes/externes).

L'équipe SecOps –

  - Responsable SecOps: [Tavares Bruno]
  - Akka Farah
  - Baroudi Anas [Alternant]
  - Guillois Kévin
  - M'Chichi Amine
  - Martin Christophe
  - Oufker Issam [Alternant]
  - Thi Thi Mehdi
  - Ade Electra [Alternante]
  - Decarnelle Samuel [Stagiaire]
  - Kasmi Zaineb
  - Corre Valentin [Externe]
  - Djouab Tarik [Externe]
  - El Haytam [Externe]
  - Moussaoui [Externe]
  - Dahmani Samy [Externe]

Mon rôle au sein de l'équipe

Observation

Mon premier mois (du [01/06/25] au [01/08/25]) a été dédié à l'observation active et à la compréhension du fonctionnement d'ITS et des interactions de SecOps avec les autres pôles. J'ai principalement assisté à des réunions thématiques, sans intervenir sur le fond, afin de cartographier les processus, les outils et les responsabilités :

- Onboarding SecOps et présentation des politiques sécurité du réseau KPMG.
- Revue hebdomadaire des incidents avec le SOC (use cases, priorisation, MTTR).
- CAB/Comité des changements : planification des déploiements et fenêtres de maintenance.
- Comité "Vulnérabilités & Patching" : état des scans, priorisation par criticité, risques résiduels.
- Point EDR/Endpoint : tuning des politiques, couverture du parc, campagnes d'isolation/test.
- Sécurité identité : MFA/Conditional Access et gestion des exceptions encadrées.
- Cloud SecOps : posture management (CSPM), durcissement des images et secrets management.
- Post incident review (RCA) sur un incident P2 pour capitaliser les enseignements.
- Préparation audit/contrôles : dossiers de preuves, conformité aux standards internes.
- Continuité d'activité : préparation d'un test PRA (Plan de Reprise d'Activité) / PCA et validation des plans de restauration.

Ce mois d'immersion m'a permis d'acquérir une vision claire de la chaîne de valeur ITS (Workplace, Infrastructures, Cloud, Applications) et de la place de SecOps entre la détection (SOC), la remédiation (Infra/EUC/Cloud) et la conformité (GRC). J'ai également compris le cadre ITIL (incidents, problèmes, changements) et le rôle de la CMDB et des SLA dans le pilotage opérationnel.

Réalisation

À l'issue de cette phase, j'ai progressivement pris part aux activités de l'équipe en mode "projet", avec un positionnement d'appui à la préparation, à l'exécution et au suivi des actions sécurité. Mon rôle s'est articulé autour de trois axes :

- Préparation : collecte d'informations (inventaires, états de conformité), rédaction ou mise à jour de documents de référence (checklists, mini playbooks, guides d'usage).
- Exécution : participation à des déploiements encadrés (par ex. ajustement de politiques EDR ou règles Conditional Access sur des périmètres pilotes), suivi des changements via l'outil ITSM et contrôle post déploiement.

- Suivi et reporting : consolidation d'indicateurs (couverture EDR, conformité patching, statut des remédiations), préparation de supports pour les comités (SecOps Weekly, CAB).

Projets accompagnés –

- Durcissement des postes et serveurs avec mise à jour de politiques EDR/AV.
- Amélioration du processus de gestion des vulnérabilités (cycle scan → priorisation → patch → vérification).
- Renforcement de l'authentification (MFA/Conditional Access) sur des populations ciblées.
- Contribution à la préparation des exercices PRA/PCA et à la collecte de preuves pour audits. Les détails, résultats chiffrés et retours d'expérience de ces projets seront développés dans la partie "Mes missions".

Mon périmètre au quotidien –

- Coordination avec les équipes EUC/Infra/Cloud pour planifier et tracer les remédiations.
- Ouverture/suivi de tickets dans l'ITSM, alimentation de la CMDB et respect des workflows de changement.
- Vérifications post déploiement (contrôles de conformité) et remontée d'écarts.
- Mise à jour de la documentation opérationnelle (procédures courtes, modes opératoires).

Compétences développées –

- Méthodes : application pratique d'ITIL, gestion multi interlocuteurs, priorisation par le risque.
- Techniques : bases d'EDR, principes de MFA/Conditional Access, lecture de rapports de vulnérabilités.
- Communication : synthèse d'indicateurs, préparation de supports de comité, vulgarisation des décisions sécurité.

Valeur apportée –

En libérant du temps aux ingénieurs SecOps pour les tâches d'analyse avancée et en fiabilisant la préparation/traçabilité des déploiements, j'ai contribué à accélérer la mise en œuvre des contrôles sécurité tout en respectant le cadre de gouvernance (SLA, CAB, conformité).
Nous avions également des points hebdomadaires, afin de faire les validations des priorités.
Ces points m'ont permis de comprendre l'organisation d'une équipe de sécurité au sein d'une grande entreprise et à prioriser les risques élevés.

Présentation de mes missions

Mise en place d'un outil de détection des cycles de vie des systèmes d'exploitation

1. Enjeux et contexte opérationnel

Problématique métier : dans un environnement sensible comme KPMG/ITS, la présence de machines en fin de support ou en fin de vie représente à la fois un risque de cybersécurité (absence de correctifs, exposition aux vulnérabilités exploitées) et un enjeu de conformité (audits clients, référentiels internes, exigences contractuelles).

Situation de départ : nous disposons d'une visibilité riche via Microsoft Defender for Endpoint (MDE), déjà largement déployé sur le parc France. MDE "voit" les machines actives, leurs OS, leurs versions, et expose des horodatages (FirstSeen/LastSeen) utiles pour apprécier la fraîcheur des données.

Besoin exprimé –
Obtenir un dispositif fiable, reproductible et facilement diffusable pour :

- Identifier les machines EOS/EOL,
- Prioriser les actions de remédiation,
- Fournir des rapports clairs aux équipes EUC/Infra et aux comités (SecOps Weekly, CAB, audits internes/externes).

2. Choix d'architecture et principe de la solution

Source de vérité parc : un export MDE (Excel) des machines connectées au réseau KPMG France. Ce choix assure que nous analysons des actifs réellement observés et non "théoriques".

Référentiel cycle de vie : l'API publique endoflife.date, qui maintient un inventaire à jour des produits (OS, distributions, etc.) avec leurs jalons de support (dates de fin de support, de fin de vie, prolongations éventuelles).

Orchestrateur :
Un script PowerShell que j'ai développé "from scratch", avec comme objectifs :

- Simplicité de mise en œuvre (un fichier .ps1 et un Excel en entrée),
- Robustesse (gestion des erreurs, rate limiting, cache en mémoire),
- Lisibilité des résultats (rapports Excel multi onglets prêts à être partagés).

3. Chaîne de traitement détaillée (pipeline)

- Étape 1 – Ingestion
  Le script lit le fichier Excel exporté depuis MDE (module ImportExcel), en ciblant les colonnes pertinentes : DeviceName/ID, OSPlatform, OSVersion/Build (ou équivalent), DeviceGroup/Tags, LastSeen, etc.

- Étape 2 – Normalisation
  Les libellés MDE peuvent être hétérogènes. J'ai ajouté des fonctions de normalisation :
    - Standardisation des formats (ex. "22h2" → "22H2"),
    - Traitement spécifique Windows Server (extraction de l'année 2016/2019/2022 depuis le libellé),
    - Mapping "noms MDE → produits API" (ex. "Windows 10" → api product "windows", "Windows Server 2019" → "windows-server", "Ubuntu" → "ubuntu").

- Étape 3 – Optimisation des appels API
  Plutôt que d'interroger endoflife.date pour chaque machine, je regroupe par couple OS+Version.
  Résultat : un parc de 5 000 machines mais seulement quelques dizaines d'appels API si l'hétérogénéité est raisonnable. Un cache en mémoire évite les redondances.

- Étape 4 – Interrogation de l'API endoflife.date
  Via Invoke-RestMethod, le script récupère le cycle de vie du produit/versions ciblées (dates EOL/EOS, états intermédiaires). Il gère :
    - Le rate limiting (retries et pauses adaptatives),
    - Les échecs transitoires réseau,
    - Les versions imprécises (ex. OS "reconnus" mais version difficile à classifier).

- Étape 5 – Classification
  Chaque machine reçoit un statut calculé :
    - Supported (entièrement supporté),
    - Alerte (EOL/EOS dans moins de X jours, paramètre -WarningDays, 180 jours par défaut),
    - End of Support (EOS atteint),
    - End of Life (EOL atteint),
    - Non analysable (données OS insuffisantes/ambigües). Le script tient compte de LastSeen pour signaler les données "stales" (ex. >30 jours) afin d'éviter les sur réactions sur des équipements inactifs.

- Étape 6 – Restitution
  Génération d'un classeur Excel multi feuilles :
    - Résumé global (répartition par statut, compteurs, éventuellement un mini pivot),
    - Détails par OS/Version (pour comprendre où se concentrent les risques),
    - EOL_Machines (liste actionnable à transmettre à EUC/Infra),
    - EOS_Machines (priorité forte),

- Résumé par statut avec “Top OS à risque”. Un export CSV est aussi produit pour intégration éventuelle dans d’autres outils.

Schéma de flux (vue synthétique) –

MDE (export Excel) → ImportExcel → Normalisation OS/Version → Grouping OS+Version → Appels API endoflife.date (+ cache/rate limiting) → Classification (Supported/Alert/EOS/EOL) → Rapports Excel + CSV

4. Résultats concrets et valeur pour SecOps/ITS

- Visibilité unifiée : une photographie fidèle des OS “vus par MDE”, enrichie d’un statut lifecycle “à jour” grâce à l’API. Finis les croisements manuels multiples.
- Priorisation opérationnelle : des onglets “EOL_Machines” et “EOS_Machines” prêts à l’emploi pour déclencher des plans d’assainissement (upgrade, reimage, décommission).
- Gain de temps : le regroupement OS+Version et le cache réduisent les appels API, accélèrent l’exécution et rendent l’outil viable même pour de grands volumes.
- Traçabilité et partage : un livrable Excel normé, facilement partageable en comité et attachable à des tickets ITSM ou des demandes de changement.
- Fondations pour l’industrialisation : paramètres (WarningDays, BatchMode, ShowOnlyProblems) et logs d’exécution posent les bases d’une automatisation ultérieure.

5. Difficultés rencontrées et apprentissages clés

Apprendre PowerShell “sur le tas” car je suis parti de zéro et j’ai dû assimiler rapidement :

- Le paradigme objet (pipeline, Select-Object, Where-Object),
- La manipulation de tableaux/hashtables et la création de PSCustomObject,
- La lecture/écriture Excel via ImportExcel.

Normalisation des versions : le mapping des versions Windows (10/11, Server 2016/2019/2022) n’est pas trivial. J’ai développé une logique de normalisation et des règles spécifiques (extraction d’année pour Server, capitalisation des labels).

Robustesse des appels REST : j’ai implémenté des retries avec backoff et un mode BatchMode plus conservateur pour protéger l’API endoflife.date et éviter les timeouts.

Performance et volumétrie : le passage d’un traitement “machine par machine” à une approche “groupée + cache” a été déterminant pour la vitesse et la stabilité.

Qualité de données : j’ai appris à gérer les cas “Non analysable” (versions incomplètes, machines peu fraîchement vues) en les signalant clairement pour investigation ultérieure.

6. Gouvernance, sécurité et limites assumées

Dépendances externes maîtrisées : l'API endoflife.date est publique et pratique.
Pour un usage récurrent en entreprise :

- Prévoir une allowlist proxy et une note d'architecture,
- Documenter le comportement en cas d'indisponibilité (cache de secours, reprise).

Fiabilité des données source : MDE reflète les machines observées. Une machine non vue récemment peut biaiser la photographie. Je l'ai traité avec des indicateurs de fraîcheur (LastSeen) et une catégorie "à confirmer".

Périmètres exclus/particuliers : certaines machines (lab, exceptions réglementées, ESU) peuvent suivre des règles différentes. Le rapport prévoit des statuts "exception/à qualifier" pour ne pas mélanger ces cas avec le reste du parc.

7. Axes d'amélioration et feuille de route

Automatisation bout en bout :

- Remplacer l'export manuel par un appel planifié à l'API MDE (Advanced Hunting) via un runbook (Azure Automation) ou une tâche planifiée, afin de disposer de rapports à date fixe (hebdo/mensuel).
- Publier automatiquement les résultats sur un partage contrôlé et notifier via Teams/email.

Intégration outillée :

- Création automatique de tickets ITSM par périmètre (EUC/Infra), avec les onglets EOL/EOS en pièces jointes.
- Publication d'un dataset Power BI pour suivi des tendances et pilotage par objectifs.

Résilience :

- Ajouter un cache persistant (JSON horodaté) pour pallier l'indisponibilité ponctuelle d'endoflife.date et tracer la version du référentiel utilisée.
- Enrichir la logique "fraîcheur MDE" (paliers J+7/J+30/J+60) pour séparer les actifs réellement actifs des traces historiques.

Qualité logicielle :

- Externaliser les mappings OS dans un fichier de configuration versionné (facile à maintenir)

8. Mode d'emploi (rappel)

```
    # Exécution standard (alerte à 180 jours)
    powershell -ExecutionPolicy Bypass -File .\EOS-Checker.ps1 -ExcelPath
".\MDE_AllDevices_20250625.xlsx" -WarningDays 180

    # Gros volumes avec prudence (rate limiting renforcé) + cache en mémoire
    powershell -ExecutionPolicy Bypass -File .\EOS-Checker.ps1 -ExcelPath
".\MDE_AllDevices.xlsx" -BatchMode -UseCache
```

9. Conclusion personnelle

Cette mission m'a permis de livrer une capacité concrète de détection EOS/EOL, immédiatement exploitable par SecOps et les équipes EUC/Infra. Au-delà du résultat visible (rapports prêts à l'emploi), j'ai beaucoup appris : PowerShell "en conditions réelles", intégration d'une API publique, gestion de la performance et de la robustesse, et surtout mise en forme de livrables compréhensibles pour accélérer la décision.
La solution actuelle répond au besoin ; la feuille de route proposée permettra de l'industrialiser et de la rendre encore plus résiliente, tout en améliorant l'expérience des équipes consommatrices.

Localisation des machines sur lequel Credential Guard était dysfonctionnel

Credential Guard Credential Guard est une fonctionnalité de sécurité de Windows (basée sur la virtualisation, VBS) qui isole les secrets d'authentification du processus LSASS (Local Security Authority Subsystem Service). Elle empêche la récupération de mots de passe, de hachages NTLM (NTLM est un package de sécurité qui fournit l'authentification, l'intégrité et la confidentialité aux applications) ou de tickets Kerberos (Protocole d'authentification de réseau) et réduit les attaques de type pass-the-hash / pass-the-ticket. Quand il n'est pas opérationnel, le poste est plus exposé aux vols d'identifiants.

Problème initial :

Nos outils de sécurité remontaient des machines "Credential Guard non opérationnel" dans un parc international. Or KPMG France n'est responsable que d'une partie de ces postes.
Les données disponibles n'offraient pas directement la réponse "cette machine relève de la France" : il manquait un lien fiable machine → utilisateur → localisation. Agir sans ce cadrage risquait des envois de matériel au mauvais périmètre, une mauvaise priorisation et des coûts inutiles.

Problématique :

Identifier de façon fiable et justifiée quelles machines en anomalie "Credential Guard" appartiennent au périmètre KPMG France, en rattachant chaque machine à son utilisateur puis à sa localisation, afin de déclencher (uniquement pour ces postes) les actions de remédiation/remplacement.

Mon rôle :

Le premier mois, j'ai surtout observé. J'ai participé aux réunions, suivi les échanges entre les équipes sécurité, EUC et support, et j'ai cherché à comprendre pourquoi une alerte "Credential Guard non opérationnel" ne se traduisait pas automatiquement par un remplacement de poste côté France. Très vite, nous avons choisi de ne pas agir au plus vite, mais au plus juste : notre responsabilité était de traiter les machines réellement rattachées à KPMG France, pas celles des autres pays.

À partir de là, j'ai pris le relai technique. On m'a fourni deux exports : d'un côté la liste des machines signalées par Credential Guard, de l'autre un inventaire machines-utilisateurs.
J'ai décidé de rattacher chaque machine problématique à une personne concrète. C'est moi qui ai conçu et écrit le script PowerShell pour y parvenir. Mon objectif était simple: partir d'un fichier Excel "UserName, Hostname", interroger l'Active Directory, récupérer les informations pertinentes, puis produire un export propre que l'équipe puisse exploiter sans effort.

Je me suis appuyé sur une contrainte réelle du SI : l'AD ne maintient pas un lien officiel "machine → propriétaire". J'ai donc choisi de remonter par l'utilisateur. Mon script lit l'Excel source, vérifie sa structure, installe au besoin les modules nécessaires (ImportExcel et ActiveDirectory) sans droits administrateur, puis interroge l'AD à partir du UserName. J'ai

commencé par une donnée volontairement sobre : la Description. Dans notre contexte, c'est un champ suffisamment discriminant pour inférer la localisation de l'utilisateur, car il reprend des mentions internes qui permettent de distinguer la France des autres entités. L'idée n'était pas de tout prendre, mais de prendre ce qui fait foi et reste lisible. J'ai ajouté un journal de traitement (Write-Log) pour suivre l'exécution et des statistiques de fin de course pour rendre le résultat immédiatement compréhensible.

Côté décision, nous avons choisi de classer les cas en trois familles : « France » quand les marqueurs convergent clairement vers KPMG France, « Hors France » lorsque les informations pointent vers une autre entité, et « À confirmer » quand les signaux sont incomplets ou ambigus. Je préfère assumer quelques « À confirmer » plutôt que d'automatiser à tort des décisions fragiles. Cette posture m'a permis d'expliquer, ligne par ligne, pourquoi telle machine relevait (ou non) de notre périmètre, et sur quelle information je m'appuyais.

Au terme du deuxième mois, le livrable que j'ai produit répondait exactement au besoin : un fichier Excel unique, lisible, avec pour chaque machine l'utilisateur rattaché, l'information AD utile et la décision de périmètre. Concrètement, cela nous a permis d'identifier les personnes dont KPMG France était responsable et de déclencher l'envoi de nouvelles machines fonctionnelles. Les autres pays, disposant de leurs propres services informatiques, pouvaient prendre le relai pour leurs collaborateurs. Ce résultat est simple, mais il tient debout : il est traçable, vérifiable, et actionnable.

Conclusion :

Ce que je retiens, c'est la valeur d'une chaîne maîtrisée de bout en bout. J'ai choisi une solution proportionnée aux enjeux : un rapprochement d'exports, une interrogation ciblée de l'AD, et un export final propre. J'ai privilégié la clarté et la preuve, afin que nos décisions soient faciles à défendre devant le comité et simples à exécuter par les équipes terrain.

Mise en place d'une solution d'auto-patching pour les serveurs Linux de KPMG France

En arrivant sur cette mission, il n'existait pas d'outil unifié de patching pour les serveurs Linux de KPMG France. Le parc était hétérogène (PRE PROD et PROD, distributions différentes), et "appliquer les dernières mises à jour" sans cadre créait un risque de dérive entre environnements, d'absence de réversibilité, et de difficulté d'audit. J'ai donc démarré from scratch : pas de code existant à adapter, ni de cadre technique prédéfini à suivre.
L'objectif n'était pas uniquement d'installer des patchs, mais d'industrialiser un processus reproductible, traçable, réversible, et utilisable à la fois par un humain (TUI) et par un ordonnanceur (CLI/OpCon).

"Appliquer les dernières mises à jour" sans cadre commun exposait à trois difficultés majeures :

- Drift entre PRE PROD et PROD : entre le moment où PRE PROD valide un patch et celui où PROD l'applique, les dépôts peuvent publier une version plus récente.
  Résultat : PRE PROD et PROD ne tournent plus exactement sur les mêmes versions.
- Réversibilité incertaine : en cas de régression (performance, dépendances, compatibilité applicative), revenir proprement en arrière est difficile et lent si l'on ne sait pas précisément quelles versions étaient installées.
- Traçabilité perfectible : sans artefacts normalisés (listes gelées, métadonnées, logs), il est compliqué d'auditer "qui a installé quoi, quand, et sur quelle version".

Je suis donc parti from scratch, sans code préexistant. Mon objectif était de transformer un besoin risqué ("patcher la PROD") en un processus industrialisé, reproductible, traçable et réversible, utilisable à la fois en mode manuel (TUI) et automatisé (CLI/ordonnanceur).

Nécessité de trouver une solution et explication des choix –

Dès le départ, j'ai cadré mon travail autour d'un objectif prioritaire : mettre au maximum à l'abri l'environnement de production tout en garantissant que PRE PROD reste strictement représentatif de la PROD. Les choix que j'ai faits sont le fruit de mon analyse ; je les ai retenus parce qu'ils me paraissaient, dans ce contexte, les plus pertinents pour réduire le risque et supprimer le drift entre environnements.

- Séparer "gel de versions" et "installation" (idée directrice que j'ai proposée) : plutôt que d'installer ce que proposent les dépôts le jour J, je fige d'abord, à date donnée, la liste paquet=version, puis je l'applique plus tard. Ce découplage impose une référence commune et stable : testée en PRE PROD, réappliquée telle quelle en PROD. Selon moi, c'est la méthode la plus fiable pour éviter des écarts invisibles, sources de régressions non reproductibles.

- Introduire une fenêtre de validation PRE PROD avant toute PROD (choix de processus) : la PROD n'applique jamais du "non validé". Elle n'installe que ce qui a

réellement été exécuté et observé en PRE PROD. Cette temporalité protège la PROD et laisse le temps d'identifier/exclure un paquet problématique.

- Réversibilité native via sauvegardes + métadonnées + checksum (décision pour sécuriser les retours arrière) : avant installation, je crée un snapshot exploitable (liste, contexte système, intégrité vérifiée). En cas d'incident, on revient à l'état antérieur en connaissance de cause (vérification SHA256, alertes d'incompatibilité). C'est, de mon point de vue, la façon la plus pragmatique de réduire drastiquement le MTTR (Le MTTR est le temps moyen nécessaire à la réparation d'un système, généralement technique ou mécanique).

- Portabilité multi distributions avec détection automatique (contrainte adressée dès la conception) : je ne voulais pas d'un outil valable uniquement pour une famille Linux. La détection dynamique du gestionnaire de paquets (APT, DNF/YUM, Pacman, Zypper) et l'adaptation des commandes rendent la solution homogène à l'usage et partageable avec d'autres « member firms », en réduisant l'erreur humaine.

- Double interface TUI/CLI (choix délibéré pour l'exploitation) : les opérations manuelles bénéficient d'un menu simple, tandis que l'automatisation s'appuie sur des commandes idempotentes et des codes de retour explicites. J'ai fait ce choix pour concilier la réalité des équipes d'exploitation et les exigences d'industrialisation.

Solution mise en place (architecture et composants) –

Je l'ai structurée en un orchestrateur principal et des scripts spécialisés, avec une organisation de répertoires et une journalisation normalisée.

Orchestrateur : "auto-patching-manager.sh"

- Rôle : point d'entrée unique. Gère l'initialisation, les répertoires (log/, package-list/, package-list-old/, backups/), les permissions, l'affichage d'un menu TUI, le routage des commandes CLI, et les logs.
- Sécurité/robustesse : vérification d'exécution en root ; set -euo pipefail ; gestion des erreurs ; codes de sortie ; nettoyage/maintenance (ex. rotation de logs).
- Multi distributions : détection de la distribution/du gestionnaire (APT, DNF/YUM, Pacman, Zypper) et export des commandes adaptées.
- TUI et automatisation : menu interactif (téléchargement/installation/backup rollback/status/help) et exécution directe des scripts via arguments (setup, download, install, backup/rollback, status, help).
- Journaux et traçabilité : log horodaté avec niveaux (INFO, WARNING, ERROR, SUCCESS, SYSTEM) en console + fichier, pour audit et diagnostic.

Gel des versions : "download.sh"

- Rôle : figer la liste des packages à un instant T pour constituer une "référence" commune PRE PROD/PROD.
- Sorties : liste(s) de paquets par distribution/gestionnaire, horodatées et archivées (package-list/, package-list-old/).

- Intention : découpler “découverte des mises à jour” et “installation”, afin de reproduire exactement en PROD ce qui a été validé en PRE PROD.

Installation contrôlée : “install.sh”

- Rôle : appliquer en PRE PROD puis en PROD la liste gelée, avec options permettant d’imposer une sauvegarde préalable.
- Comportement : s’appuie sur le gestionnaire détecté pour installer précisément ce qui figure dans la liste gelée (cohérence stricte des versions).
- Filet de sécurité : possibilité de forcer la création d’un snapshot avant installation (intégration avec la brique de sauvegarde/rollback).

Sauvegarde et retour arrière : “rollback.sh”

- Rôle : gérer la création, la liste, la restauration et la suppression de sauvegardes.
- Sauvegarde : génère un fichier .backup (inventaire paquets), des métadonnées .metadata (distro, gestionnaire, timestamp, hôte, noyau, taille), un checksum .sha256 (intégrité), et, selon le gestionnaire, un inventaire détaillé.
- Restauration : contrôle de compatibilité (distribution/gestionnaire), vérification d’intégrité (SHA256), confirmations explicites en cas d’écart, et restauration propre des états précédents.
- Traçabilité : commandes list/restore/delete avec retours clairs; affichage lisible (colonnes, tailles, dates, hôte, distro).

Organisation et exploitation :

- Répertoires normalisés : log/ pour les journaux ; package-list/ pour les listes gelées ; package-list-old/ pour l’archivage ; backups/ pour les snapshots.
- Maintenance : nettoyage des logs anciens (seuil conservateur), alerte sur l’espace disque.
- Expérience opérateur : couleur des logs en console, bannières claires, messages d’état explicites.
- Intégration : exécutable en TUI pour l’humain ; en CLI pour l’ordonnanceur (tâches planifiées, codes de retour).

Ma façon de penser –

Déroulé de mes réflexions qui m’ont mené à cette solution en partant de zéro :
J’ai d’abord formulé le problème en termes de risques opérationnels (drift, non réversibilité, non traçabilité) plutôt qu’en termes d’“outils”.

De là, ma réflexion a suivi quatre étapes :

- Stabiliser la variable la plus instable : la version des paquets. D’où l’exigence de geler les versions séparément de l’installation.
- Empêcher l’introduction de nouveautés en PROD : instaurer une fenêtre de validation PRE PROD obligatoire, avec observation et possibilité d’exclure un paquet.
- Rendre l’échec “sûr” : avant toute installation, créer un snapshot vérifiable et restaurable (métadonnées + checksum) pour garantir un retour arrière rapide et documenté.

- Réduire la complexité perçue par l'opérationnel : un orchestrateur unique, un menu TUI clair, et des commandes CLI simples, tout en gérant en interne la diversité des distributions.

À chaque étape, j'ai privilégié la simplicité robuste : des scripts Bash clairs, une séparation nette des responsabilités (orchestrateur vs scripts spécialisés), des logs explicites, et des artefacts concrets (listes, métadonnées, checksums) plutôt que des mécanismes implicites. C'est cette logique qui m'a conduit à l'architecture actuelle.

Limites actuelles et perspectives –

- Dépendance aux dépôts : si une version disparaît des dépôts, une restauration peut nécessiter un miroir/caching interne.
- Portée du rollback : la couche "paquets systèmes" est couverte ; certaines applications peuvent exiger des processus spécifiques (ex. migrations de schéma) en complément.
- Pistes d'amélioration : listes d'exclusion par application critique ; rapport synthèse post opération (HTML/CSV) ; intégration au change management (tickets + artefacts) ; cache/miroir interne des versions gelées ; mode "dry run" listant précisément les actions sans exécuter.

Conclusion –

Cette solution, conçue et réalisée par moi en partant de zéro, répond à la priorité de KPMG : Protéger la production tout en alignant PRE PROD et PROD. En séparant le gel des versions de l'installation, en imposant une validation PRE PROD, en outillant la réversibilité (sauvegardes, métadonnées, checksums) et en gérant nativement la diversité des distributions via un orchestrateur TUI/CLI, elle fournit une gouvernance de patching robuste, explicable et auditable.
Le résultat est un processus reproductible et réversible, adapté à un contexte multi distributions et partageable avec d'autres « member firms » si nécessaire, tout en réduisant le risque et en facilitant l'exploitation au quotidien.

Conclusion

Du point de vue professionnel

Ce stage m'a fait passer d'une approche "technique avant tout" à une véritable posture d'ingénieur orientée production et service. En partant d'un besoin peu cadré, j'ai appris à clarifier les attentes, à transformer une idée en solution industrialisable et à la faire adopter par l'exploitation. Concevoir un dispositif de patch management robuste m'a obligé à intégrer très tôt des exigences souvent sous estimées « traçabilité, réversibilité, conformité, fenêtres de maintenance », et à raisonner en risques concrets plutôt qu'en hypothèses. J'ai gagné en autonomie dans les choix d'architecture, en rigueur dans le code (gestion d'erreurs, logs exploitables) et en sens de la preuve grâce à des artefacts vérifiables et auditables.

Au-delà de la technique, j'ai surtout développé une méthode. J'ai appris à faire évoluer un prototype vers un processus opérable, documenté et compréhensible par d'autres. Les échanges réguliers avec l'équipe m'ont appris à écouter les contraintes du terrain, à expliquer clairement les arbitrages et à assumer des décisions. J'ai découvert la valeur d'un flux de changement maîtrisé : préparer, tester en préproduction, mesurer, déployer, observer, et, si nécessaire, revenir en arrière sans drame. Cette discipline m'a donné confiance pour gérer des sujets transverses où l'infra, la sécu et l'applicatif se rencontrent.

Ce stage m'a également ouvert les yeux sur la réalité opérationnelle d'une très grande entreprise comme KPMG. J'y ai découvert l'ampleur des environnements, la maturité des processus et le niveau d'exigence associé à des écosystèmes à la fois massifs et très développés, à une échelle que je n'aurais pas pu imaginer avant d'y être confronté. Cette immersion m'a permis de comprendre comment se coordonnent gouvernance, sécurité, conformité, exploitation et delivery à grande échelle, et ce que cela implique en termes de rigueur, de traçabilité et de qualité de service.

Enfin, j'ai eu l'occasion de comparer concrètement les approches open source, qui structurent notre formation, avec des solutions éditeurs largement déployées en entreprise. Voir les outils réellement utilisés sur le terrain, leurs fonctionnalités avancées, leurs modèles de support, leur intégration dans des chaînes d'outillage complexes, a été extrêmement formateur. Cette exposition m'a aidé à mieux évaluer les compromis entre coût, maintenabilité, support, time to value et souveraineté technique, et à adopter une vision plus pragmatique : choisir l'outil qui sert le mieux l'objectif, dans un cadre de contraintes réelles.

Je ressors de cette expérience avec des réflexes SRE (Consiste à utiliser des outils logiciels pour automatiser les tâches d'infrastructure informatique telles que la gestion du système et la surveillance des applications) / DevOps plus affirmés, une meilleure lecture des impacts opérationnels et une éthique de travail orientée clarté, responsabilité et transmission.

Plus que des lignes de code, j'emporte une manière de penser la production : anticiper, documenter, prouver, servir les équipes et composer avec les contraintes d'une organisation

de grande taille. Ces acquis constituent un socle solide pour la suite de mon parcours professionnel.

Du point de vue personnel

À l'issue de ce stage, je me découvre partagé entre deux trajectoires possibles.
D'un côté, je ne suis pas certain de vouloir poursuivre immédiatement au sein d'une entreprise aussi vaste que KPMG. Cela peut paraître paradoxal tant le niveau y est élevé : on n'y côtoie pas seulement des talents, mais de véritables experts, dont l'exigence et la maîtrise sont inspirantes.
De l'autre, je pressens qu'en début de carrière, évoluer dans une structure plus compacte, où les rôles sont moins figés et les périmètres plus transverses, peut offrir l'opportunité de "se faire les armes" sur un spectre plus large : toucher à l'architecture comme à l'exploitation, à la sécurité comme au delivery, et apprendre à relier les points dans des contextes variés.

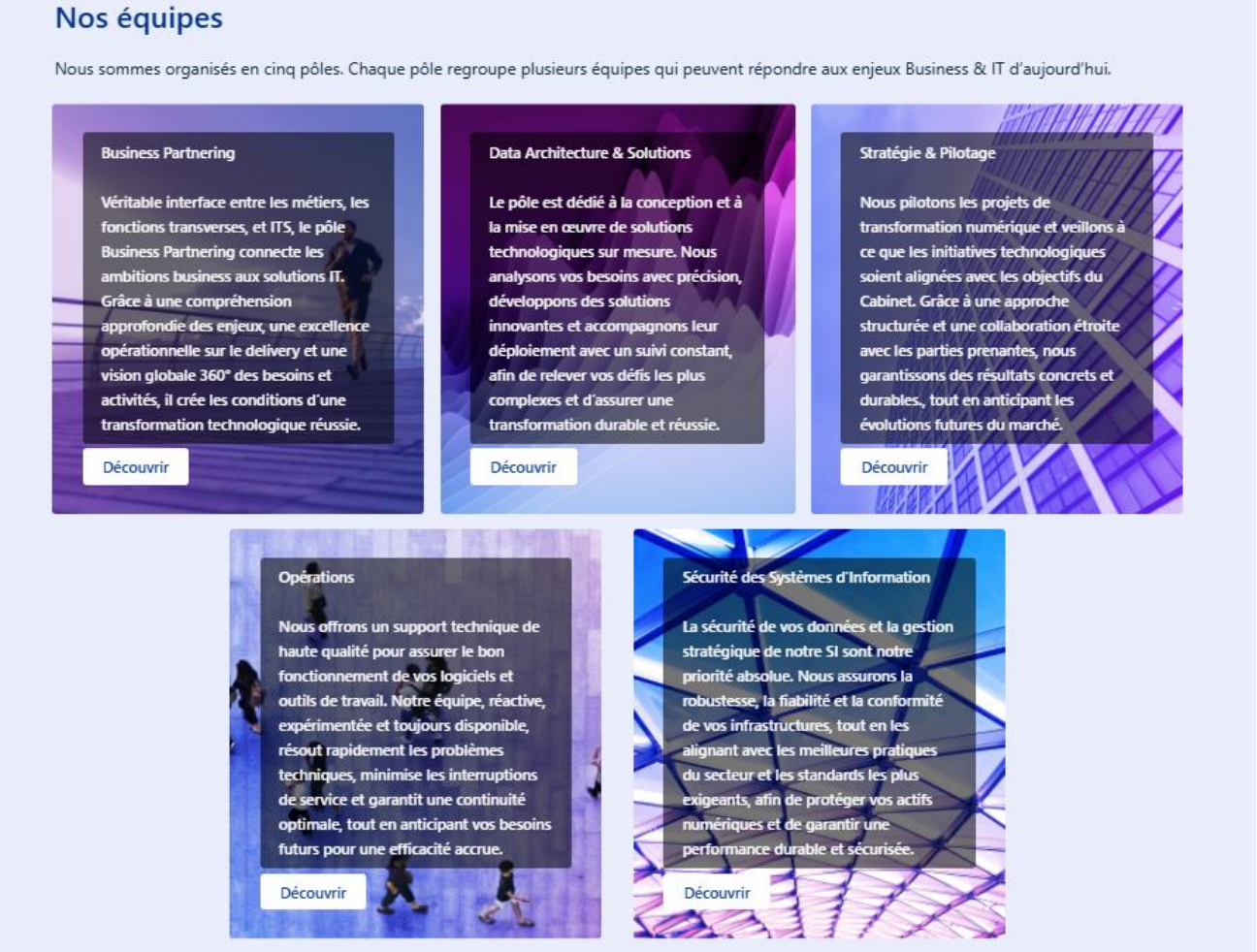
Cette hésitation n'est pas un frein ; elle est le signe d'un choix à somme positive.
Travailler aux côtés d'équipes d'un niveau exceptionnel tire vers le haut, impose de la rigueur et accélère l'apprentissage par l'exemple.
Évoluer dans une structure plus agile oblige, lui, à développer une polyvalence opérationnelle, à prendre des responsabilités tôt et à apprendre vite par la pratique. Je suis pressé de voir ce que l'avenir me réserve, d'autant plus que ne pas trancher aujourd'hui entre ces deux voies signifie, à mes yeux, qu'aucune n'est mauvaise : chacune porte des apprentissages précieux, chacun à son rythme et à son échelle.

En somme, je suis confiant : que je choisisse l'excellence cumulative des grandes organisations ou l'intensité formatrice d'une structure plus petite, il y a une place pour Samuel Decarnelle dans ce domaine. Je continuerai d'avancer avec curiosité, exigence et envie d'apporter de la valeur, convaincu que le meilleur chemin se dessinera en marchant.

Annexes

Pour consulter mes scripts, cliquez ici : https://github.com/Asashi-Git/scripts/tree/main/stage-decarnelle-samuel-kpmg

Les différentes équipes ITS :



Autres ressources utiles –

- Nature du réseau : KPMG (en.wikipedia.org)
- Effectifs et activité : KPMG (en.wikipedia.org)
- KPMG un membre des “Big Four” (fr.wikipedia.org)
- Les valeurs de KPMG (kpmg.com)