RAPPORT DE SYNTHÈSE D'ALTERNANCE

Développeur Back-End Java

Grégoire MAROUSÉ

2ème année de BUT Informatique

IUT de Vannes : 8 rue Michel de Montaigne, 56000 Vannes

Crédit Agricole Technologies et Services : 21 avenue de Keranguen, 56000 Vannes

Maître d'alternance - CA-TS Vannes	Tuteur pédagogique - IUT de Vannes
Rachid Hannaoui	Jean-François Kamp

Référents alternance - IUT de Vannes
G. Kerbellec & T. Godin

Année universitaire 2025 – 2026

Table des matières

I. Présentation de l'entreprise

1.1 Économie et positionnement

Crédit Agricole Technologies et Services (CA-TS) est une filiale informatique du groupe Crédit Agricole. Sa mission est la conception, la fabrication et la maintenance du système d'information bancaire des 39 Caisses régionales du groupe. Avec environ 1 800 collaborateurs répartis sur six sites en France, CA-TS s'impose comme un acteur central de la transformation numérique du premier groupe bancaire mutualiste français.

Son périmètre couvre un spectre métier large : développement agile, DevOps, Cloud, cybersécurité, gestion de la donnée, expérience utilisateur (UX), automatisation robotisée (RPA) et numérique responsable. Ces activités sont organisées en tribus et en squads autonomes, fonctionnant selon des principes issus des méthodes agiles à l'échelle (SAFe, Scrum).

En tant qu'entité captive du groupe Crédit Agricole, CA-TS ne vend pas ses services sur un marché ouvert. Son modèle repose sur des contrats de service internes aux Caisses régionales, lui assurant une stabilité structurelle. Cela ne la soustrait pas pour autant aux pressions de marché : la nécessité d'accélérer la modernisation du système d'information face aux néobanques et aux grandes plateformes technologiques crée une tension permanente entre continuité et innovation.

1.2 Localisation

CA-TS est implanté sur six sites en France : Paris, Vannes, Nantes, Annecy, Lyon et Montpellier. Cette répartition permet une couverture nationale et une proximité avec les principales Caisses régionales partenaires.

Mon affectation s'effectue sur le site de Vannes, en Bretagne, où sont notamment hébergées des équipes dédiées aux services de paiement digital.

1.3 L'équipe Paiements Digitaux

Au sein du site de Vannes, j'intègre la Squad Paiements Digitaux, une équipe pluridisciplinaire organisée selon la méthode Scrum. Elle est chargée du développement et de la maintenance des API d'enrôlement liées aux services de paiement mobile du Crédit Agricole : l'application CA, l'intégration avec Apple Pay et Samsung Pay, ainsi que le portefeuille électronique client.

L'équipe est composée de 8 à 10 membres aux rôles complémentaires :

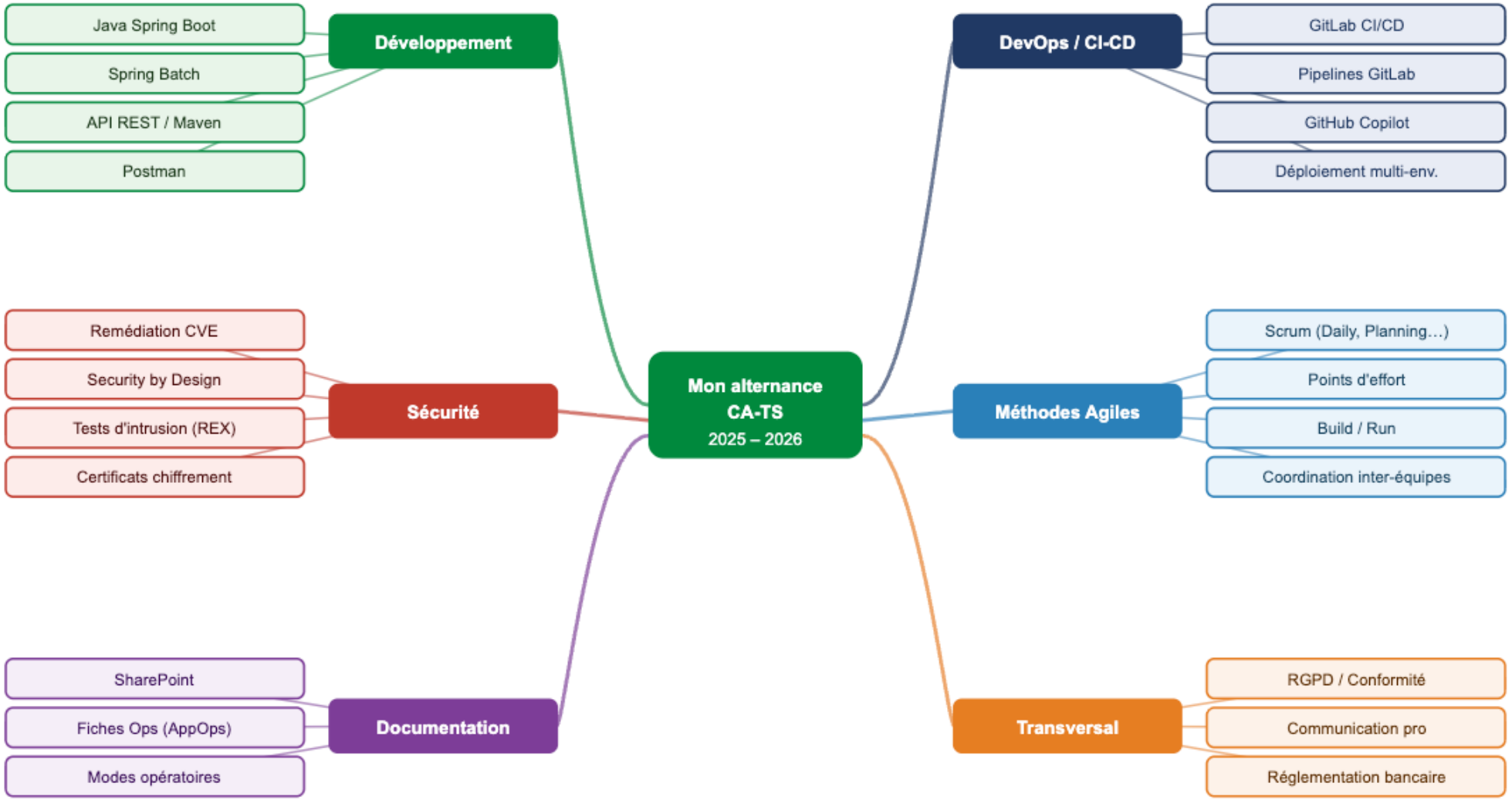
Membre	Rôle	Responsabilités
Aliénor	Product Owner	Définit les priorités, représente le métier
Abdelmounim (arrivé en janvier)	Scrum Master	Facilite les cérémonies, lève les obstacles
Rachid Hannaoui	Leader Tech / Maître d'alternance	Oriente les choix techniques, garantit la qualité
Johnatan	Développeur back-end	Développement API
Oumayma	Développeuse back-end	Développement API
Adama (parti en mars)	Business Analyst	Tests bout-en-bout, validation métier
Emeline (arrivée en décembre)	Business Analyst	Tests bout-en-bout, validation métier
Annaël (arrivée en mars)	Business Analyst	Tests bout-en-bout, validation métier
Grégoire MAROUSÉ	Développeur alternant	Développement API
Steven (arrivé en mai)	Développeur Cobol/Java	Développement API et Mainframe
Benoît	Développeur Front	Personne du Chapitre digital front associé à notre équipe : Développement des BFF.

Cette équipe est née en juillet 2025 de la séparation d'une équipe en 3 nouvelles équipes. En conséquence, tout au long de l'année, il y a eu de nombreux mouvement dans les ressources humaines, enrichissant la dynamique collective et offrant une perspective extérieure utile sur les pratiques de l'équipe. Cela m'a permis d'observer la construction et la structuration d'une nouvelle équipe dans une grande entreprise et de faire la rencontre de plein de personnes avec des directions différentes.

II. Présentation des missions

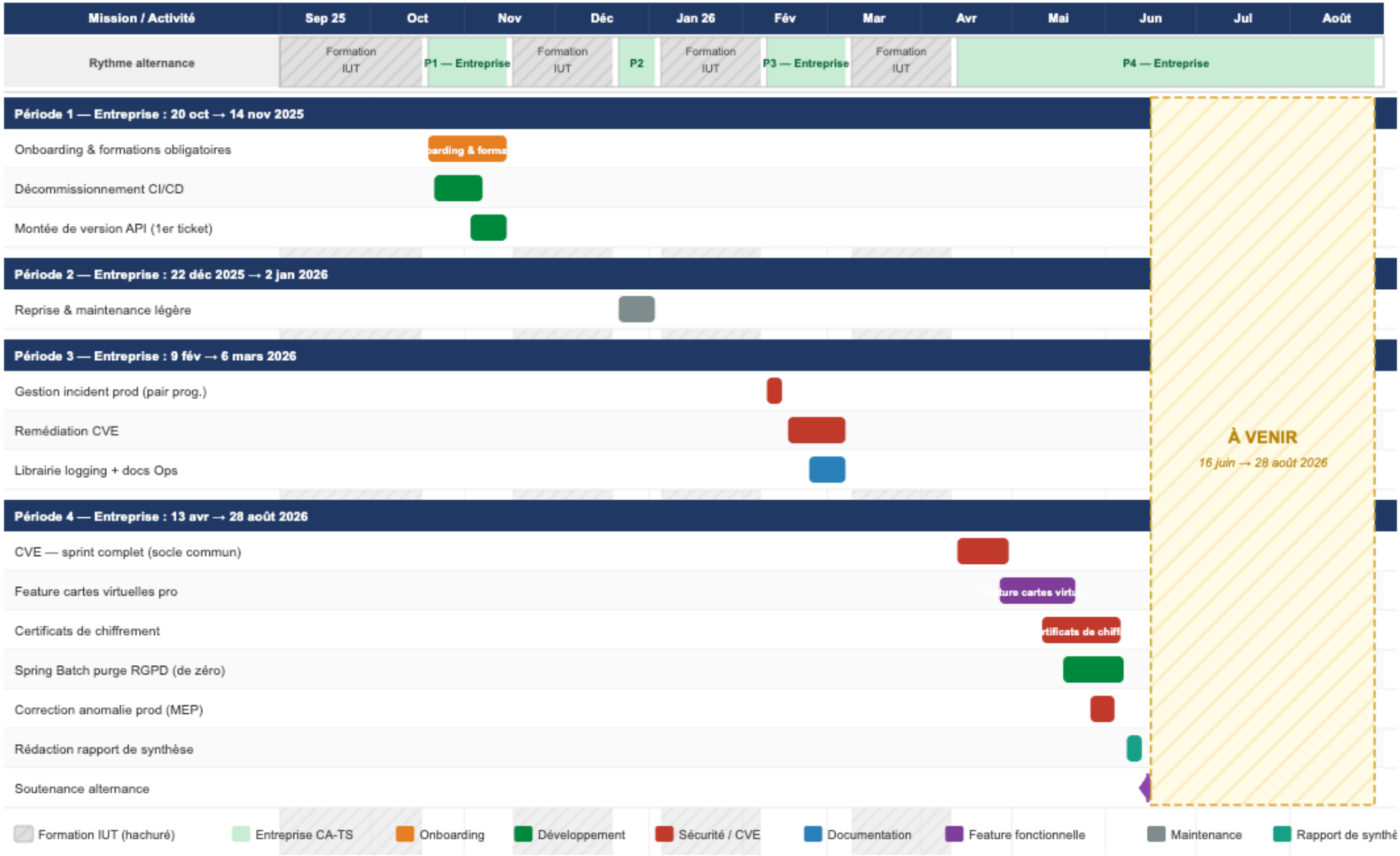
2.1 Vue d'ensemble des compétences

La carte mentale ci-dessous synthétise les grandes familles de compétences mobilisées au cours de l'alternance. Elle illustre la diversité des activités réalisées, qui ne se limitent pas au développement logiciel mais englobent également la sécurité applicative, la coordination inter-équipes et la dimension organisationnelle propre au travail en environnement bancaire.



2.2 Chronologie de l'alternance

Le diagramme de Gantt présente la répartition des missions sur les cinq périodes en entreprise de l'année universitaire 2025-2026. Chaque période alterne avec des semaines de formation à l'IUT, selon le calendrier de l'alternance BUT Informatique.



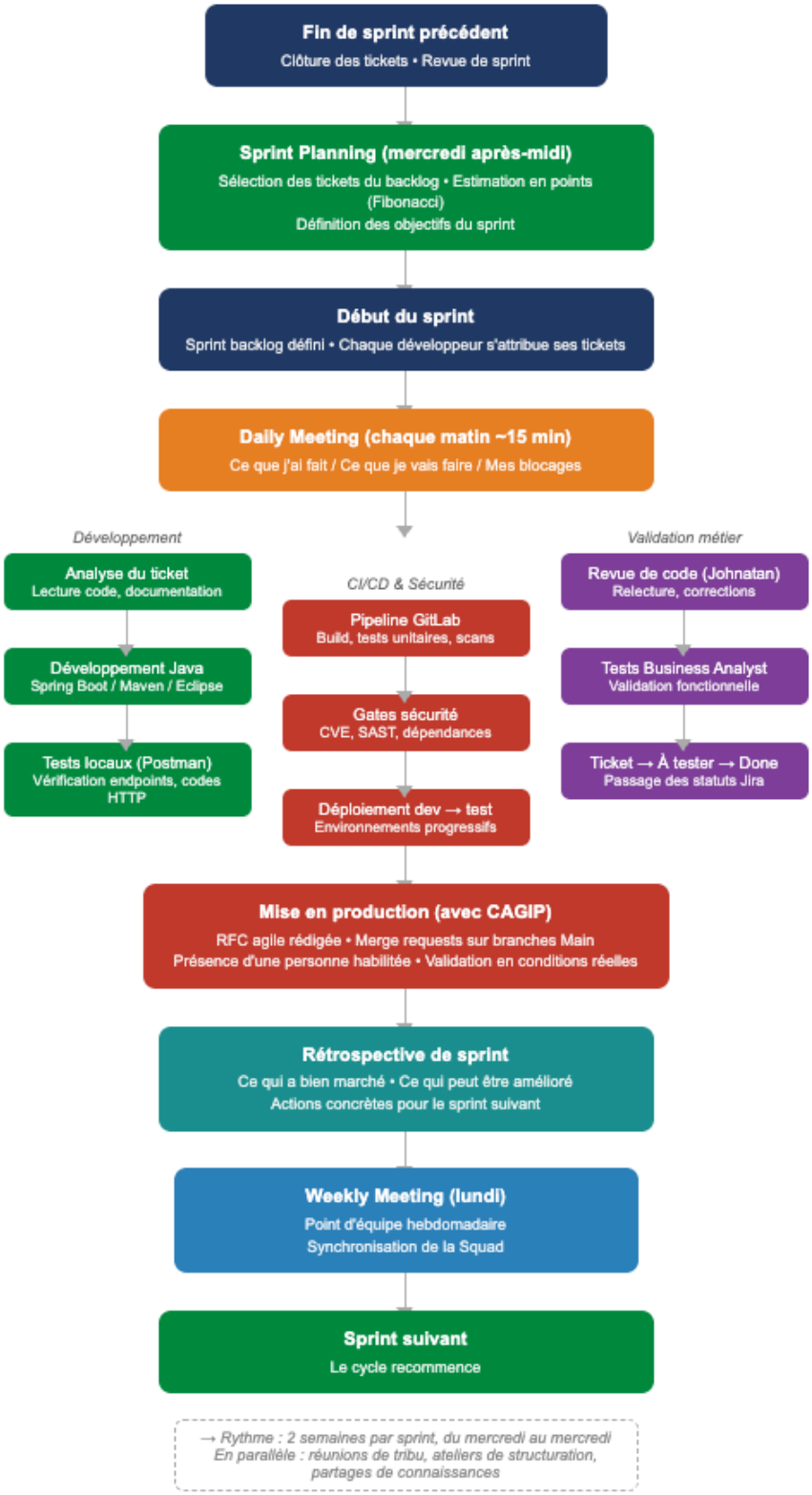
2.3 Tableau récapitulatif des missions

N°	Mission	Période	Domaine	Compétences développées	Technologies
1	Onboarding et formations internes obligatoires (Agile, cybersécurité, conformité, paiements)	P1 oct. 2025	Formation / Découverte	Découverte du SI bancaire, méthode Scrum, réglementation groupe	GitLab, Jira, outils internes CA-TS
2	Décommissionnement d'un événement via la chaîne CI/CD	P1 oct. 2025	Infrastructure / DevOps	Intégration continue, déploiement par environnements, gestion des pipelines	GitLab CI/CD, Java
3	Montée de version d'une API (premier ticket technique)	P1 nov. 2025	Maintenance technique	Gestion des dépendances Maven, tests API, revue de code	Java, Maven, Spring Boot, Postman
4	Reprise et maintenance légère post-formation	P2 déc. 2025 / jan. 2026	Maintenance	Continuité de travail, prise d'information, autonomie de reprise	Java, GitLab
5	Gestion d'un incident urgent en production	P3 fév. 2026	Gestion d'incident	Pair programming, diagnostic, correction, déploiement d'urgence	Java, Maven, Spring Boot, GitLab CI/CD
6	Remédiation de vulnérabilités CVE (première série)	P3 fév. / mars 2026	Sécurité applicative	Analyse d'impact, montée de version de librairies, passage des gates sécurité CI/CD	Maven, GitLab CI/CD, Spring Boot
7	Évolution de la librairie de logging interne	P3 mars 2026	Développement	Conception d'une évolution, tests unitaires, revue de code avec le Lead Tech	Java, Spring Boot, Maven
8	Co-conception de l'outil d'automatisation des jeux de données	P3 mars 2026	Automatisation / Collaboration	Expression de besoins, débogage collaboratif, coordination avec l'automaticien	Outils internes, Java
9	Analyse d'impact de la montée de version Java 25 + Spring Boot	P3 mars 2026	Analyse technique	Compréhension de l'architecture applicative, gestion des dépendances, rédaction d'un compte rendu	Java 25, Spring Boot, Maven
10	Structuration de la documentation SharePoint de la Squad	P3 et P4	Documentation	Rédaction technique, HTML/CSS natif, conception orientée onboarding	SharePoint, HTML

11	Remédiation CVE et montées de version du socle commun (sprint complet)	P4 avr. 2026	Sécurité / Maintenance	Coordination inter-équipes, gestion multi-solutions en parallèle, MEP	Maven, GitLab CI/CD, Spring Boot
12	Feature cartes virtuelles pour les clients professionnels	P4 avr. / mai 2026	Développement fonctionnel	Évolution fonctionnelle, mise à jour de librairie partagée inter-équipes, déploiement en homologation	Java, Spring Boot, Maven, GitLab CI/CD
13	Renouvellement des certificats de chiffrement	P4 mai / juin 2026	Sécurité / Coordination	Tests de bout en bout, pilotage multi-acteurs en autonomie, mise à jour des collections Postman	Certificats TLS, Postman, GitLab CI/CD
14	Développement du Spring Batch de purge RGPD (de zéro)	P4 mai / juin 2026	Développement from scratch	Architecture Spring Batch, configuration YAML multi-env., CI/CD, conformité RGPD	Spring Batch, Java, JDBC, GitLab CI/CD
15	Correction d'une anomalie en production et mise en production autonome	P4 juin 2026	Maintenance corrective	Diagnostic, correctif, rédaction de la RFC, procédure MEP complète en autonomie	Java, Spring Boot, GitLab CI/CD
16	Rédaction du rapport de synthèse d'alternance et préparation de la soutenance	P4 juin 2026 (semaine de congés)	Transversal	Synthèse et communication des compétences acquises, rédaction professionnelle	Suite Office

2.4 Déroulement d'un sprint

Les missions de l'équipe sont organisées en sprints de deux semaines, du mercredi au mercredi. Le schéma ci-dessous représente le cycle d'un sprint standard tel que je l'ai vécu tout au long de l'année. La régularité de ce rythme constitue un cadre structurant qui favorise la planification, la transparence et la livraison continue. Des essaies et des discussions sont actuellement en cours pour ajuster la durée des sprint plutôt à trois semaines au lieu de deux.



III. Veille technologique

Réalisée en amont du démarrage de l'alternance, ma veille technologique portait sur l'environnement technique attendu dans le secteur bancaire : les raisons du choix de Java et de Spring Boot, et la comparaison avec les principales alternatives disponibles.

3.1 Contexte : pourquoi Java dans un SI bancaire ?

Le secteur bancaire est l'un des environnements informatiques les plus exigeants au monde. Les systèmes en place traitent des millions de transactions par jour, avec des exigences absolues de fiabilité, de sécurité et de conformité réglementaire. Deux technologies historiques dominent ce paysage :

- COBOL, présent depuis les années 1960, assure encore aujourd'hui les traitements de nuit, les virements et les opérations batch sur les systèmes mainframe. Difficile à faire évoluer, il reste conservé pour sa fiabilité exceptionnelle (disponibilité 99,999 %).
- Java, dominant depuis les années 2000, sert de couche intermédiaire entre le legacy COBOL et les interfaces modernes. Il est le langage de référence pour les nouvelles API REST exposées aux applications mobiles et web.

Spring Boot, framework fondé sur Spring, simplifie et accélère le développement en Java en proposant une auto-configuration, une gestion centralisée des dépendances et une intégration native des bonnes pratiques de sécurité. C'est la brique technique centrale de l'ensemble des solutions de la Squad.

3.2 Analyse des alternatives

La veille a permis de comparer Java / Spring Boot / API REST face aux principaux écosystèmes concurrents, selon six critères pertinents pour un contexte bancaire.

Écosystème	Coût mise en œuvre	Performance / Scalabilité	Maturité	Sécurité / Conformité	Facilité dev.	Interopérabilité
Java + Spring Boot + REST	Moyen à élevé	Bonne	Très élevée	Excellente	Moyen	Excellente
.NET Core (C# + ASP.NET)	Moyen	Très bonne	Élevée	Très bonne	Accessible	Bonne
Node.js (Express / NestJS)	Faible à moyen	Moyenne	Moyenne	Moyenne	Très bonne	Bonne
Deno (TypeScript natif)	Faible init.	Bonne	Immature	Excellente sécurité	Très bonne	Limitée
Go (Golang + gRPC/REST)	Moyen	Excellente	Moyenne	Moyenne à bonne	Bonne	Bonne
SOAP / Monolithes legacy	Élevé (maintenance)	Faible	Ancienne	Bonne histo.	Faible	Faible

3.3 Conclusion de la veille

L'analyse confirme que l'écosystème Java + Spring Boot + REST est le choix le plus pertinent pour le développement d'API bancaires. Sa maturité, sa robustesse éprouvée et son adoption massive dans le secteur en font une base de confiance, parfaitement adaptée aux exigences de sécurité, de conformité et de scalabilité.

Les alternatives présentent des atouts sectoriels réels : .NET Core sur les environnements Microsoft, Go pour les microservices haute performance, Node.js pour les interfaces réactives légères. Chez CA-TS, cette complémentarité existe déjà : C# est utilisé pour certains autres services du groupe, illustrant une logique de bonne technologie au bon endroit plutôt qu'une domination monolithique d'un seul langage.

> Lien avec la pratique : cette veille et le PoC associé se sont révélés directement utile en entreprise. La compréhension de Spring Boot et de l'écosystème Java m'a permis d'être opérationnel plus rapidement lors de la prise en main des différentes solutions de la Squad.

IV. Environnement économique et concurrence

4.1 Le secteur bancaire français en mutation

Pendant des décennies, le secteur bancaire français a fonctionné dans un cadre relativement stable : quelques grands acteurs historiques (Crédit Agricole, BNP Paribas, Société Générale, LCL, Caisse d'Épargne, Banque Postale), une réglementation protectrice et des clients peu mobiles. Ce modèle est aujourd'hui soumis à une transformation profonde sous l'effet conjugué de trois grandes forces.

Facteurs réglementaires

- La directive DSP2 (révisée en DSP3) impose aux banques l'ouverture de leurs APIs aux acteurs tiers, créant un régime d'open banking qui rompt le monopole historique de l'accès aux données clients.
- Le RGPD renforce les obligations de protection et de gestion des données personnelles, avec un impact direct sur les systèmes d'information bancaires, en particulier sur les durées de rétention et les politiques de purge.
- Bâle III et IV imposent des exigences renforcées en fonds propres, augmentant la pression sur les marges et incitant à la réduction des coûts opérationnels, notamment par la mutualisation des infrastructures informatiques.

Contexte macro-économique

- La remontée des taux directeurs (2022-2024) a restauré les marges nettes des banques après une décennie de taux bas, mais elle s'accompagne d'une pression accrue sur les coûts liée à l'inflation.
- La fermeture accélérée des agences physiques (environ 15 % de réduction sur cinq ans en France) déplace la relation client vers les canaux numériques, en particulier le mobile.
- La hausse du risque cyber (attaques par rançongiciel, fraudes, phishing bancaire) contraint les établissements à investir massivement en sécurité applicative, au détriment parfois des projets d'innovation.

Transition numérique

Environ 80 % des clients du Crédit Agricole utilisent désormais l'application mobile comme canal principal de leur relation bancaire. Cette évolution transforme les exigences de qualité, de performance et de sécurité des systèmes back-end, qui doivent désormais gérer des volumes de requêtes mobiles en temps réel, tout en assurant la continuité des traitements batch hérités.

4.2 Le marché du paiement mobile en France

Le paiement mobile en France est un marché en forte croissance, structuré autour de plusieurs familles d'acteurs aux logiques économiques très différentes.

Les wallets des constructeurs (Apple Pay, Google Pay, Samsung Pay)

Lancés à partir de 2014 en France, les wallets embarqués dans les smartphones se sont imposés comme le mode de paiement sans contact de référence. Leur adoption est massive : en 2024, Apple Pay est accepté dans plus de 98 % des terminaux sans contact en France. Leur modèle économique repose sur une commission prélevée sur chaque transaction, directement supportée par les banques émettrices des cartes estimée entre 0,15 % et 0,17 % du montant pour Apple Pay. Ce modèle crée

une dépendance structurelle des banques envers des acteurs technologiques américains et érode mécaniquement leurs marges sur les paiements.

Sur le plan technique, ces wallets imposent des protocoles stricts (EMVCo, NFC, tokenisation) et des certifications applicatives exigeantes, qui conditionnent directement le travail des équipes comme la Squad Paiements Digitaux.

Wero : la réponse européenne

Lancé en 2023, Wero est le service de paiement mobile développé par EPI (European Payments Initiative), un consortium qui réunit les principales banques françaises, allemandes, belges et néerlandaises, dont le Crédit Agricole. Wero est le successeur de Paylib, avec une ambition élargie : devenir le wallet de référence européen face à Apple Pay et Google Pay. Il permet les virements instantanés entre particuliers, les paiements en commerce et, à terme, des paiements e-commerce.

Pour le Crédit Agricole, Wero représente une opportunité stratégique de reprendre la main sur la relation client de paiement, en internalisant des revenus jusqu'ici captés par les géants américains. Le déploiement de Wero auprès des clients CA est une priorité, et CA-TS en assure une part de l'infrastructure technique.

Les néobanques : un modèle disruptif

Revolut, N26, Lydia (devenu Sumeria), Orange Bank (maintenant fermée) et quelques autres acteurs ont profondément modifié les attentes des clients en proposant des interfaces ultra-fluides, des frais réduits et une ouverture internationale. Leur force repose sur des architectures techniques modernes (Go, Node.js, APIs ouvertes) et une absence de legacy, ce qui leur permet un time-to-market très court.

En contrepartie, leur absence de réseau physique, leur couverture réglementaire limitée et leur manque de profondeur de bilan constituent des points faibles face aux banques traditionnelles. Revolut, valorisée à 45 milliards de dollars en 2024, commence toutefois à obtenir des licences bancaires en Europe, renforçant la pression concurrentielle.

Acteur	Type	Positionnement	Atout principal
Apple Pay / Samsung Pay	Wallet constructeur	Paiement sans contact universel	Adoption massive, UX fluide
Wero (EPI)	Wallet bancaire européen	Virement instantané, alternative CA	Souveraineté européenne, intégration bancaire
Revolut / N26	Néobanque	Compte mobile international	Time-to-market, frais faibles
Lydia / Sumeria	Fintech française	Paiement entre proches, compte courant	Communauté, interface intuitive
Crédit Agricole / CA-TS	Banque traditionnelle + ESN captive	Paiement mobile + portefeuille CA	Confiance, réseau, conformité

4.3 Positionnement du Crédit Agricole et de CA-TS

Le Crédit Agricole est la première banque de détail en France, avec environ 25 % de parts de marché en bancarisation et près de 52 millions de clients dans le monde. Sa structure mutualiste, reposant sur les 39 Caisses régionales, lui confère un ancrage territorial fort et un niveau de confiance client élevé, deux atouts que les néobanques peinent à reproduire.

Face à la digitalisation accélérée des usages, le groupe a choisi d'accélérer sa transformation numérique de l'intérieur plutôt que de sous-traiter massivement à des prestataires externes. CA-TS est le moteur opérationnel de cette stratégie : c'est elle qui conçoit, développe et maintient les solutions qui permettent à des dizaines de millions de clients de payer avec leur téléphone, de gérer leurs cartes ou de consulter leurs comptes.

La stratégie technologique de CA-TS peut se résumer en une formule : modernisation progressive sans rupture. Il ne s'agit pas de remplacer du jour au lendemain les systèmes COBOL vieux de quarante ans qui traitent les virements et les opérations de nuit, mais de les encapsuler progressivement derrière des APIs Java modernes, tout en migrant les nouvelles fonctionnalités vers des architectures microservices. La Squad Paiements Digitaux s'inscrit pleinement dans cette logique.

4.4 Enjeux pour la Squad Paiements Digitaux

Au sein de cet environnement, les enjeux concrets pour la Squad se déclinent sur plusieurs axes :

- Sécurité et conformité : les APIs d'enrôlement manipulent des données sensibles (liaisons cartes, tokens de paiement). Chaque déploiement est soumis à des contrôles automatisés (gates CI/CD) et des audits réguliers (tests d'intrusion externalisés). La gestion des CVE et le respect des standards PCI-DSS sont des exigences permanentes, non négociables.
- Time-to-market : la concurrence des néobanques et des wallets constructeurs exerce une pression sur la rapidité de déploiement des nouvelles fonctionnalités. L'organisation en sprints de deux semaines vise précisément à raccourcir les cycles de livraison.
- Conformité RGPD : les données liées aux enrôlements (tentatives, erreurs, historiques) ne peuvent pas être conservées indéfiniment en base. La mise en place de mécanismes de purge automatique des données obsolètes est une obligation légale directement traduite en tickets techniques, comme le Spring Batch de purge développé.
- Interopérabilité : l'intégration avec Apple Pay, Samsung Pay et Wero impose de respecter des protocoles stricts définis par des acteurs extérieurs. Toute mise à jour de certificat de chiffrement, de librairie partagée ou de version API nécessite une coordination inter-équipes et des tests de bout en bout rigoureux.

V. Mission détaillée : l'évolution continue des API d'enrôlement

5.1 Mise en contexte avec l'objectif global d'apprentissage

Contrairement à certains alternants dont le travail s'articule autour d'un projet unique et continu, mon positionnement au sein de la Squad consiste à contribuer à l'évolution permanente d'un ensemble de solutions existantes. La Squad maintient et développe plusieurs API et librairies interconnectées, qui constituent ensemble le socle technique du paiement mobile Crédit Agricole côté back-end.

Cette organisation reflète la réalité des environnements de développement en production : un produit bancaire n'est jamais « terminé ». Il évolue continuellement pour répondre à de nouveaux besoins fonctionnels, corriger des anomalies détectées en production, se mettre en conformité avec les nouvelles réglementations et maintenir sa sécurité face à des menaces en constante évolution.

L'objectif pédagogique de cette mission est donc de me faire vivre la réalité d'un poste de développeur back-end en entreprise, dans toute sa diversité : du débogage d'un incident urgent à la conception d'un composant de zéro, en passant par la gestion des dépendances, la documentation et la coordination avec des équipes partenaires.

> Le périmètre technique de la Squad : API REST en Java Spring Boot, Maven pour la gestion des dépendances, GitLab CI/CD pour l'intégration et le déploiement continus, Postman pour les tests d'API. Le système legacy COBOL est accessible en lecture via des appels à un mainframe IBM, exposé par l'outil Quick3270.

5.2 Panorama des missions de l'année

Sur l'ensemble de l'année, les activités techniques peuvent se regrouper en trois grandes familles, que j'ai abordées à des degrés d'autonomie croissants.

Maintenance de sécurité : la gestion des vulnérabilités

La sécurité applicative est une préoccupation permanente dans un SI bancaire. Les CVE (Common Vulnerabilities and Exposures) sont des vulnérabilités identifiées dans des bibliothèques open source utilisées par les projets. La chaîne CI/CD GitLab embarque des outils de scan automatique qui bloquent le déploiement si une vulnérabilité est détectée (les « gates » de sécurité).

J'ai traité des remédiations CVE à plusieurs reprises au cours de l'année, selon un cycle bien défini :

- Identification : la CVE est signalée par le scan CI/CD ou remontée par l'équipe sécurité.
- Analyse d'impact : identification des dépendances affectées dans les fichiers POM Maven, évaluation du risque.
- Correction : montée de version de la ou des librairies concernées dans le POM parent ou dans les projets impactés.
- Tests en local : validation via Postman que les endpoints fonctionnent correctement après la mise à jour.
- Déploiement et gates : passage des pipelines GitLab avec vérification du passage des scans de sécurité.
- Transmission pour mise en production : création de la RFC agile, revue de code avec Johnatan, validation par la business analyste.

Ce travail, répétitif en apparence, est stratégique : une vulnérabilité non corrigée dans une librairie de gestion des tokens de paiement pourrait exposer les données de millions de clients. La rigueur de cette démarche est la première ligne de défense applicative.

Au sujet de la sécurité, l'équipe fait passer régulièrement des tests d'intrusion sur les solutions. Il faut alors corriger les vulnérabilités et problèmes remontés par le prestataire dans le même cycle que pour une CVE.

Évolution fonctionnelle : les cartes virtuelles professionnels

Une part importante du travail de la Squad consiste à mettre en œuvre des évolutions fonctionnelles demandées par les Caisses régionales ou les équipes métier. La feature « cartes virtuelles professionnels » illustre bien ce type de mission : elle consistait à étendre le service de cartes virtuelles existant aux clients professionnels du Crédit Agricole.

Techniquement, cette évolution impliquait la mise à jour d'une librairie commune au produit carte bancaire, partagée avec d'autres équipes (ce qui impose une coordination inter-équipes préalable), puis la mise à jour des ressources API spécifiques à la Squad. Le travail s'est étalé sur un sprint complet, avec des ajustements en cours de route jusqu'à un déploiement en environnement d'homologation pour validation par la Business Analyst.

Cette mission a développé ma capacité à travailler dans un périmètre technique partagé entre plusieurs équipes, à gérer les dépendances croisées et à adapter mon organisation face aux imprévus.

Évolution technique : le Spring Batch de purge RGPD

Le projet Spring Batch de purge de base de données constitue le sujet le plus structurant de mon année. C'est le premier projet que j'ai construit de zéro, sans existant à maintenir, ce qui représente une dimension nouvelle par rapport aux missions précédentes.

Contexte et enjeu RGPD

Les API d'enrôlement de la Squad génèrent et conservent des données en base de données à chaque opération d'enrôlement : tentatives, succès, erreurs, métadonnées de session. Ces données, nécessaires pour le suivi et le débogage, ne peuvent pas être conservées indéfiniment au regard du Règlement Général sur la Protection des Données (RGPD). Au-delà d'une certaine période de rétention définie par la politique interne du groupe, ces enregistrements doivent être supprimés.

Un mécanisme de purge existait antérieurement, mais il ne fonctionnait plus correctement. La Squad a décidé de le remettre à plat en développant un nouveau composant selon les standards architecturaux actuels de l'équipe.

Qu'est-ce que Spring Batch ?

Spring Batch est un framework open source de l'écosystème Spring, conçu pour le traitement de volumes importants de données en mode batch (par opposition aux traitements en temps réel). Il est particulièrement adapté aux opérations périodiques : imports massifs, calculs nocturnes, purges de bases de données. Il fournit une infrastructure standardisée comprenant :

- Un Job : l'unité de traitement de haut niveau, correspondant à l'exécution de la purge.
- Des Steps : les étapes séquentielles du Job (par exemple : identifier les enregistrements obsolètes, les supprimer, journaliser le résultat).
- Un ItemReader : lit les données sources par lots (chunks) depuis la base de données via JDBC.

- Un ItemWriter : effectue les suppressions en base par lots, garantissant les performances et la cohérence transactionnelle.
- Un mécanisme de reprise (restart/retry) : en cas d'erreur, le Job peut reprendre là où il s'est arrêté sans retraiter les enregistrements déjà purgés.

Cette architecture par lots est directement héritée de la logique des traitements batch historiques en COBOL, ce qui illustre la continuité technique entre le legacy et les nouvelles architectures Java.

Développement et difficultés rencontrées

Le développement s'est appuyé sur un projet exemple présent sur le GitLab interne de l'entreprise, servant de base architecturale validée. Les principales étapes ont été :

- Configuration des dépendances Maven : intégration de Spring Batch, Spring Data JPA, des drivers JDBC spécifiques aux environnements de base de données de l'équipe.
- Développement des DAO (Data Access Objects)
- Configuration des paramètres par environnement : les secrets de connexion, les URLs JDBC et les paramètres de rétention diffèrent entre les environnements de développement, de test, d'homologation et de production. Cette configuration via les fichiers YAML a représenté une partie significative du travail.
- Mise en place du pipeline CI/CD GitLab : création du dépôt, configuration du projet de déploiement, passage des gates de sécurité.

La phase de déploiement a été la plus complexe : de nombreux paramètres dans les fichiers YAML ont nécessité des adaptations, impliquant une session de travail de deux heures et demie avec Johnatan pour tout remettre en ordre par essais successifs. Cette expérience m'a appris à être plus méthodique dans la lecture des fichiers de configuration et à ne pas sous-estimer la complexité de l'infrastructure.

5.3 Mise en regard avec la veille technologique

Le développement du Spring Batch illustre directement les enseignements de la veille technologique. Spring Batch est une extension naturelle de l'écosystème Spring Boot : il partage les mêmes principes d'injection de dépendances, de configuration automatique et de gestion transactionnelle. Choisir Spring Batch pour ce type de traitement est la conséquence directe du choix de Java et Spring Boot comme stack principale.

Par ailleurs, la logique des traitements batch lire des données en base, les filtrer selon des critères, les supprimer par lots est un héritage direct de l'architecture des systèmes bancaires historiques en COBOL. Le fait de retrouver cette même logique dans un framework Java moderne illustre la continuité entre le legacy et les nouvelles architectures, et confirme l'analyse de la veille : Java est le standard bancaire précisément parce qu'il est capable d'hériter de la rigueur des architectures batch tout en s'intégrant aux interfaces modernes.

Les alternatives analysées (Node.js, Go) n'auraient pas offert la même maturité pour ce type de traitement : l'écosystème Spring Batch en Java est directement pensé pour les environnements critiques, avec des mécanismes de reprise, de monitoring et de traçabilité que les frameworks équivalents en Go ou Node.js ne proposent pas avec le même niveau de maturité industrielle.

5.4 Cycle de vie, futur et devenir en production

Les travaux réalisés au cours de l'année ne s'inscrivent pas tous dans le même cycle de vie. Certains sont instantanément absorbés dans le flux continu de la maintenance, d'autres constituent des composants durables qui continueront d'évoluer, d'autres encore sont en attente d'une mise en production qui interviendra après la soutenance. Cette hétérogénéité est précisément ce qui caractérise le travail de développeur back-end en environnement bancaire : on ne livre pas un produit, on contribue à un système vivant.

Le cycle de vie des contributions selon leur nature

Les remédiations CVE illustrent le cycle le plus court et le plus répétitif. Une vulnérabilité est détectée, corrigée, déployée, et le cycle recommence dès que de nouvelles CVE sont publiées sur les librairies utilisées. Ce travail n'est jamais « terminé ». Il est structurellement intégré au rythme de la Squad, au même titre que les cérémonies Scrum.

Les évolutions fonctionnelles, comme la feature cartes virtuelles professionnels, ont un cycle de vie plus long : de la conception au déploiement, en passant par le développement et les différentes phases de tests. Ma contribution au code source restera présente, même si elle sera progressivement amendée par les développeurs qui suivront.

La documentation SharePoint est la contribution la plus transversale dans le temps. C'est une ressource vivante : mise à jour à chaque évolution de l'architecture, elle constitue la mémoire opérationnelle de la Squad. Sa valeur s'accroît avec le temps, à condition qu'elle soit maintenue.

À court terme : les mois qui suivent la soutenance

Deux sujets portés en fin de période nécessitent une reprise directe par l'équipe. Le Spring Batch de purge RGPD doit être finalisé : choix du mode de déclenchement, puis déploiement en production. J'ai rédigé un ticket de suivi détaillant l'état d'avancement et les points ouverts, qui sera traité sur la prochaine période.

Le renouvellement des certificats de chiffrement, piloté jusqu'à l'avant-dernier environnement, devra quant à lui être finalisé en production lors de la période à venir.

En parallèle, j'entamerai vraisemblablement de nouveaux sujets d'ici août, qui seront ensuite repris par l'équipe à mon départ. L'organisation agile et la documentation permettent cette continuité naturelle entre les membres de la Squad, quel que soit le niveau de rotation.

Après août : le devenir des travaux une fois la période terminée

À mon départ fin août, les sujets en cours seront absorbés dans le flux habituel de la Squad. Le Spring Batch, une fois en production, entrera dans un cycle de maintenance autonome : exécutions périodiques supervisées par les outils de monitoring de l'équipe. Son périmètre pourra évoluer si on identifie de nouveaux types de données à purger.

Les évolutions fonctionnelles livrées deviendront des composants ordinaires des applications du crédit agricole : accessibles aux clients et maintenus collectivement.

Sur un horizon plus large, les enjeux qui ont structuré mon quotidien au sein de la Squad s'inscrivent dans des tendances de fond. L’évolution des aspect économiques et concurrentielles vu précédemment se traduiront par de nouveaux tickets sur les API d'enrôlement. La pression réglementaire continuera de s'intensifier avec l'évolution des standards PCI-DSS et la mise en œuvre

de la directive DSP3, renforçant encore le rôle de la gestion des CVE et des certificats dans le quotidien de l'équipe.

À plus long terme, les composants que j'ai développés ou maintenus s'inscrivent dans la modernisation progressive du SI bancaire portée par CA-TS : encapsulation du legacy derrière des APIs Java modernes, migration vers des architectures plus modulaires. Mes contributions n'en sont qu'une brique parmi d'autres mais c'est précisément cette logique de contribution collective et continue que j'emporte comme principal enseignement de cette année.

VI. Bilan

Cette année d'alternance au sein de CA-TS a représenté ma première immersion prolongée dans un environnement professionnel de grande entreprise. Le bilan est positif sur l'ensemble des dimensions attendues.

6.1 Évolution de la posture

	Sept – Nov 2025	Fév – Mars 2026	Avr – Mai 2026	Juin 2026
Posture	Onboarding	Autonomie progressive	Force de proposition	Autonomie bout-en-bout
Signature	Découverte de l'environnement SI, formations, premier ticket	Gestion d'incidents, CVE, librairie de logging	Feature fonctionnelle, Spring Batch, certificats	A venir

6.2 Compétences acquises

Techniques	Sécurité	Méthodologiques	Transversales
Java Back-End, Spring Boot, Spring Batch, Maven, API REST, GitLab CI/CD	Gestion CVE, Security by Design, tests d'intrusion, certificats de chiffrement	Scrum (Daily, Sprint Planning, Review, Rétro), gestion en points d'effort, coordination inter-équipes	Communication professionnelle, documentation technique, conformité RGPD, réglementation bancaire

6.3 Points positifs et difficultés

Les points positifs sont nombreux : une formation technique complète couvrant l'intégralité du cycle de vie d'une application (conception, développement, tests, déploiement, correction en production), une intégration sociale réussie au sein d'une équipe bienveillante, et l'opportunité de contribuer comme n'importe quel développeur de l'équipe, sans que le statut d'alternant ne soit discriminant.

Les principales difficultés tiennent à la complexité intrinsèque d'un SI bancaire de grande entreprise : la densité des environnements techniques, les temps d'attente liés aux processus de validation et de droits d'accès, et la nécessaire montée en compétence à chaque retour après une période école. Ces obstacles ont néanmoins été des opportunités d'apprentissage à part entière.

Sur le plan du projet professionnel, cette année a pleinement confirmé mon choix de poursuivre en cybersécurité à l'ENSIBS de Vannes à la rentrée 2026. L'exposition quotidienne aux enjeux de sécurité applicative, la participation aux debriefs de tests d'intrusion et le rôle des Security Champions dans l'équipe ont ancré cette orientation de manière concrète et vécue, au-delà d'un simple intérêt théorique.

VII. Conclusion

Cette année d'alternance au Crédit Agricole Technologies et Services m'a permis de vivre de l'intérieur ce que signifie être développeur back-end dans un environnement bancaire de production. Loin des projets scolaires à périmètre contrôlé, j'ai été confronté à la réalité d'un système d'information vivant, en évolution permanente, soumis à des exigences de sécurité, de conformité et de disponibilité qui n'admettent pas le compromis.

Les compétences acquises dépassent le seul registre technique. L'année m'a appris à travailler dans un cadre agile réel, à communiquer avec des interlocuteurs variés (développeurs, business analysts, architectes, équipes partenaires, équipes d'exploitation), à gérer des imprévus en production et à documenter mon travail pour ceux qui me suivront. Elle m'a aussi appris une posture : proposer, pas seulement exécuter.

L'environnement économique dans lequel s'inscrit CA-TS illustre les tensions profondes qui traversent le secteur bancaire : la pression des néobanques sur le time-to-market, celle des wallets constructeurs sur les marges, et celle de la réglementation sur les pratiques de données. Ces tensions ne sont pas abstraites : elles se traduisent concrètement en tickets, en sprints et en choix d'architecture dans le quotidien de la Squad.

Je repars de cette première année professionnelle avec la conviction que le chemin vers la cybersécurité que je m'apprête à suivre à l'ENSIBS est le bon. Les bases techniques acquises en développement back-end seront des atouts précieux pour comprendre les systèmes que je serai amené à sécuriser, et l'expérience du monde bancaire m'a fourni un cadre de référence concret pour mesurer les enjeux réels de la protection des données et des systèmes critiques.

> Grégoire Marousé : Alternance du 1er septembre 2025 au 28 août 2026 Crédit Agricole Technologies et Services, Vannes BUT Informatique 2ème année, IUT de Vannes