

Protection et gestion des licences

Présentation - Gestion de projet

Sami Babigeon, Louka Boivin, Kaci Hammoudi, Alexis Osmont

27 janvier 2022

Master Informatique - 1ère année

1. Présentation du sujet
2. Analyse fonctionnel
3. Solution technique
4. Stratégie qualité
5. Organisation du projet
6. Risques

Présentation du sujet

Protection et gestion des licences

Objectifs :

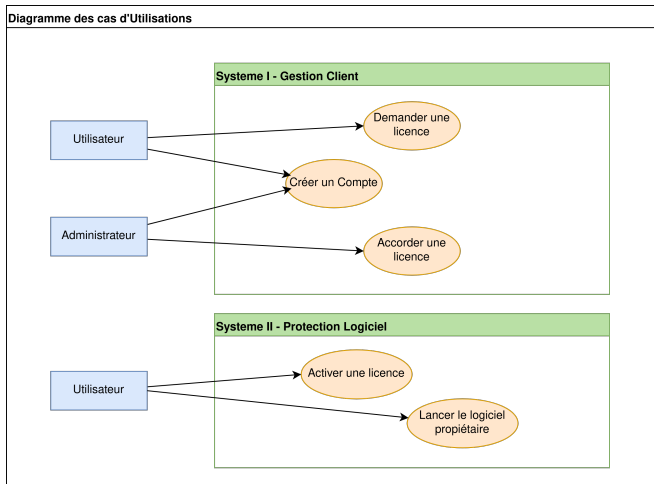
- Génération et vérification des licences
- Plateforme de gestion pour le client, de demande pour les utilisateurs
- Protection des logiciels du client

Analyse fonctionnel

Comment distribuer les logiciels que le client a développé ?

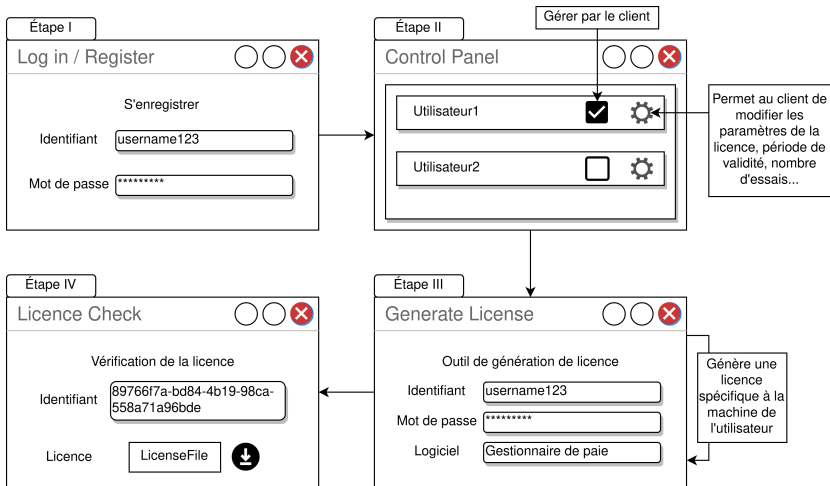
- Créer une licence unique pour une machine et un logiciel donné
- Une plateforme commune au client et aux utilisateurs
- Une librairie lui permettant d'intégrer la vérification à ses logiciels ou un outil permettant de greffer automatiquement

Cas d'utilisations



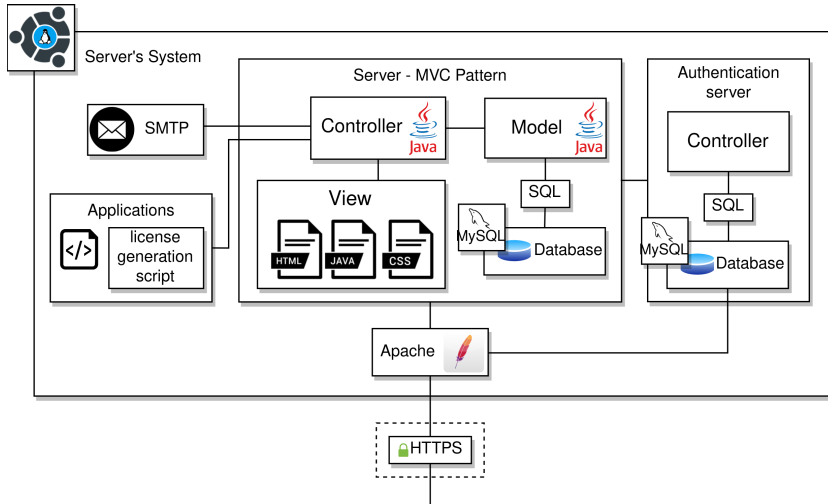
- Sécurité
- Compatibilité Windows
- Maintenabilité

Fonctionnement globale

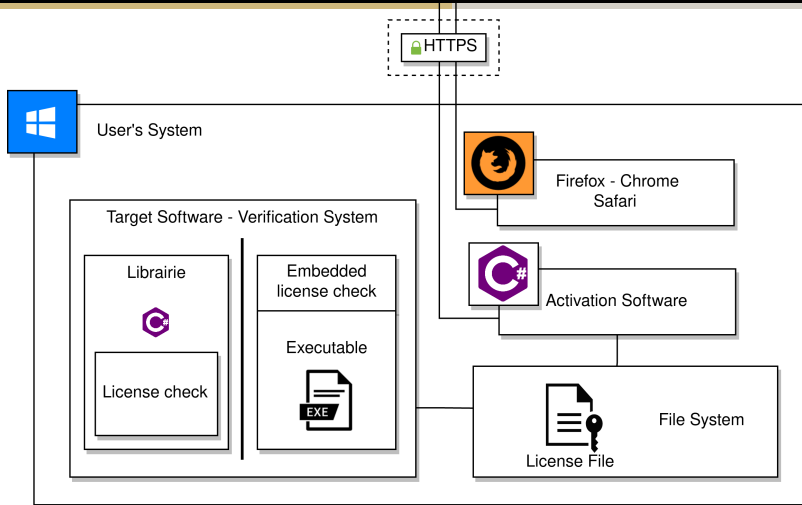


Solution technique

Architecture logicielle - Serveur



Architecture logicielle - Client



Plateforme sûre :

- Chiffrement (HTTPS)
- Serveur d'authentification (isolement, configuration)
- Hashage des mots de passe (PBKDF)
- Sécurisation des bases de données (configuration, requêtes préparées)
- API Rest (utilisation de tokens d'authentification)

Génération et vérification de licence :

- Signature (El Gamal)
- Obfuscation

Stratégie qualité

Actions

- Gestion et journalisation des bugs
- Serveur de tests
- Tests unitaires sur le modèle
- Tests d'intégration
- Tests d'intrusion

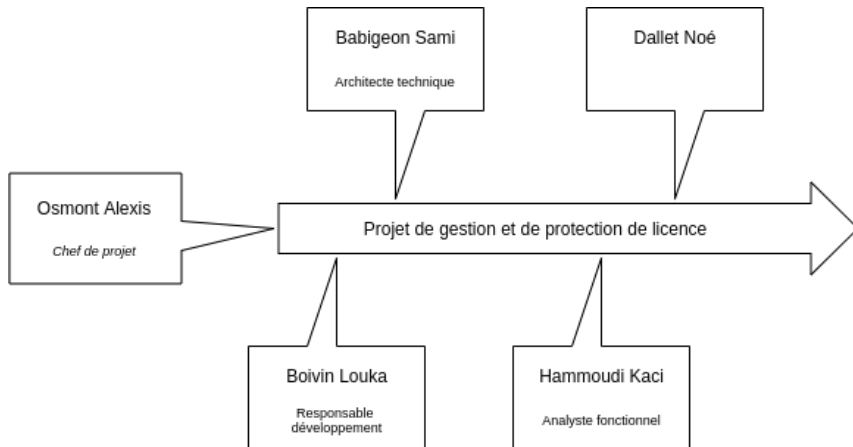
Outils

- MantisBT
- Docker
- Jupiter
- À définir
- Scripts automatiques

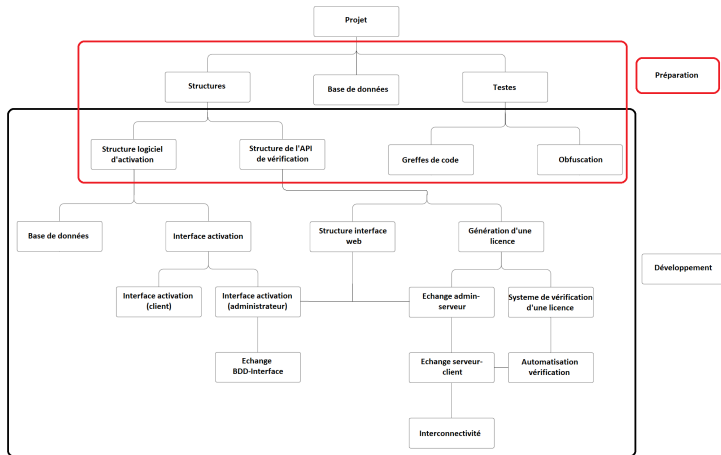
- Définition d'un ordre de priorité sur les erreurs
- Rapport concernant les problèmes/faillles généré régulièrement
- Charte de code et documentation pour un suivi et une maintenabilité du projet

Organisation du projet

Affectation des rôles



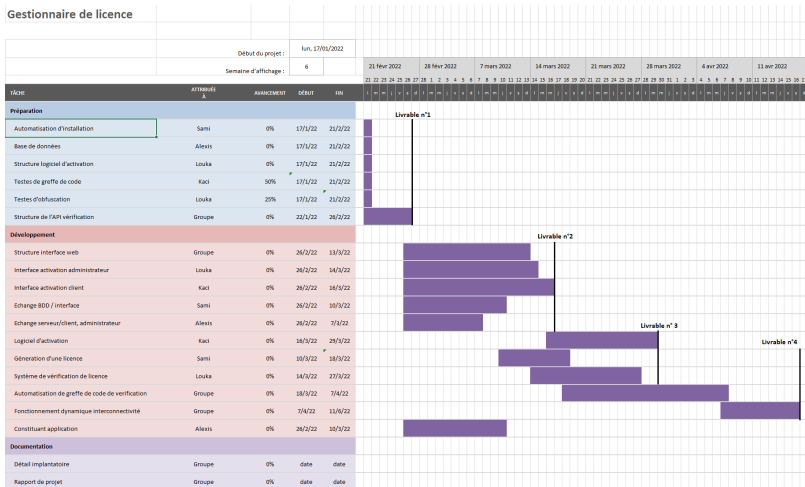
Découpage des tâches



Préparation

Développement

Diagramme de Gantt



Risques

Risques principaux

Risque	Craintes	Effets	Impacts	Prob.	Stratégie
Faibles d'implémentation	Cracking du logiciel et de son injection	Perte de contrôle du logiciel et Insatisfaction du client	4	4	Effectuer des tests unitaires sur les logiciels et les exécutables.
Échec de l'injection de code	Que la complexité de la tâche soit trop élevée et que l'objectif ne soit pas atteint	Insatisfaction client / Perte de temps et d'énergie	2	3	Effectuer une preuve de concept avant le début de la phase de développement.
Signature ElGamal	Complications lors de l'accès au logiciel	Impossible d'utiliser le logiciel / Objectif injection de code non atteint	3	2	Effectuer une preuve de concept au plus tôt.
Obfuscation du code	Cracking du logiciel et de son injection	Perte de contrôle du logiciel d'injection de code	4	2	Effectuer des tests unitaire sur les logiciels et les exécutables.
Respect des échéances	Manque de temps pour les livrables de fin de sprint	Prise de retard par rapport au échéances de développement	3	1	S'organiser et séparer efficacement les tâches pour bien gérer le travail.

Merci de votre écoute

Questions ?