

# **Journal d'activité octobre 2021 janvier 2022**

## **Projets**

- Correction du tentacule Picklog de l'outil de test appelé le Kraken
- Correction de l'activité VerifyCheckSignature du tentacule C21Whitebox du kraken
- Correction de bugs graphiques sur l'interface web Parcel Tracker 5 qui permet de suivre les colis dans le système d'information

## **Colis 21**

Colis 21 est un système d'information. Il s'agit d'un ensemble de programmes qui communiquent entre eux pour assurer une représentation informatique des réels colis en circulation afin de les tracer, les suivre et d'effectuer des actions en cas d'anomalies détectés.

## **Le kraken**

Le kraken est un programme qui sert à tester Colis 21 en simulant des colis dans le système d'information

## **Parcel Tracker 5**

Parcel Tracker 5 est la 5e version de l'outil Parcel Tracker. Parcel Tracker est une interface web qui permet de suivre l'évolution des colis. Celle-ci est faite en React et se base sur une web service en .Net.

## **Périodes**

### **Correction de l'activité VerifyCheckSignature du tentacule C21Whitebox du Kraken**

#### **Objectifs et missions poursuivies**

- Comprendre le fonctionnement du tentacule.
- Comprendre l'API de Colis21
- Comprendre le fonctionnement du moteur
- Corriger l'activité du tentacule

#### **Action concrètes (1 à 3 actions par objectif, 5 à 6 lignes par action)**

- Reproduction du bug sur l'environnement test-v sur lequel travaille les testeurs puis en local en configurant mon environnement de façon adéquate. Avant même de me lancer dans la correction du bug j'essaie tout d'abord reproduire le bug dans l'environnement de test appelé test-v sur lequel le

bug a tout d'abord été détecté. Pour cela j'ai besoin de me connecter en vpn au réseau de l'entreprise afin d'avoir accès aux différents environnements sur lesquels le kraken est déployé.

- Une fois le bug reproduit sur l'environnement de test j'essaie de le reproduire en local en configurant mon environnement local avec une nouvelle branche git basée sur la même branche utilisée pour l'environnement test-v et je mets à jour ma base de données mongodb par rapport à la base de données utilisée sur test-v.
- Il me faut ensuite analyser quelles sont les tentacules nécessaires à lancer pour pouvoir lancer mon cas de tests sans avoir. L'idéal est de lancer uniquement les tentacules nécessaires au cas de test choisi et pas les autres car cela aurait pour effet de ralentir mon poste en local.

## **Difficultés rencontrées et solutions trouvées (environ 1 page)**

### **Analyse**

**Difficulté** Afin de pouvoir me lancer dans la correction du bug j'avais tout d'abord besoin de bien comprendre le fonctionnement de l'application en elle même. Comme plusieurs services faisant partie du système d'information partagent le même code source en commun, celui-ci est très fourni et il peut être assez difficile d'investiguer quels sont les zones pertinentes dans la résolution du problème. Pour cela j'ai effectué des recherches dans le block note en ligne que partage l'équipe sur One Note pour rechercher des informations concernant le projet sous un aspect technique mais aussi fonctionnel.

**Solution** Longues périodes d'analyse du code source et du block note contenant des informations sur la partie fonctionnelle et technique du projet

### **Compatibilité de versions**

**Difficulté** Compatibilité des branches git de développement avec les versions de la base de données dans mongodb

**Solution** Mise en place d'un script facilitant la mise à jour des bases de données en local grâce à des fonctions bash

### **Échec de comparaison**

**Difficulté** Erreur dans la comparaison de signatures sous format svg à 1 pixel près

**Solution** Sollicitation du Tech Lead pour m'orienter dans la recherche de la cause du problème

### **Compétences acquises (hard et soft skills) (environ 1 page)**

- Montée en compétences sur RabbitMQ et sur MongoDB

### **Travail sur Parcel Tracker 5**

#### **Objectifs et missions poursuivies**

- Comprendre l'organisation du code source du projet React -> Lecture du code de l'application React et expérimentation sur de nouvelles branches git pour étudier le fonctionnement des composants
- Comprendre la hiérarchie des composants React dans l'application -> Installation du plugin React-Dev-Tools sur firefox pour afficher la hiérarchie des composants dans la console de firefox
- Trouver leur définition dans le code source -> Recherche des occurrences des noms de composants en utilisant grep

#### **Action concrètes (1 à 3 actions par objectif, 5 à 6 lignes par action)**

- Déploiement de l'outil Parcel Tracker en local pour reproduire le bug en question
- 
- Action 3

#### **Difficultés rencontrées et solutions trouvées (environ 1 page)**

- Compréhension d'un système très large et complexe pas toujours très bien documenté.

### **Compétences acquises (hard et soft skills) (environ 1 page)**

- Montée en compétences sur React

## **Glossaire**

- Système d'information: Un système d'information est un ensemble de services utilisant des technologies adaptées pour interagir entre eux afin de collecter, traiter et stocker des données.
- Environnement (Informatique): Un environnement est une machine permettant de déployer une copie du système d'information afin de le développer et tester ses services.
- Terminal : Application de bureau permettant d'utiliser les programmes d'une machine en tapant une commande sous forme textuelle
- Bash : Langage permettant d'exécuter des commandes pour exécuter des programmes

- React: Framework front-end permettant de rendre des applications web plus dynamiques
- Framework: Ensemble de règles et de bibliothèques qui définissent l'organisation d'un code source pour un type d'application prédéfini.
- Composant: Terme propre à React qui correspond à une abstraction permettant de simplifier l'organisation du code source