

Analyse de la compétence acquise au regard des AC/CE

Apprentissages Critiques :

- *AC23.01 — Automatiser l'administration système avec des scripts*

J'ai appris à mettre en place des scripts de démarrage pour le master, les routeurs et les clients, bien que finalement non utilisés, ainsi que des procédures reproductibles d'installation et de déploiement sur plusieurs machines virtuelles. Pour la base de données, j'ai confirmé ma compétence d'automatisation des tâches de création de la base de données, de ses tables et d'un utilisateur associé.

- *AC23.02 — Développer une application à partir d'un cahier des charges donné*

Mon projet a été développé à partir du cahier des charges imposant un routage oignon et l'utilisation d'au moins 3 routeurs pour une communication anonymisée et un chiffrement assymétriques, ainsi qu'une interface graphique utilisateur. Les contraintes fonctionnelles, et techniques ont bien été respectées : communication fonctionnelle, interfaces graphiques pour le master et le client, démonstration de l'anonymisation des échanges, choix des outils d'environnement.

- *AC23.03 — Utiliser un protocole réseau pour programmer une application client/serveur*

J'ai implémenté une application réseau reposant sur des communications TCP client/serveur : le master est l'aiguilleurs, les routeurs sont des intermédiaires de transit, et les clients sont les points d'entrée ou sortie. J'ai appris à maîtriser les bases des mécanismes réseau (sockets, écoute, connexions multiples, forwarding) pour implémenter des échanges structurés, serialisés et passant par plusieurs sauts.

- *AC23.04 — Installer, administrer un système de gestion de données*

J'ai installé et configuré une base de données MariaDB sur une machine dédiée et géré la configuration réseau du serveur de base de données (accès distant avec bind-address), la création des utilisateurs, des tables et des droits. Cela a consolidé mes compétences en gestion de bases de données, déjà souvent effectuées pour d'autres projets depuis le 1^e Semestre.

- *AC23.05 — Accéder à un ensemble de données depuis une application*

Mon application Master accède dynamiquement aux données stockées en base (tables routeurs, logs) et les clients récupèrent indirectement ces données via le master pour construire les circuits. Les interfaces graphiques permettent également de consulter ces informations, ce qui montre l'intégration complète entre application réseau et base de données.

Composantes Essentielles :

- *CE3.01 — Être à l'écoute des besoins du client*

J'ai pris en compte les attentes explicites de l'enseignant (échanges multi-saut anonymisés sur 3 routeurs minimum, environnement multi-VM, interfaces graphiques).

- *CE3.02 — Documenter le travail réalisé*

N'ayant pas l'habitude de développer en général dans mon temps libre, ni beaucoup d'expériences dans ce domaine, j'ai mal évalué le temps conséquent nécessaire à une documentation complète et de qualité pour un projet comme celui-ci. Je me suis focalisée sur le développement fonctionnel de l'application en dépit donc de la documentation trop peu détaillée pour refléter le travail de développement.

Pour ce projet, j'ai réalisé une vidéo de démonstration de l'environnement et des manipulations à effectuer pour tester le routage oignon fonctionnel ainsi qu'un document écrit reprenant les points de la vidéo en détail. Je me suis aussi familiarisée avec la rédaction usuelle d'un README github.

De cette expérience, je retiens la nécessité d'une plus longue fenêtre de temps pour travailler la doc de mes futurs projets de développement.

- *CE3.03 — Utiliser les outils numériques à bon escient*

J'ai utilisé des outils adaptés au contexte : machines virtuelles pour l'environnement de test, GitHub pour la gestion du code et la journalisation du travail, MariaDB pour la persistance des données, PyQt5 pour les interfaces graphiques, ainsi que des outils système standards (SSH, scp) pour la communication entre machine hôte et VM. J'avais aussi utilisé les fonctionnalités VirtualBox de Dossier Partagé et du Presse-papier partagé pour un travail sur VM plus rapide et confortable pendant le développement. La configuration d'un dossier partagé était malheureusement trop source d'erreur pour la faire refaire par un lecteur de ma documentation d'installation.

- *CE3.04 — Choisir les outils de développement adaptés*

Mes choix technologiques ont été faits en fonction des contraintes pédagogiques et techniques : Python pour la rapidité de développement et la lisibilité, sockets TCP pour le réseau, PyQt5 pour les interfaces imposées, MariaDB pour la base de données et PyMySQL pour un connecteur mysql-python léger.

- *CE3.05 — Intégrer les problématiques de sécurité*

La sécurité a été prise en compte à travers l'architecture même du projet : anonymisation des flux par chiffrement en couches (oignon), non-divulgation du chemin complet à un seul nœud, et stockage contrôlé des informations sensibles (clés publiques, logs). Les limites et faiblesses du chiffrement utilisé dans ce contexte pédagogique ont également été identifiées et analysées.

Conclusion sur la SAE avec les + et – et le choix de travailler seule

C'était un projet intense mais très intéressant et instructif tant sur le plan technique que sur les us et coutumes d'un travail de développeur/euse et la gestion d'un projet de cette envergure, bien que minime en comparaison à un développement destiné à une production réelle.

Je suis très satisfaite d'avoir choisi de travailler seule, j'ai obtenu ce que je cherchais, à savoir, une supervision totale du projet et l'acquisition d'expérience dans toutes les compétences attendues : programmation, développement, virtualisation, documentation. Concernant la gestion de projet, je n'avais de compte à rendre à des collègues de travail en cas de modification du planning ou si je souhaitais m'attarder sur l'approfondissement de fonctionnalités, non nécessaires au rendu mais source d'apprentissage global. J'ai pu organiser mon travail comme je le souhaitais. Ici, j'ai procédé étapes par étapes afin de mieux comprendre les différents mécanismes : d'abord un fonctionnement en local en cli, puis déployé sur VM, toujours en cli, et enfin en GUI.

Cela a tout même eu des inconvénients : Le suivi de gestion de projet est moins rigoureux car ressenti moins nécessaire et mon travail repose uniquement sur mes compétences, de base, faibles, dans le domaine du développement.