# Protection et gestion des licences

Présentation - Gestion de projet

Sami Babigeon, Louka Boivin, Kaci Hammoudi, Alexis Osmont

28 janvier 2022

Master Informatique - 1ère année



### Sommaire

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



# Présentation du sujet



### Intitulé

### Protection et gestion des licences

#### Client:

• M. Ziadi

### Objectifs:

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



# **Analyse fonctionnel**



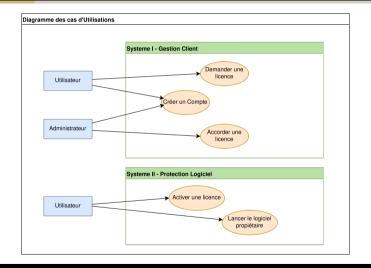
#### Besoin du client

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 libraire lui permettant d'intégrer la vérification à ses logiciels ou un outil permettant de greffer automatiquement



### Cas d'utilisations





## Exigences

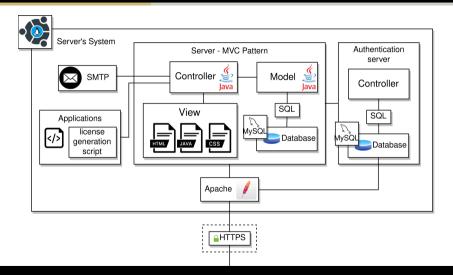
- Sécurité
- Compatibilité Windows
- Maintenabilité



# Solution technique

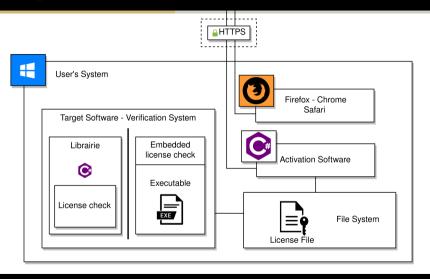


### Architecture logicielle - Serveur





### Architecture logicielle - Client





### Solutions techniques

#### 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



### Stratégie adoptée

#### **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



### Vérification des exigences

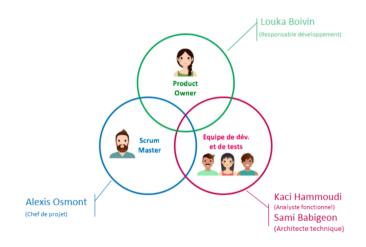
- Définition d'un ordre de priorité sur les erreurs
- Rapport concernant les problèmes/failles généré régulièrement
- Charte de code et documentation pour un suivi et une maintenabilité du projet



# Organisation du projet

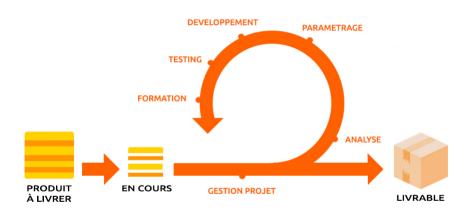


### Affectation des rôles



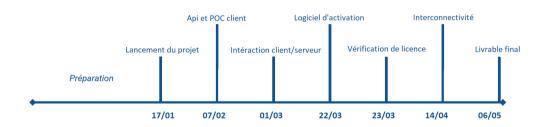


## Méthode Agile



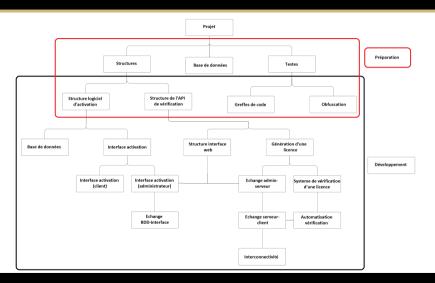


### Livrables



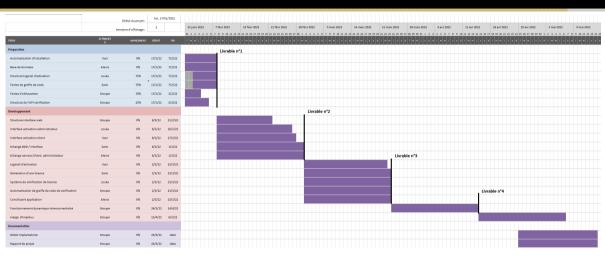


### Découpage des tâches





# Diagramme de Gantt





# Risques



# Risques principaux

Risque	Craintes	Effets	Impacts	Prob.	Stratégie
Failles d'implé- mentation	Cracking du logiciel et de son injection	Perte de controle du logicel et In- satisfaction du client	4	4	Effectuer des tests uni- taires sur les logiciels et les exécutables.
Échec de l'injection de code	Que la com- plexité de la tâche soit trop élevée et que l'objecif 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 dévelop- pement.
Signature ElGamal	Complications lors de l'accès au logicel	Impossible d'utliser le logicel / Ob- jectif 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 controle du logicel d'in- jection de code	4	2	Effectuer des tests uni- taire 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 tra- vail.



# Merci de votre écoute

Questions?

