

Rendu Préparatoire du Projet

SAE 3.02 Développer des
applications communicantes



Sommaire

Table des matières

Présentation du projet.....	2
Nom & description :	2
Mais en quoi conste t-il ?.....	2
Répartition du travail en groupe	2
Constitution du groupe : Emeraldas :	2
Les rôles seront :.....	2
Tâches à réaliser :	3
Fonctionnalités prévues	3
Fonctionnalités essentielles :	3
Fonctionnalités secondaires :	4
Fonctionnalités bonus :.....	4
Planning de développement.....	4
Durée totale du projet et répartition des tâches.....	4
Jalons clés	5
Organisation et outils.....	5
Outils :	5
Réunions :	5
Risques identifiés	6

Présentation du projet

Nom & description :

Le projet se nomme « Conception d'une architecture distribuée avec routage en oignon »

Mais en quoi consiste-t-il ?

Ce projet consiste à concevoir et implémenter un système de communication anonyme basé sur une architecture distribuée et un routage en oignon.

Le système repose sur plusieurs routeurs virtuels, un serveur maître et des clients capables d'échanger des messages de manière anonyme.

L'objectif principal est de garantir la confidentialité et l'anonymat des utilisateurs grâce à un chiffrement asymétrique simplifié et une architecture multi-sauts.

Le public cible est composé des étudiants, des enseignants ou tout autre utilisateur souhaitant analyser, comprendre ou démontrer le fonctionnement de cette architecture.

Répartition du travail en groupe

Constitution du groupe : Emeraldas :

⇒ *Imène MIRA & Chloé BRISSE*

Prenant en compte le fait que nous sommes uniquement deux dans le groupe, nous ne voyons pas la nécessité de désigner un chef de projet.

Pour une meilleure efficacité, nous préférons partager toutes les tâches.
En effet, les tâches sont réparties mais vérifiées par le groupe

Les rôles seront :

- Dev front
- Dev back
- Design
- Test

Tâches à réaliser :

- ⇒ Création et gestion des VMs
 - Tâche partagée
- ⇒ Développement du Master (gestion des clés, topologie, supervision).
 - Tâche partagée
- ⇒ Développement des Routeurs (génération de clés, déchiffrement, transmission).
 - Tâche partagée
- ⇒ Développement du Client (construction des couches d'oignon, interface Qt).
 - Tâche partagée
- ⇒ Mise en place de la base de données MariaDB.
 - Tâche partagée
- ⇒ Gestion du dépôt Git et suivi de projet.
 - Tâche partagée
- ⇒ Conception du planning et documentation (Document de suivi mis à jour tout au long du projet)
 - Tâche partagée

Fonctionnalités prévues

Fonctionnalités essentielles :

- ⇒ Génération et distribution des clés publiques (Master).
- ⇒ Création d'un chemin de routage aléatoire par le client.
- ⇒ Construction du message chiffré en couches successives.
- ⇒ Transmission du message via plusieurs routeurs.
- ⇒ Déchiffrement d'une seule couche par routeur.
- ⇒ Base de données : stockage des clés et tables de routage (MariaDB).
- ⇒ Interface Qt du client et du master.

Fonctionnalités secondaires :

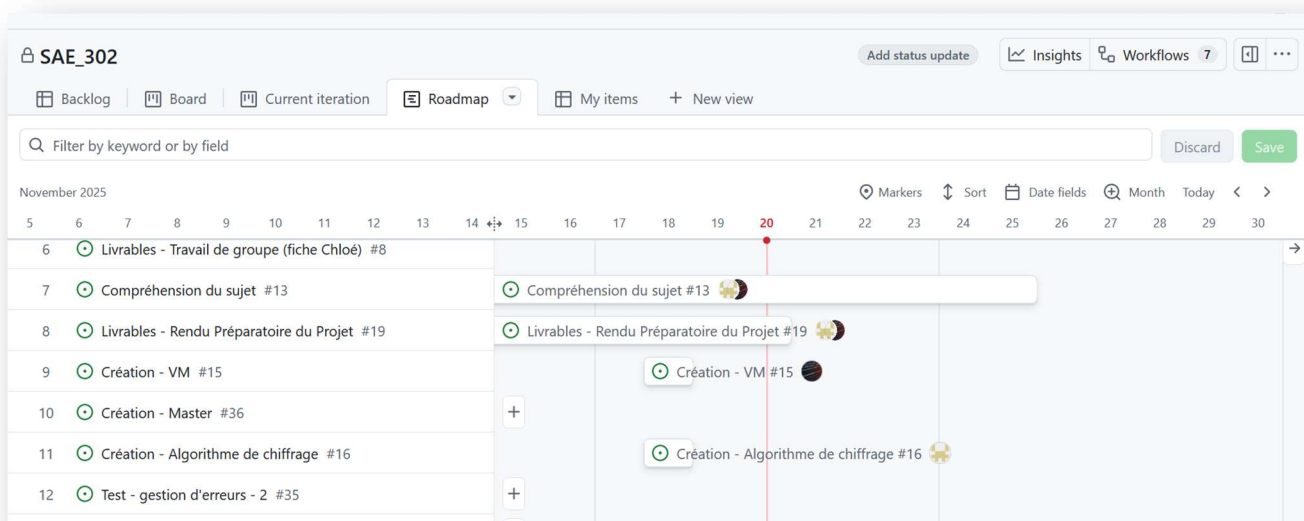
- ⇒ Visualisation graphique de la topologie.
- ⇒ Affichage des logs dans le Master.
- ⇒ Statistiques de trafic anonymisé.

Fonctionnalités bonus :

- ⇒ Démarrage automatisé des routeurs depuis le Master.
- ⇒ Interface avancée avec carte du réseau.

Planning de développement

Durée totale du projet et répartition des tâches



Le Gantt est disponible sur GitHub. Onglet « Project ».

Pour nous, le projet s'étend sur toute la durée disponible, c'est-à-dire jusqu'au 31 décembre. Nous ne voulons pas nous dire inconsciemment "il y a de la marge" et risquer, au final, d'avoir du retard. Nous visons de finir en avance le travail avant le 31 décembre.

Jalons clés

Les tâches sont réparties au fur et à mesure de l'évolution du projet, s'il y a du retard, si l'une de nous a plus d'affinité ou pour toutes autres raisons. Nous préférons être flexibles pour éviter de rester bloqués sur une tâche, il nous est même possible de travailler en même temps sur un élément ou de refaire une répartition des tâches si l'une de nous est dans une impasse. Il s'agit de travailler de concert et de travailler de façon fusionnelle pour rendre le meilleur travail possible.

Organisation et outils

Outils :

Suivi projet : GitHub & GitHub Project,

Base de données : MariaDB,

Communication : Discord,

Gestion de projet : Trello,

Développement : Python, PyQt5/6, Socket, Threading.

Réunions :

- ⇒ Avant chaque cours avec l'enseignant,
- ⇒ Débriefing après chaque fin d'avancement,
- ⇒ Le suivi se fera via GitHub avec des commits fréquents et explicites des différents prompts ou « readme ».

Risques identifiés

- ⇒ **Complexité technique** : gestion correcte des threads et sockets → test progressif,
- ⇒ **Dépendances externes** : MariaDB → prévoir un script d'installation,
- ⇒ **Débogage multi-machines** : variabilité des environnements → documentation précise,
- ⇒ **Temps** : charge importante → planning strict et jalons obligatoires.