UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

PROJECT PRIME SPACE DISTRIBUTED SCRAPING

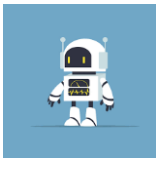
by Streaming 4 Good

Par Oualid GUENNIF, Grasaan GRATION et Nasser HARTI.
L3 MIAGE Classique

MIAGE_

Sous la direction de Nicolas HERBAUT et de Floriane OWCZAREK.

SOMMAIRE



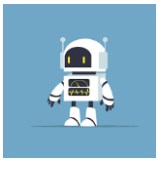
REMERCIEMENTS

En premier lieu, nous tenons à remercier Monsieur Nicolas Herbaut, Monsieur Daniel Diaz et Madame Floriane Owczarek d'avoir permis à notre équipe de participer et d'apporter sa contribution au projet Prime-Space vous avez su nous guider et nous transmettre vos expériences et savoirs en répondant à nos questions .

Un très grand merci à Monsieur Benjamin Guillier qui nous a fait passer un cap en déploiement et intégration continue grâce à son atelier DevOps.

Nous remercions également l'équipe OceanBox actuellement en M1 composée de Abdel Benamara, Julien Doujet, David Ekchajzer et Mathieu Ridet qui nous ont éclairés dès le départ du projet avec leurs conseils.

Nous souhaitons remercier l'ensemble des étudiants de la Miage de PARIS 1 Panthéon-Sorbonne qui participent à ce méta-projet.



INTRODUCTION

Les statistiques nous permettent de mieux comprendre le monde qui nous entoure et notamment sa manière de fonctionner. Dans ce sens, la Miage Paris 1 a décidé d'initier une collaboration avec l'école des médias et du numérique de la Sorbonne (EMNS) et la chaire PcEN et de nous proposer l'opportunité de participer à travers un projet commun L3/M1/M2 à la réalisation d'un objectif, celui d'apprendre davantage sur l'utilisation des plateformes de streaming vidéo en mettant en place la structure informatique et technologique nécessaire.

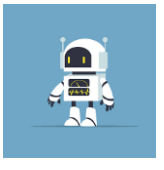
En 2018, les eurodéputés réunis à Strasbourg ont adopté une législation qui s'applique à tous les diffuseurs de contenus dont les plateformes telles que Netflix, YouTube ou Facebook, ainsi qu'à la retransmission en direct sur les plateformes de partage de vidéos. L'une des règles est la fixation à 30% du quota de contenu européen dans les catalogues des plateformes de vidéo à la demande (VOD).

Mais comment mesurer la diffusion de contenu sur une plateforme ou le choix de visionnage appartient à l'utilisateur ? C'est notamment autour de ce postulat que le chercheur Grégoire Bideau et son équipe développe l'idée d'un PRIME-SPACE équivalent au PRIME-TIME de la télévision.

"Nous voulons identifier les mécanismes qui orientent la visibilité des contenus sur Netflix. L'objectif général est d'évaluer le rôle économique de la mise en avant dans le secteur audiovisuel délinéarisé."

L'objectif de ce projet est de permettre à cette équipe de chercheurs de procéder à l'extraction automatique et décentralisée de données issues de plateformes de streaming comme Netflix et YouTube. Notre équipe intervient dans la création d'un environnement de laboratoire contrôlé. En effet, la plateforme actuelle récolte des données qui proviennent des utilisateurs de l'extension chrome. L'idée pour nous est de mettre en place des robots de scraping de données se basant sur des "persona" à leur création, ils seront déployés sur des systèmes embarqués de type Raspberry PI afin de fournir aux chercheurs une source de données complémentaire aux données humaines.

En collaboration avec nos collègues, les Raspberry PI contiendront les programmes et les extensions nécessaires à l'extraction des données. Celle-ci sera mise en place notamment avec des scripts Selenium (exécutés en parallèle) chargés d'automatiser l'expérience utilisateur sur ces plateformes. Par ailleurs, la collecte de ces données sera aussi écrite sur les systèmes via un Dashboard local ergonomique regroupant toutes les informations que chaque appareil aura acquis. UpSwift est le Dashboard de contrôle qui permet de gérer la flotte de Raspberry à distance pour les mises à jours et la gestion d'incident.



1 – ANALYSE

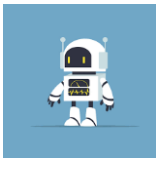
Dans cette partie nous verrons le déroulement et les conclusions des analyses concernant le projet.

1.1 – ANALYSE DU TERRAIN

Les outils que nous développerons lors de ce méta projet L3/M1/M2 en partant de l'existant serviront comme stipuler sur la page de présentation de la chaire PcEN à analyser la diversité et la qualité des contenus culturels et médiatiques mais aussi le traitement des données dans le respect de la vie privée et du débat démocratique. La priorité est donnée actuellement à la construction d'un observatoire statistique afin de produire des études et analyses de marché et de fournir des outils d'aide à la décision pour les opérateurs privés et publics.

La manière de procéder pour avoir des données statistiques, provenant des applications SVOD (en particulier Netflix dans notre cas) avant d'être automatisés par ce projet, était de faire appel à des acteurs humains qui utilisaient ces services. Ainsi, la résultante était des statistiques reposant sur leurs actions. Le talon d'Achille de cette méthode est qu'elle limite grandement le nombre d'utilisateurs à part bien évidemment si l'on en emploie une multitude en parallèle. Par ailleurs, pour détenir des données correctement utilisables il faut d'une part en détenir une quantité importante et d'autre part couvrir presque tous les sujets qu'offre la plateforme pour mieux comprendre la manière dont elle se comporte. Pour finir, il est compliqué voire impossible de posséder des données sur l'application liée à la position géographique en utilisant un vpn. Effectivement, la sécurité de Netflix restreint énormément les possibles contournements que l'on peut faire pour consommer ce genre de contenu à partir d'une localisation différente à la nôtre.

De cette manière, notre projet se veut essentiel. En effet, il permettrait de réduire grandement la durée de traitement : là où les utilisateurs mettraient 3 semaines pour un certain nombre de données , il n'en mettrait au plus qu'une puisqu'il peut tourner en continu.



1.2 – LA CONDUITE DE PROJET

1.2.1 – Gestion du temps

Afin de réaliser au mieux le projet, nous avons listé les différentes tâches puis nous les avons ordonnées. C'est-à-dire voir quelles tâches étaient co-dépendantes. Pour rentrer dans les délais, qui était de 4 mois, nous avons réalisé un diagramme de Pert qui nous a permis de voir combien de temps il nous fallait pour chaque tâche. Ce qui nous a permis de voir le chemin critique, c'est-à-dire les différentes tâches sur lesquelles nous ne pouvons pas prendre de retard au risque de prendre du retard sur le projet dans sa globalité.
Un des défis majeurs ici était de pouvoir trouver le temps de réaliser les différentes tâches du projet en parallèle à plusieurs autres projets.

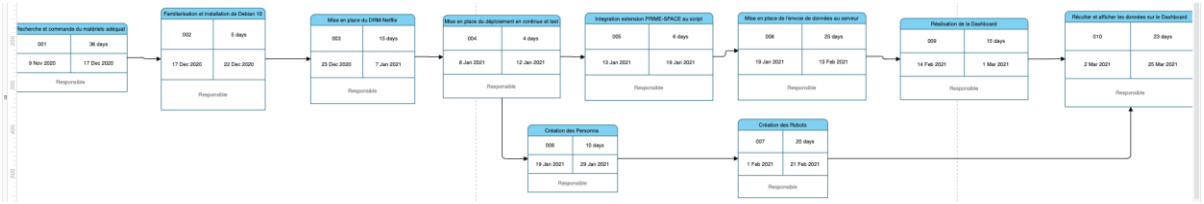


Figure 1 : Diagramme PERT

Une fois le temps trouvé pour chaque tâche, il a fallu faire un diagramme de Gantt. Ce diagramme permet de savoir dans quel ordre réaliser les tâches mais également celles qui peuvent être réaliser en parallèle d'autres.

C'est de ce fait que nous avons pu faire quatre tâches en parallèle au début du projet. Durant le déroulement nous avons remarqué que ce ne sont pas les tâches qui nous prenaient le plus de temps, mais la recherche. En effet, la difficulté de notre projet résidait dans la méconnaissance des différents logiciels comme Docker, l'architecture Arm et la distribution Debian ou encore le développement d'une API. De plus, certaines tâches sont le fruit de nos recherches effectuées durant le projet.

C'est pourquoi, nous avions planifié des tâches assez générales avec des dates très approximatives pour prendre en compte la période de recherche des différentes solutions ainsi que les possibles erreurs qu'on pourrait rencontrer. Cependant, ce qui nous avait pris par surprise était la gestion du temps dès le départ et tout au long de ces trois mois. En effet, nous ne nous attendions pas à participer à autant de projets inspirant au cours de ce semestre.

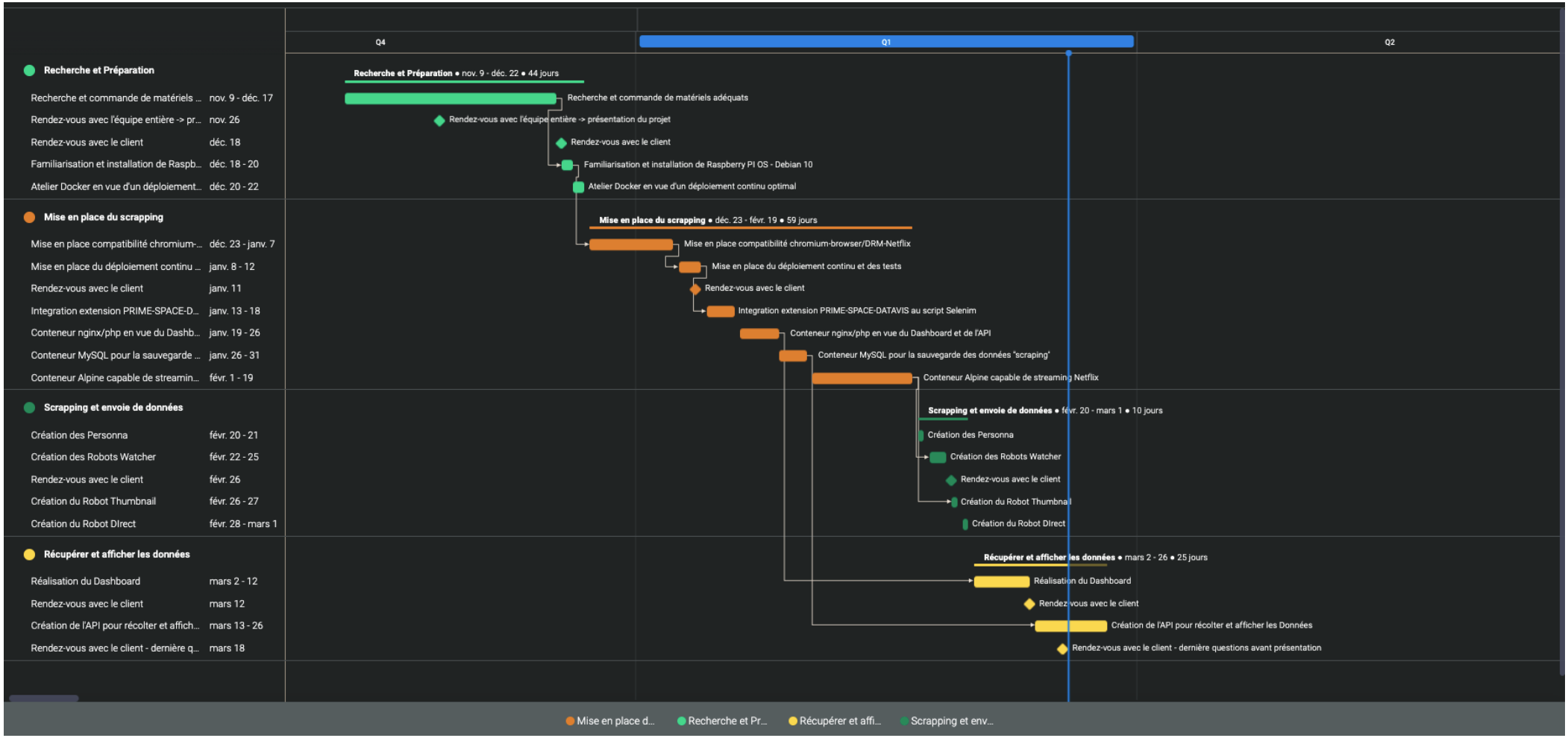
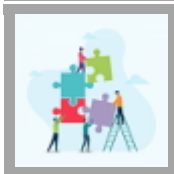
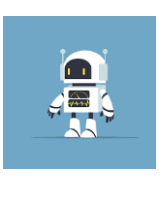


Figure 2 : Diagramme GANTT

	slack	zoom		
 **Avec le client**	- Communication rapide - Échange de fichiers -Echange officiel entre les chercheurs et les doctorants de l'ENMS et la MIAGE	- Jalons		
 **Avec l'équipe complète (L3 app, M1, M2)**		- Réunion générale	- Échange d'information rapide entre les différentes équipes.	- Échange d'information rapide entre les membres de notre équipe
 **Au sein de l'équipe Scraping Netflix**			- Travaille en équipe - Tenir informé l'avancement de chacun	- Idée spontanée à partager - Communication rapide - Partage de fichier utile



1.2.2 – Outil de Communication

La communication au sein de l'équipe est un des éléments fondamentaux pour la bonne réalisation d'un projet. Cette importance a été encore plus soulignée par le contexte unique provoqué par la pandémie. En effet, nous avions plusieurs logiciels pour pouvoir communiquer au sein de l'équipe mais aussi avec notre client Mr. Herbaut.

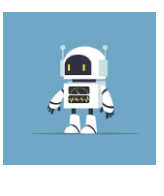
Pour ce faire nous avions utilisé Slack et Discord pour pouvoir communiquer rapidement avec notre client ainsi que l'ensemble de l'équipe (L3 app, M1, M2). Le logiciel Zoom nous a beaucoup servis lors des jalons afin de tenir le client au courant de notre avancée dans le projet et lui poser des questions pour éviter l'effet tunnel.

Quant au sein de l'équipe, nous partagions essentiellement sur WhatsApp les idées spontanées ainsi que certains fichiers utile grâce à sa facilité d'utilisation sur le Raspberry. Enfin, nous travaillions souvent ensemble sur Discord afin de créer un environnement de travail propice à l'entraide et la productivité.

1.3 – LA QUALITÉ ATTENDUE

Le système doit être facilement maintenable. En effet, il est prévu que les nano-ordinateurs soient envoyés à des laboratoires informatiques. De ce fait, s'il arrivait un problème il faudrait pouvoir rapidement le résoudre à distance. D'un autre côté, c'est également important qu'il est cette caractéristique dans le sens où si nous voulions mettre à jour le système il serait appréciable de le faire facilement.

Effectivement notre projet se veut évolutif. C'est en conséquence qu'il faut prévoir une implémentation qui permet cela. Ainsi nous utiliserons notamment les principes "SOLID". De cette manière, même si nous ne sommes plus sur le projet cela pourrait permettre aux futurs équipes de travailler dessus aisément.



1.4 – LE SYSTÈME : VUE FONCTIONNELLE

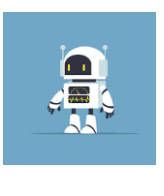
Pour expliquer notre système nous allons le diviser en trois diagrammes d'utilisations. Le premier du point de vue du Dashboard du chercheur. Un autre du point de vue du Dashboard du développeur et le dernier qui concerne le fonctionnement du robot.

1.4.1 - Fonctionnement du Dashboard utilisateur

L'utilisateur du Dashboard locale doit pouvoir accéder et extraire des données provenant de Netflix.

Pour cela il a pour le moment trois robots Netflix. Le premier est le robot "thumbnails" qui consiste notamment en l'extraction de vignettes Netflix. Le second est le robot "direct" qui a pour rôle de récolter les données du direct de Netflix. Enfin, le robot watcher simule un scénario utilisateur et récolte les données associés.

Nous représentons cela tout de suite sous forme du diagramme de cas d'utilisation.



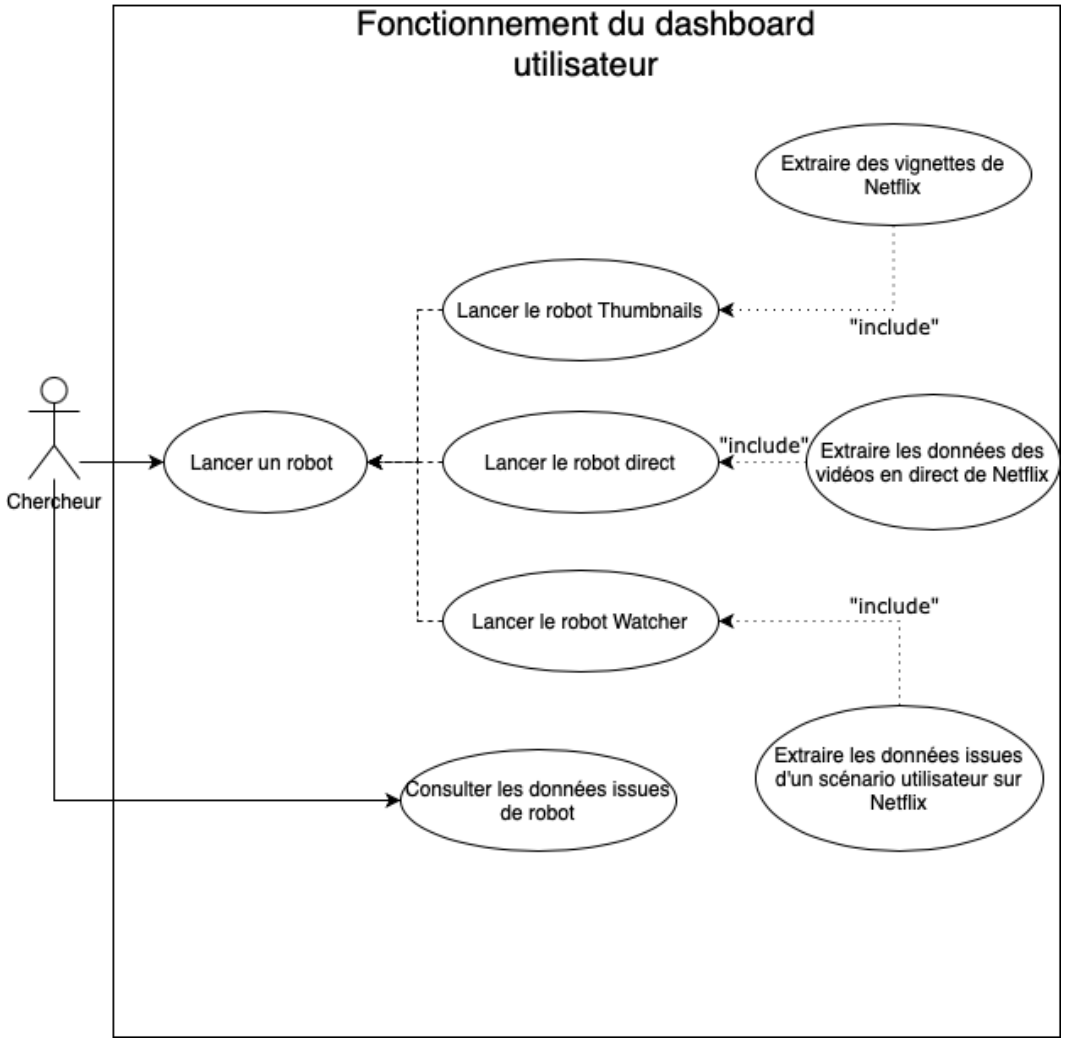


Figure 3 : Diagramme de cas d'utilisation : Point de vue utilisateur

Les chercheurs, qui seront les principaux utilisateurs de notre projet, vont pouvoir lancer le ou les robots qu'ils veulent pour récolter les données et les analysées par la suite.

Tout cela sera possible grâce au Dashboard directement connecté au différents robots. La facilité d'utilisation du système par tout type de personne sera aussi important que le bon affichage de données sur celui-ci.
En effet, le vrai challenge ici sera de pouvoir afficher les données de façon lisible et claire avec une interface épurée.

Le chercheur pourra ainsi lancer le ou les robots selon son type : watcher, direct ou thumbnails. De plus, les robots watcher sont programmés pour sélectionnés un différent persona/profil à chaque lancement !

Un persona, dans notre cas, est un personnage fictif qui représente les goûts d'un groupe de personnes. Par exemple, nous pouvons créer un robot qui va regarder beaucoup de contenue pour enfant afin de simuler au mieux

le comportement d'un enfant. Cela nous permettra de collecter les recommandations que Netflix fera dans le cas d'un enfant afin d'être analysé par le chercheur.

1.4.2 - Fonctionnement du Dashboard développeur

Le développeur doit pouvoir accéder à n'importe quel moment à l'état du Raspberry. Il doit pouvoir connaître complètement son état comme si c'était un appareil qu'il a sous la main.

Nous pouvons le représenter par le diagramme de cas d'utilisation ci-dessous.

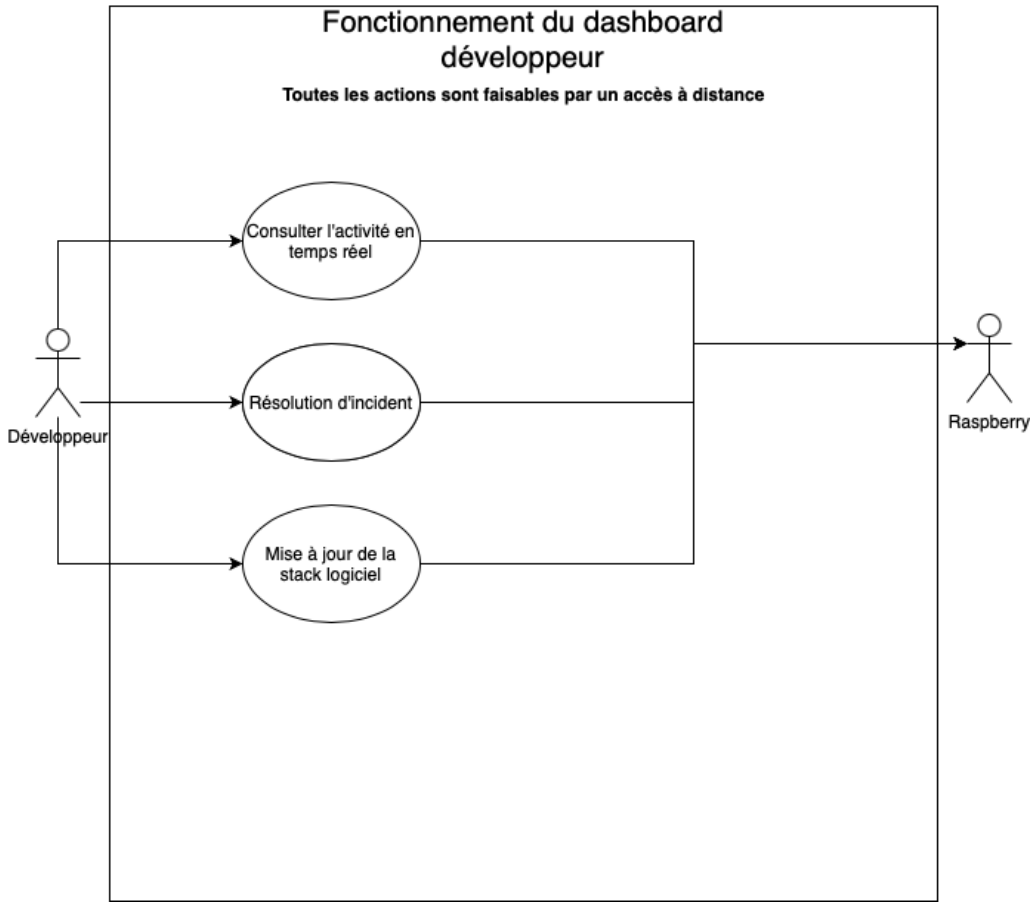


Figure 4 : Diagramme de cas d'utilisation : Point de vue développeur

Le Dashboard UpSwift sera le centre de commande du développeur afin que celui-ci puisse gérer les possibles incidents et vérifier l'état des Raspberry lâchés dans la nature. De plus, les mise à jour récurrentes des robots ainsi que l'ajout de certaines fonctionnalités sera faisable facilement, sans interruptions et à distance grâce à la flexibilité des scripts de chaque robots.

1.4.3 - Fonctionnement des robots

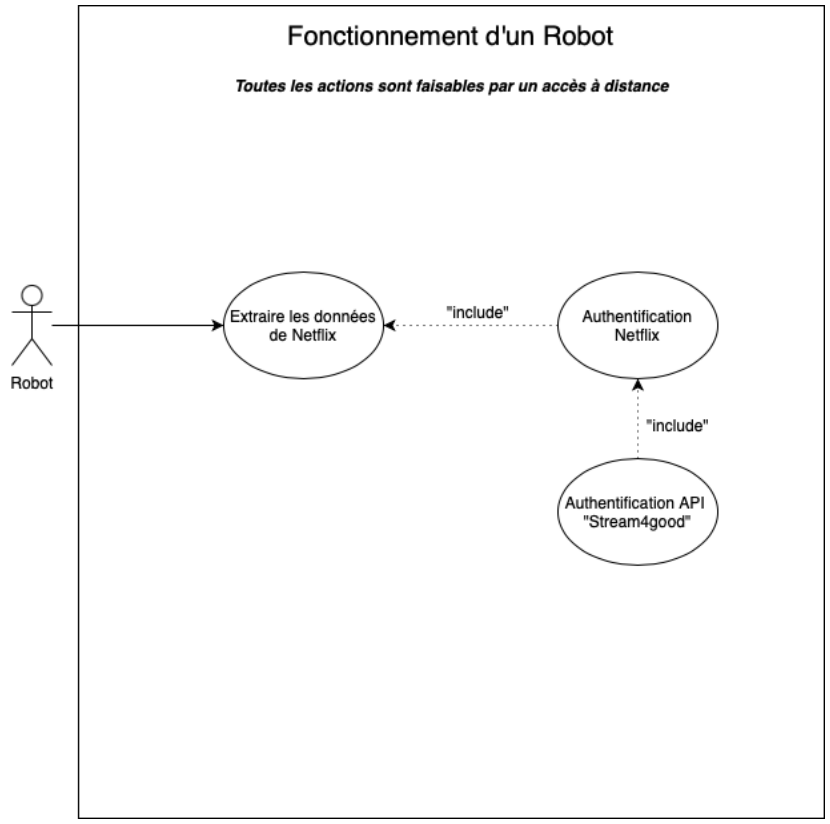
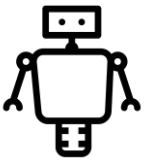


Figure 5 : Diagramme de cas d'utilisation : Point de vue robot

Les robots sont les éléments fondamentaux de notre projet. Leurs fonctionnalités sont assez moindre. En effet, lorsque l'utilisateur va lancer le robot, celui-ci va s'authentifier à l'API Stream4good. Enfin, le robot en question va effectuer sa tâche.



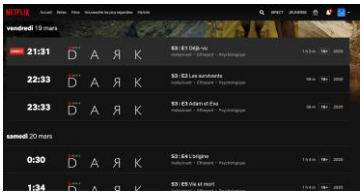
Regarde des films/série selon son persona

Robot Watcher

Récolte les films recommandés par Netflix

Robot Thumbnail

Récolte le planning du Netflix Direct



Robot Direct

Tandis que le robot Direct fonctionne indépendamment, les robots watcher et thumbnail se complètent. En effet, thumbnail s'occupe tout simplement de récolter les films recommandés par Netflix suite au visionnage des films/séries par le robot watcher.
Ce dernier, base son visionnage sur son persona afin d'imiter au mieux des comportements humains.

1.5 – LE SYSTÈME : VUE STATIQUE

Nous pouvons représenter notre système via une vue statique présentant différentes couches pour différents niveaux d'usages et d'autorisations. Le voici ci-dessous

.

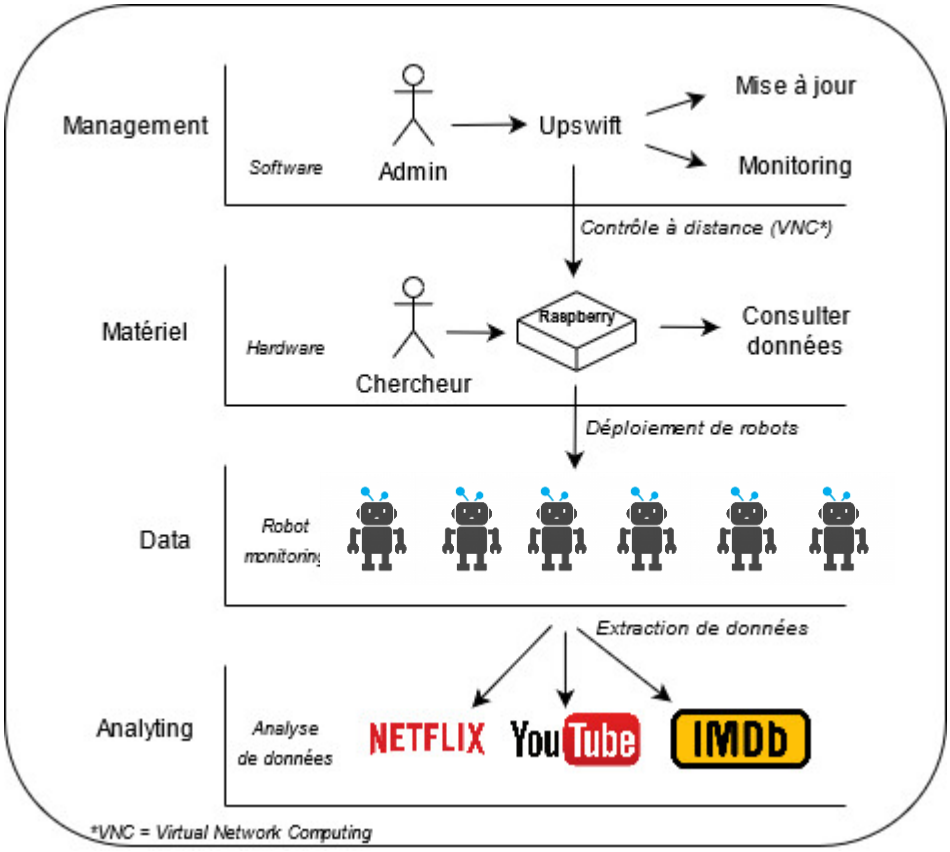
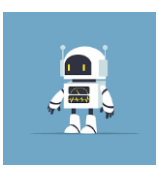


Figure 6 : Schéma d'architecture du fonctionnement du scraping

La couche la plus haute est celle qui permet le plus de possibilités. Effectivement, elle permet de gérer tout le système. La seconde quant à elle représente les Raspberry PI détenus par les chercheurs. Ce sont ces objets qui vont permettre de déployer les robots que l'on retrouve à l'étape suivante. Ceux-là vont se charger de récolter les données et les faire remonter dans le sens inverse de la description des différentes couches qui vient d'être fourni.



1.6 – LE SYSTÈME : VUE DYNAMIQUE

Un développeur a accès aux mêmes données qu'un chercheur car en soit il a plus de droits ou tout du moins il sait mieux les utiliser. Ce que nous recherchons avant tout dans ce projet est d'extraire les données provenant d'applications web de streaming telle que Netflix. Pour se faire on va voir comment on a pensé cela via un diagramme de cas d'utilisation plus orienté chercheur où l'on comprend également son interaction avec le robot via notre système. Par ailleurs, dans un second temps nous verrons comment l'ordinateur dialogue avec le service de streaming pour récolter les données.

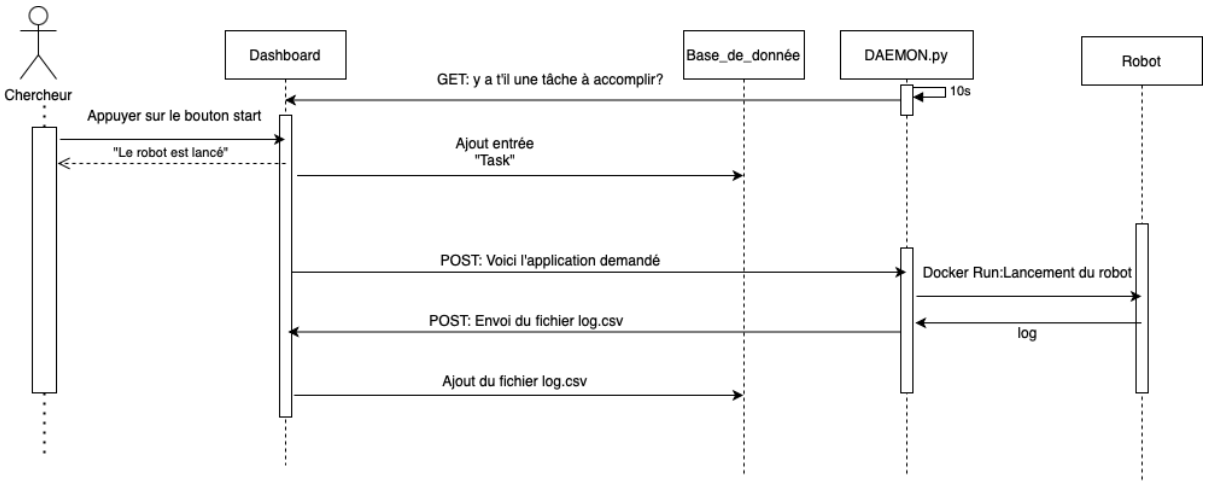


Figure 7 : Diagramme de séquence du lancement d'un robot

Comme nous l'avons vu précédemment avec la figure 3, les chercheurs peuvent lancer le ou les robots de leurs choix via le Dashboard. Le fonctionnement de celui-ci est en partie grâce au script DAEMON.py qui relie le Dashboard aux robots. En effet, celui-ci vérifie tous les 10 sec si une tâche a été lancé et seulement dans ce cas il exécute la commande docker run du robot concerné.
Celui-ci envoie les données au DAEMON.py qui sont directement remontées au Dashboard à travers un fichier csv.
Enfin, ce fichier est stocké dans la base de données pour avoir un historique des lancements.

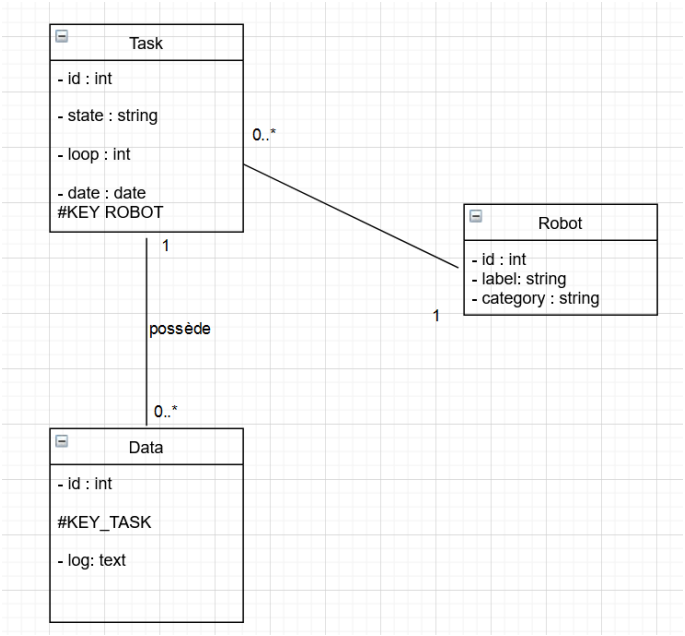


Figure 8 : schéma de la base de donnée

Comme nous l'avions mentionné, le Dashboard crée une tâche dans la base de données lorsque l'utilisateur veut lancer un robot. Ce même Dashboard est interrogé par le DAEMON.py tous les 10 sec pour qu'il puisse savoir s' il y'a une tâche dont l'état (state) est "en attente" dans la base de donnée.

Enfin, lorsqu'une tâche est complétée, le DAEMON.py envoie un fichier csv contenant les données au Dashboard afin que celui-ci puisse le stocker dans la BD. Chaque fichier est associé à une tâche, mais aussi par une catégorie pour savoir si les données proviennent de Netflix, Youtube ou IMDB.

Pour complémenter cette analyse, nous allons déterminer la manière dont le robot va utiliser la plateforme de vidéos en ligne.

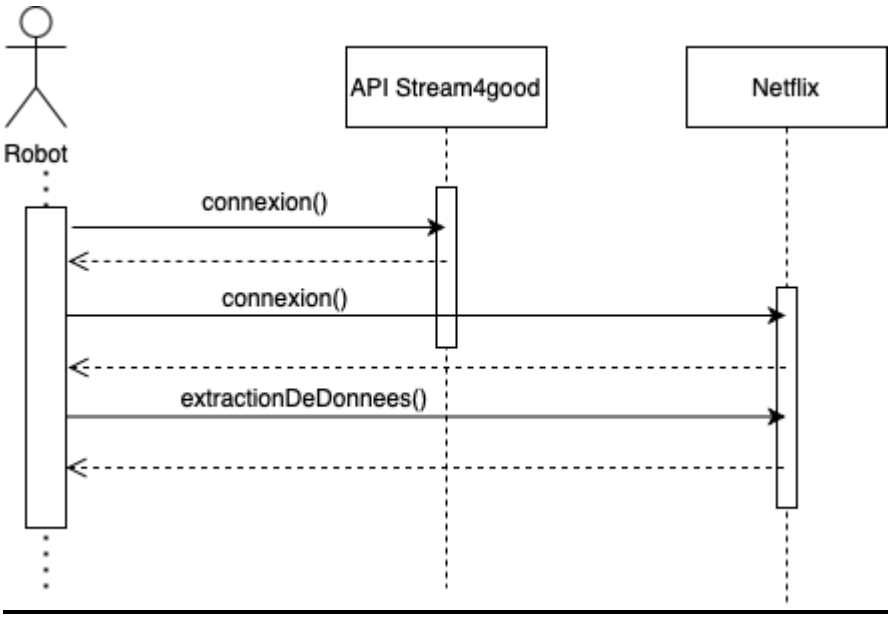
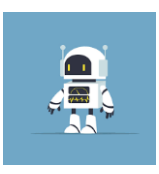


Figure 9 : Diagramme de séquence des récoltes de données du robot

Pour accéder à l'extraction, le robot doit se connecter à l'api stream4good afin de demander les identifiants qui lui permettront de se connecter sur Netflix. Une fois connecté sur Netflix ce dernier entamera son prélèvement de données sur le site qu'il transmettra ensuite au chercheur comme vu dans le diagramme précédent.



1.7 – CHOIX D'IMPLÉMENTATION / TECHNIQUES

En amont, avant de commencer le développement des scripts de robot en python, nous reviendrons d'ailleurs sur le choix de ce langage, il a fallu établir un environnement d'intégration et de déploiement continu, de ce fait nous avons choisi d'utiliser le logiciel de gestion de versions décentralisé git en rejoignant l'organisation Stream4Good sur Github, en bénéficiant d'un standard de livraison Travis/Docker, nous avons donc dès le départ réfléchis à intégrés chaque robot à son conteneur docker pour faciliter la diffusion à toute la flotte de Raspberry PI en cas de mise à jour en une seule commande à distance.

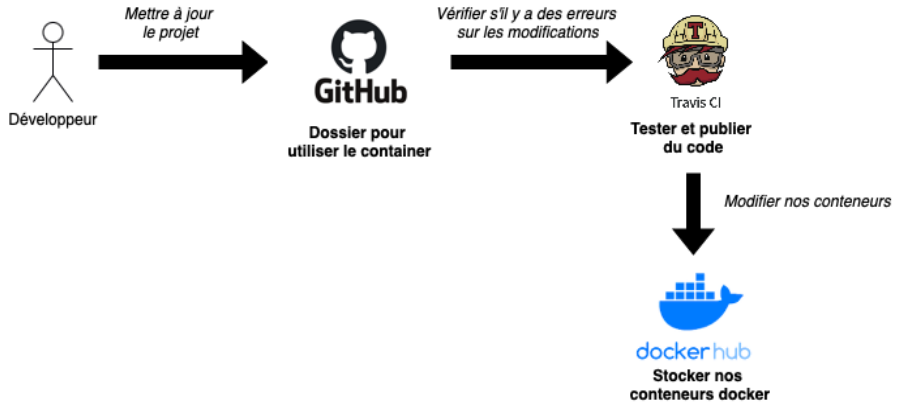


Figure 10 : Schéma du déroulement d'une mise à jour

Figure 11 : Les logo de arm et docker

Les conteneurs Docker:

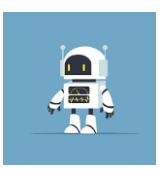
Les Robots:

Concernant nos robots qui ne demandent pas à visionner du contenu en streaming, un Dockerfile avec une image basé sur Alpine, un navigateur internet, un compilateur et python 3 est suffisant.

Vous remarquerez la mystérieuse ligne "apk add xvfb" dans nos dockerfile, pourquoi ajouter un serveur d'affichage au lieu de lancer chromium-browser en headless (c'est à dire sans affichage graphique) pour éviter une surcharge de la ram et l'intense utilisation CPU connu des serveurs d'affichages surtout à bord d'un PI? La réponse se trouve du côté de l'extension chrome, il est en effet impossible de lancer chromium en headless et en même temps installer l'extension PRIME-SPACE-DATAVIZ au lancement du navigateur voir l'issue officiel ici

Le premier challenge pour notre robot "Watcher" c'est qu'il n'existe aucun navigateur compatible Netflix sur une distribution open source pour l'architecture ARM, les seules autorisées à visionner du contenu Netflix sont Chrome OS et Android, nous étions donc bien loin de ce territoire. Cependant grâce à la communauté active autour de l'OS Raspberry PI un certain Vpetkov à développer un script qui extrait les librairies DRM nécessaire de Chrome OS pour les implémentés dans le Raspberry PI, le fichier est "*libwidevine.so*" qui a fonctionnés à bord de nos systèmes en modifiant l'User Agent au lancement du navigateur internet.

Enthousiaste à l'idée d'intégrés la même méthode à notre conteneur docker nous avons tenté la même approche sans succès, en effet chromium-browser sous Alpine est personnalisé de sorte à être le plus léger possible dépossédé de quasi toute librairie multimédia il n'était donc pas possible de l'utiliser, en comparaison le chromium-browser des contributeurs PI OS



est personnalisé de sorte à supporter un maximum de contenu multimédia.

Le second challenge était donc de créer le conteneur Docker pour le robot watcher, il n'existe pas d'image original Raspberry PI OS dans le docker hub, nous avons donc entrepris de construire une image docker en partant d'ubuntu focal en y installant le chromium-browser du Raspberry PI, il a fallu au total environ 135 paquet supplémentaires pour rendre ce navigateur personnalisé compatible avec ubuntu mais le résultat est correct, le robot watcher au sein de ce conteneur docker fonctionne de manière fluide d'après nos nombreux test, le record est un visionnage de film/séries pendant 56 h sans interruption, l'exception était dû à l'exécution d'un conteneur d'un collègue utilisant les mêmes identifiants.

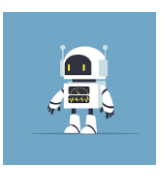
Il est évident que avec du recul une solution plus intéressante serait de baser le conteneur sur une image Debian:buster et de rajouter le répertoire Raspberry PI et sa clé publique pour installer chromium et toutes ses dépendances directement à partir d'un simple apt-get install chromium-browser cela facilitera grandement la maintenance et les mises à jour, cette tâche est en cours de réalisation, une autre façon beaucoup plus "puissante" serait d'émuler Raspberry PI OS avec QEMU et de build le dossier, cette tâche est en cours de R&D.

Le Dashboard et L'API:

Le Dashboard locale qui ordonne le lancement et l'arrêt des robots se situe à bord d'un conteneur avec comme base une image Alpine avec un ajout et configuration d'un serveur nginx et de php 7 avec toutes les extensions nécessaires au fonctionnement de Symfony 5.2, l'exposition du port 443 permet de sécuriser l'accès au conteneur.

La base de données:

MySQL est notre SGBD qui reçoit tous les logs et les données collectées par les robots, le conteneur à une image directement du docker hub mysql-amd64 et mysql-arm7l



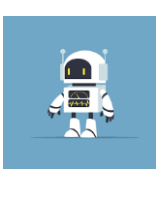
Les Scripts Python:

Les Robots:

Le choix du langage python pour le développement de nos robots fut intuitif, Python jouit d'une notoriété dans le milieu de l'extraction et l'analyse de données car il bénéficie d'une large librairie d'extensions et d'une communauté d'experts analystes de renom! Mais c'est surtout la rapidité d'exécution sur nos machines et la facilité d'intégration de sélénium et le paramétrage efficace du chrome-driver qui nous ont influencés dans notre choix. En effet Sélénium est un Framework de test informatique qui nous a permis de simuler l'interaction des utilisateurs avec le navigateur lors de la visite des sites de streaming.

Le Daemon:

Ce script est la pièce maîtresse de notre système, il permet de faire le lien entre le Dashboard et les robots, il lance et arrête les conteneurs docker et permet l'envoi de fichier log.csv à travers l'API vers le Dashboard, c'est un chef d'orchestre! Le choix d'utiliser le langage Python est tout d'abord l'existence de la librairie docker-pi qui permet d'ordonner de manière intuitive le lancement ou l'arrêt des robots et en second lieu l'efficacité des scripts python à s'exécuter en arrière-plan.

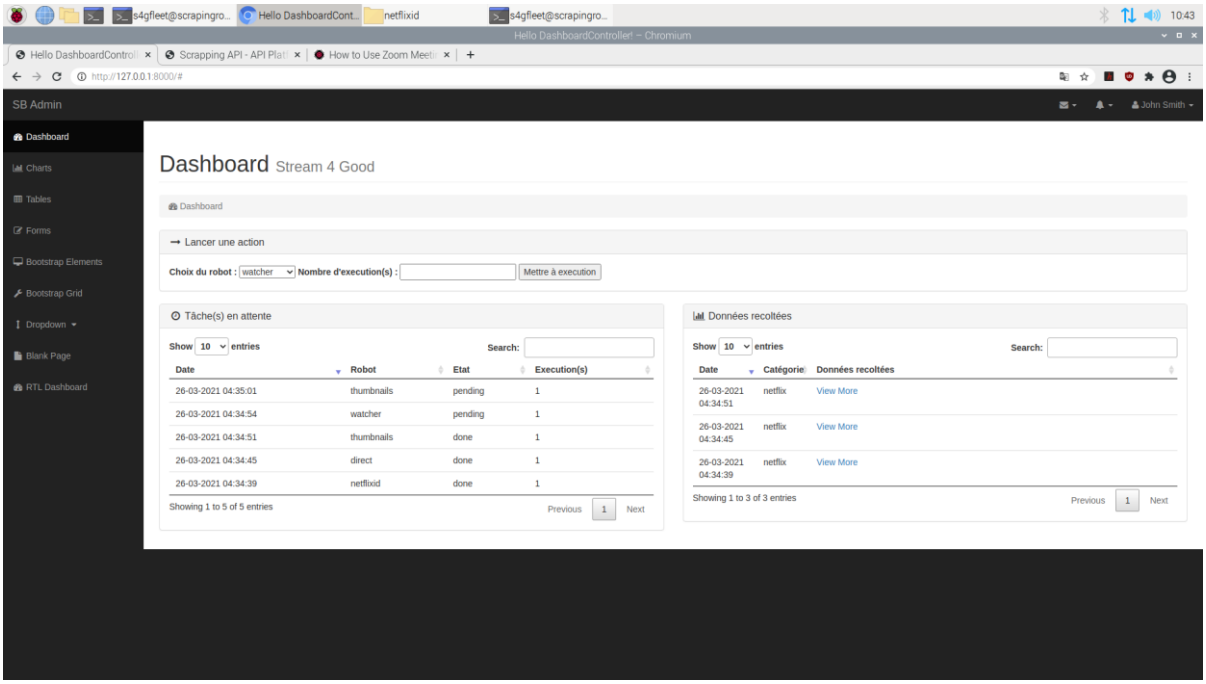


Le Dashboard locale:

Figure 12 : Logo de API platform et Symfony

Le framework français Symfony et son homologue API Platform forment la fusion parfaite d'un dashboard et d'une API efficace pour répondre rapidement au
besoin de notre système.

En effet, à bord de ce dashboard vous pouvez sélectionner les robots que vous souhaitez lancer et le nombre d'exécution de chaque robot, ils seront tous exécutés dans l'ordre, une fois finies les données récoltées seront listées à droite dans la section: données récoltées.



Le Dashboard distant:

La page de commande UpSwift nous permet l'enregistrement, le tracking gps et un contrôle à distance de notre flotte IOT, l'utilisation principale de ce dashboard sera la mise à jour automatique de toute notre flotte en un seul clic! En effet le système nous permet de lancer une ou plusieurs commandes en même temps sur nos appareils qui mettent à jour toute notre stack! Nous pouvons également instaurés un remote Ssh/VNC/Webview avec un système particuliers en cas d'incident.

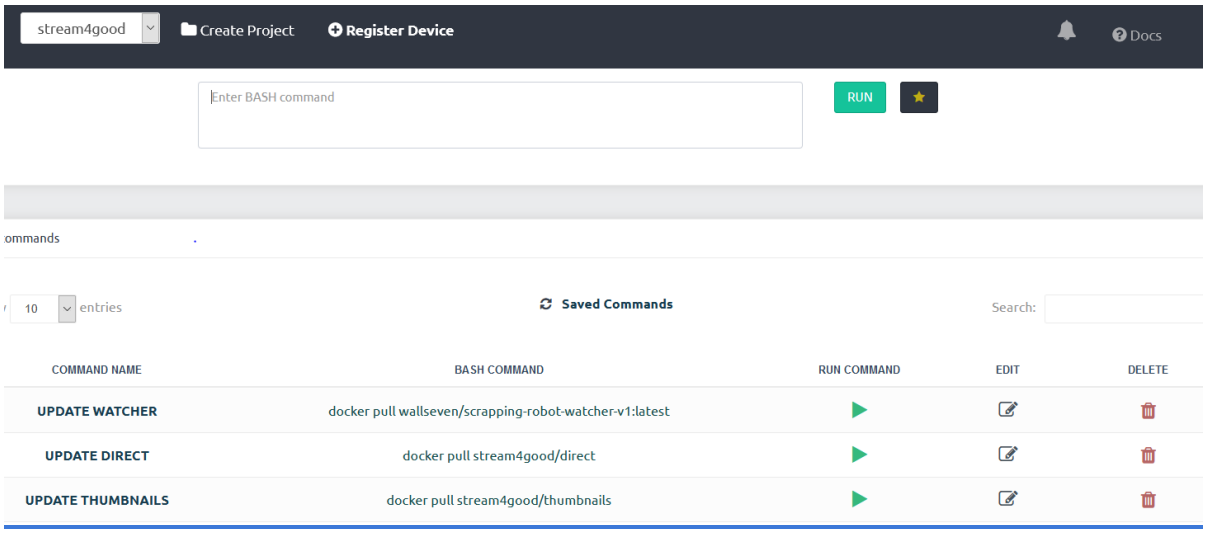


Figure 13 : capture d'écran du dashboard

Installation:

Nous avons pensé le déploiement de solution d'une façon simple et efficace, cloner le repertoire git sur votre appareil, la branche principale est réservé à la conception de robot de scraping, et les deux restantes pour le déploiement selon votre architecture des README sont disponibles pour vous aider dans ce sens.

1.8 – CHOIX MATÉRIEL

Le contexte de ce projet nous a amené à fournir un rapport pour l'achat de matériel à notre commanditaire. En effet, nous avions besoin d'ordinateurs compacts pouvant tenir sur une main et assez puissants pour faire ce dont on avait besoin.

1.8.1 - Les contraintes liées au produit

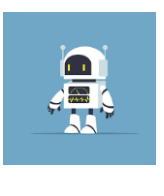
Afin de récolter nos données on devra utiliser plusieurs applications et programmes en simultanées. Ainsi, l'ordinateur doit être assez puissant pour lancer plusieurs conteneurs docker qui contiendront nos applications. De ces contraintes, nous avons déduit deux principales caractéristiques que doit posséder notre machine.

1.8.1.1 - La carte réseau

La première est d'avoir une carte réseau assez puissante pour pouvoir constamment réceptionner et envoyer les données au serveur sans bug, sans ralentissement et le plus rapidement possible. Ainsi, nous souhaitons qu'il soit compatible Ethernet car ce branchement permet d'avoir une meilleure connexion internet. Toutefois, il est bon de noter que le wifi n'est pas mis de côté : de fait, une entreprise ne souhaitant ou ne pouvant pas connecter l'ordinateur en Ethernet pourra le connecter sur le wifi.

1.8.1.2 - La RAM (mémoire vive)

Quant à la seconde : c'est qu'il doit posséder une mémoire RAM de 2 gigas au minimum. En effet, nous avons fait des tests sur nos ordinateurs respectifs (bien sûr mieux configurés) et chrome utilise 350~450 mo de ram pour un seul onglet Netflix avec Sélénium. Dans le cas du nano-ordinateur nous ne savons pas exactement l'évolution que va avoir l'extension chrome car, si une prise en charge de robots multiples avec IMDB/YT/OCS est prévue et sachant que chrome grimpe très vite en ram : il faut être prévoyant. Par ailleurs, il est bon de savoir que le Raspberry PI OS (système d'exploitation debian des raspberry) utilise entre 300 et 400 mo de ram. D'autre part, nous savons que l'utilisation continue d'applications intensifie l'utilisation de la ram. On peut donc retenir qu'on a un giga de ram utilisé. On remarque alors, que pour un raspberry comprenant un giga de ram : la ram est complètement utilisée et laisse donc la possibilité de lags. C'est en conséquence de quoi on a décidé de prendre la gamme de raspberry supérieure soit celle de 2 gigas comme énoncé ci-dessus. Cela nous semble nécessaire afin de garantir la stabilité minimale du système



1.8.1.3 - La carte mère

La carte doit consommer peu d'électricité notamment via un système d'exploitation optimisé (Raspberry Pi OS) pour sa configuration hardware. La carte doit utiliser le moins de ressources possibles pour sa construction (métaux rares, matières premières...) pour réduire l'impact environnemental de sa construction ainsi que son recyclage. Dans cette condition précise afin d'éviter des ralentissements dus à un manque de ram ou de vitesse du processeur nous opterons pour un modèle plus récent pour ne pas devoir renouveler constamment le matériel surtout si celui-ci est envoyé à l'étranger pour de longues périodes.

1.8.1.4 - L'écran

La configuration et la prise en main du raspberry doivent être facilitées à l'entreprise. Ainsi pour ne pas devoir acheter un écran d'ordinateur et un câble pour pouvoir connecter l'ordinateur : nous pensons à ajouter un écran tactile qui permet la facilité d'utilisation. Effectivement, l'entreprise n'aura finalement qu'à connecter le câble d'alimentation au Raspberry PI.

1.8.2 - Le choix du matériel

1.8.2.1 - Le Raspberry

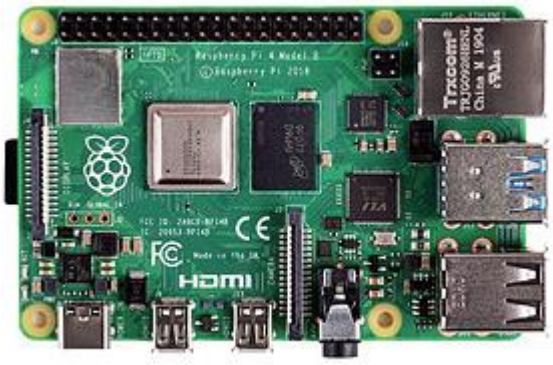


Figure 14 : Photo du micro-ordinateur

Pour choisir notre modèle on s'est donc basé sur le fait qu'il puisse prendre en charge de l'Ethernet et du wifi et qu'il ait 2 gigas de ram. Bien évidemment il doit aussi être durable (assez récent). C'est en conséquence que l'on a choisi ce produit Il y a également le raspberry PI 3 (de par sa configuration) qui nous semblait intéressant mais il n'est pas disponible avec 2go de ram sur le site partenaire LDLC Pro.

1.8.2.2 - Les composants non fourni avec

1.8.2.2.1 - Le câble d'alimentation

Figure 15 : Photo du chargeur

Après s'être documentés, on a pris connaissance de deux principales informations. La première est que ce câble essentiel au raspberry, car il fonctionne en secteur et donc uniquement branché, n'est pas fourni avec. La seconde, est qu'après avoir visité la documentation officielle raspberry, nous avons constaté qu'il est essentiel d'avoir un câble d'alimentation compatible et officiel. Autrement, cela pourrait endommager le système. On s'est donc dirigé vers le produit suivant :

1.8.2.2.2 - La carte mémoire

Figure 16 : Photo de la carte sd

Pour la mémoire, c'est similaire au chargeur. Effectivement, l'ordinateur n'en possède pas et elle est nécessaire pour que l'on puisse mettre des données dans l'ordinateur et donc in fine pouvoir l'utiliser. Nous avons porté notre choix sur une carte SD officielle avec le système d'exploitation pré-chargé. Sa capacité est de 32 gigas : ce qui nous servira amplement

pour notre utilisation. Elle est également conseillée pour une première utilisation de raspberry (facilité d'installation du système d'exploitation).

1.8.2.2.3 - Le boîtier

Figure 17 : Photo du boitier

Lorsqu'on reçoit le raspberry, on le reçoit sous sa forme naturelle : sans boite. Autrement dit on a la carte mère ainsi que tous les autres composants visibles. Nous remarquons de cette manière qu'il peut être facilement détérioré sans mentionner le fait qu'il n'est pas professionnel de fournir ce matériel de cette manière aux entreprises. On a donc pris l'initiative de prendre un boîtier. Toutefois, celui-ci est spécial car il est compatible avec l'écran dont nous parlerons juste après et qui nous paraît nécessaire.

1.8.2.2.4 - L’écran

Figure 18 : Photo de l’écran

Précédemment nous avons communiqué sur le fait qu’il nous paraît important que des entreprises puissent configurer notre matériel rapidement. En effet, il faut prendre en considération le fait qu’elles nous apportent une aide précieuse et qu’elles n’ont pas énormément de temps à nous consacrer au vu de leurs projets personnels en cours.

C’est ici que notre écran intègré au Raspberry entre en scène. De fait, lorsque les entreprises recevront le produit : elles n’auront qu’à le brancher et suivre les instructions. Du fait, de l’écran tout pourra se passer de manière efficace et rapide.

Cet écran permet également que notre produit prenne moins de place dans leurs locaux et soit ainsi moins encombrant. En effet, si on faisait le choix de l’écran d’ordinateur, ces derniers devraient réfléchir à quel espace allouer celui-ci : ce qui peut être très contraignant pour les petites entreprises.

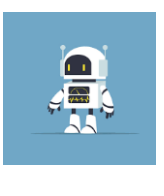
Pour limiter les risques, notre choix s’est porté sur un écran officiel et compatible.

1.8.3 - Le matériel définitif

Les spécificités que nous requérions pour ces outils étaient le minimum selon nous et ce qui est bien est que nous sommes en dessous du budget.

Ainsi, notre commanditaire a décidé de nous commander des appareils plus évolués même si plus coûteux. Par conséquent, il nous a pris une des dernières gammes de Raspberry : les Pi4B avec 8GB de RAM soit, bien plus que ce dont nous avions besoin. D'autre part, ils ont mémoire de 64 go qui est également au-dessus de nos estimations. Enfin, il n'y a qu'un seul écran car il ne sert que pour la démonstration. Si celle-ci est convaincante, tous les appareils en seront équipés.

Par conséquent, nous avons des appareils et un équipement de qualité pour travailler sur ce projet qui nous a été confié.



2 – ARCHITECTURE

2.1 – Schéma

Notre projet qui comprend trois grandes parties importantes peut être modélisé sous la forme d'un schéma. Nos micro-ordinateurs sont à la base de ce projet. On le représente sous la forme du diagramme ci-dessous.

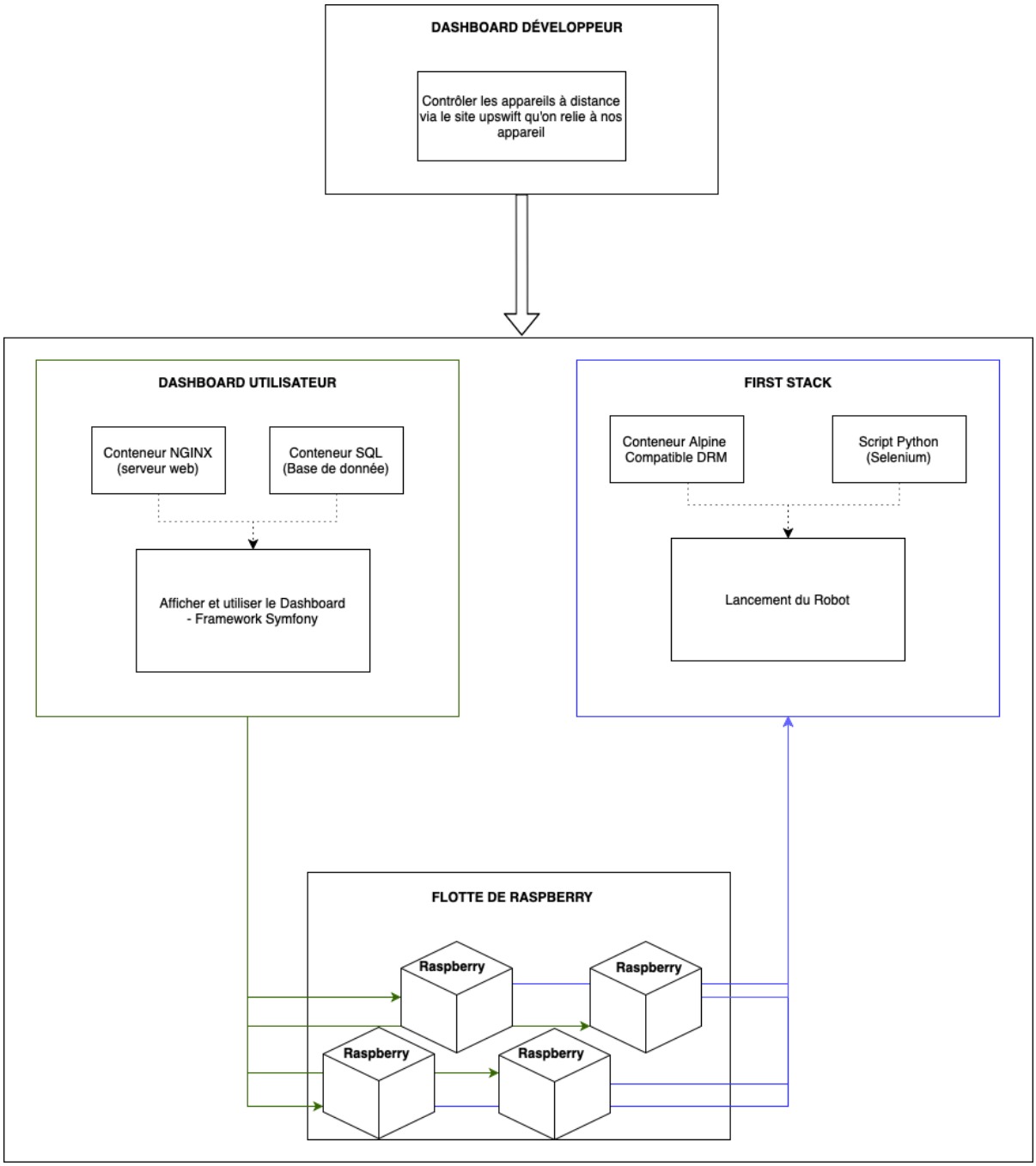


Figure 19 : Schéma d'architecture

En vert est représenté les éléments qui permettent aux chercheurs d'utiliser le Dashboard du Raspberry. Ces éléments sont deux conteneurs docker. Le premier conteneur Nginx embarque un serveur portant le même nom et sur lequel est présent dessus une instance du Framework PHP : Symfony.

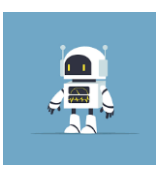
Le second est un conteneur SQL dédié à la base de données de l'application web. Ce mélange permet de créer un Dashboard permettant aux chercheurs d'utiliser convenablement les Raspberry en stockant notamment les données récupérées.

Par ailleurs, en violet on peut retrouver par quels moyens le Raspberry fait de l'extraction de données. Le script Python importe la librairie sélénium qui permet de crée un parcours de test et d'installer l'extension Prime-Space.-Dataviz

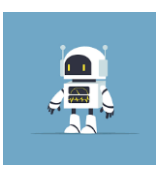
Enfin, au-dessus on peut voir de quel manière le développeur peut contrôler l'ensemble de ce système. Pour ce faire, on passe via un site qui permet de contrôler des micro-ordinateurs à distance. En effet, après avoir configuré notre machine, il est possible d'envoyer des commandes et de la contrôler complètement via cette application. D'autre part, on a accès à toutes les données de la machine comme si on l'avait sous la main y compris son emplacement.

2.2 – Questionnaire

N°	Question	Réponse
1	Votre architecture vous permettrait-t-elle de supporter facilement une autre langue utilisateur (français et anglais par exemple) ?	Oui. Il suffit de crée un fichier « translations »sur l'application web. Ainsi, on pourrait aisément changer de langue.
2 (web)	Évaluez la difficulté du passage de votre application web à celle du mobile.	La technologie web que nous utilisons nous permet de faire du responsive. En d'autres termes, notre site s'adapte à toutes les tailles d'écran y compris ceux des mobiles.
3	La modularité est un critère important de qualité des architectures (Un composant ou classe n'a qu'un seul objectif). Les composants de votre architecture	Oui, parfaitement.



	respectent-ils ce principe de modularité ?	
4	Des éléments de votre architecture seraient-ils réutilisables dans un autre projet ?	Oui. Ainsi par exemple le docker (intégration d'environnement) nous permettant de faire du symfony (framework php) peut justement être utilisé sur tous les projets symfony avec des appareils à architecture ARM.
5	Votre architecture permettrait-elle de changer de SGBD facilement ?	Oui. Sur l'application web, il suffirait seulement de déclarer le changement au framework.
6	Si vous deviez changer totalement la réalisation de l'interface graphique, y-aurait-il des composants ou classes inchangé(e)s ?	Oui. Les contrôleurs (présents sur la partie serveur) seraient inchangés car ils s'occupent principalement de la transmission de données au front (partie visuelle de l'application). C'est là l'intérêt de les diviser car cela permet la modulabilité du code.
7	Quels seraient les effets d'un éventuel changement dans le schéma de la base de données (ajout d'une nouvelle table ou nouvel attribut, changement du nom d'une table ou d'un attribut, changement sur un type d'attribut, sur une contrainte...) dans l'application ?	Sachant que notre base de données servira au stockage de données massives ; si l'on supprimait une colonne ça pourrait être néfaste puisqu'on perdrait une quantité importante de données. Par contre, celle-ci est ouverte à l'ajout et à la mise-à-jour.



3 - BILAN

3.1 - Retour sur le travail réalisé

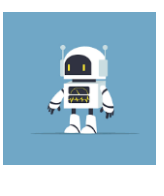
Ce projet était tout autant un accomplissement de chacun que l'occasion d'apprendre et d'élargir notre horizon dans le domaine des technos que nous apprenons en cours. Nous avons réussi à respecter toutes les demandes du commanditaire grâce à nos nombreux jalons et grâce à l'aide de tous les acteurs de ce projet. Nous avons pu créer des robots autonomes pouvant récolter des données, un Dashboard et une base de données permettant de stocker les log de et scraping de chaque robot.

Nous avons rencontré plusieurs difficultés liées d'une part à la prise en main des logiciels, nouveaux pour l'équipe et d'autre part la compatibilité d'un logiciel à un autre qui s'est révélée laborieux. Toutefois, ce projet a permis à tous les membres de l'équipe de développer des compétences techniques mais aussi pratiques. Il a apporté, à l'ensemble de l'équipe, des atouts en termes de collaboration, d'organisation ainsi que des compétences informatiques. Malgré le fait que la plupart des logiciels étaient inédits pour nous, nous avons pu nous former et nous entraider tout au long du projet. A l'instar des différentes personnes qui ont été disponibles pour nous aider et répondre à nos questions.

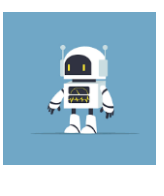
Cependant, notre projet comporte des limites. La première et la plus importante, avant qu'elle soit informatique, est dans la qualité des données, il faut pousser les Persona au possibles et diversifiés les types de parcours utilisateurs dans notre robot qui consomme du contenu multimédia afin de ne pas altérés la base de données existante par un parcours homogène qui ne reflète pas l'habitudes des visites de ces plateformes.

La seconde limite est prévisible mais inévitable, c'est le changement de code source du site de Netflix ce qui mettrait H.S. toute notre flotte de robots et peut mettre à mal l'API d'extraction, cependant les équipes de la Miage Paris 1 veillent au tournant et régleront ce problème en un seul "sprint" chrono en main!

La troisième limite et qui est certainement la plus complexe dans le cadre de ce projet sont les DRM Netflix, en effet si la communauté Raspberry ne souhaite plus trouvé de solution pour contourner ce problème notre robot watcher risque de se retrouver tôt ou tard hors d'état, et c'est précisément dans ce cadre que nous avons pris les devants pour apprendre à disséquer Chrome OS et mettre la main sur la dernière version de libwidevine.so de nos propre moyen nous allons par la suite à ce projet construire une image docker basé sur Raspberry PI OS qui intégrera un script de mise à jour automatique de cet librairie pour pérenniser au possible nos accomplissement!



La dernière limite est celle de la sécurité, nous n’avons pas l’expertise pour se prononcer à ce sujet mais d’un côté réseau nous pensons que la connexion en VNC en cryptant la connexion et en utilisant un système de vpn interne pourrait être une possibilités d’un accès à distance propriétaire en s’inspirant de ce que fait UpSwift. Du côté hardware si un individu mal intentionnés se retrouve en possession d’un Raspberry PI de notre flotte il pourrait extraire la carte micro sd et procédés à des modifications malveillantes et infectés toute la flotte ou pire encore infiltré le réseau du laboratoire qui nous “host” et faire des dégâts ce qui serait préjudiciable à notre projet et à l’image de ce que nous sommes capables de faire, cependant une solution est à l’étude et va bientôt prendre place au sein d’une équipe M1 qui souhaite utiliser une JavaCard pour empêcher toute modification de contenu non autorisé, je me permet de citer ces quelques lignes de Wikipédia:” La plate-forme de JavaCard supporte l'isolement de contexte et d’applications. L'isolement de contexte assure que les objets créés, donc en possession d’applications fonctionnant dans un même contexte, ne peuvent pas être accédés par des applications d'un autre contexte à moins que ces applications possédant ces objets ne fournissent explicitement des interfaces pour l'accès. De telles interfaces sont appelées des interfaces en commun et des objets implémentant ces interfaces, appelés *commun interface objets*, constituent des points d'entrée légaux à ces applications.”



3.2 - Retour réflexif sur l'action

3.2.1 - Oualid GUENNIF

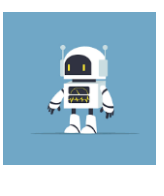
Si ma motivation à la réalisation de ce projet est sans faille, pouvons-nous en dire autant de ma place de PO au cours de ce projet et même au sein de tous les projets au cours de l'année où mes collègues m'attribue souvent cette casquette, est-ce parce que je démontre une réelle énergie à la réalisation de projets qui me parle et d'activités qui nous sortent du cadre classique des cours ou parce qu'ils sont certains que j'irais au bout de ce qu'on m'a confié.La responsabilité du PO n'est pas une mince affaire, en effet ce n'est pas une tâche aisée de répondre parfaitement aux attentes des utilisateurs à chaque nouvelle fonctionnalité, le PO est responsable de la qualité fonctionnelle, c'est-à-dire qu'il doit garantir la disponibilité du livrable à une date X!

Le choix de former ce groupe en compagnie de Grasaan et Nasser est tout d'abord leurs réactivités et leurs possibilités à trouver rapidement des solutions au cours de projets accomplis ensemble au début de cette année, ils sont d'ailleurs en tête au sein de la L3 Miage avec une appétence et quasiment un don pour la gestion de projet pour Nasser. Grasaan fournit une capacité à analyser pendant tout le processus de développement et la conformité de ce qu'on développe avec l'attente des utilisateurs et n'hésite pas à pointer du doigt lorsqu'on dérive des demandes initiales et cherche à tout prix à diminuer le ratio risque pour nos livrables, ils ont été d'une efficacité précieuse à la réalisation de ce projet !

Ce que je retiens du côté technique de ce projet c'est que la mise en place d'un environnement de développement cohérent et efficace dès le départ peut être déterminant pour le projet, j'ai d'ailleurs été agréablement surpris par la disponibilité de la communauté linux qui n'hésite pas à partager leurs astuces ou répondre aux questions avec un niveau technique conséquent !

J'ai pris ce choix de m'inscrire à la Miage en analysant en amont les UE dispensés et en étant certains de leurs concordances avec mes appétences et compétences, l'énergie que je mets dans mes travaux et leur réalisation sont motivés par une envie de réussir à accomplir des projets avec du sens et pas seulement dans la finalité du diplôme.

La citation du grand homme Albert Einstein: « dès que tu cesses d'apprendre, tu commences à mourir ». Me parle de plus en plus, je ressens un besoin de découvrir et d'apprendre dans le milieu des technologies en général, c'est d'ailleurs dans cette optique que je n'ai pas hésité à saisir le projet de scraping de données. Évidemment la liste des livrables demandés m'a tout de même fait réfléchir à deux fois, cependant Monsieur Herbaut a su nous montrer l'étendue du projet, son but, son utilité, notre rôle à jouer au sein de plusieurs équipes comme si nous étions tous à égal dans ce



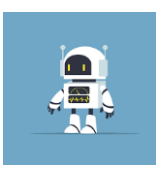
projet, à ce stade-là je ne me posais plus la question de comment je vais faire pour réaliser tout ça mais plutôt « on va réaliser le projet parce qu'on compte sur nous !»

Cette situation c'est aussi répétés lors du choix des projets de développement web, en effet le site d'e-commerce de soins capillaires que nous mettons en place avec mon équipe est destinés à la mise en vente de produit en provenance d'une société britannique, lors des choix des sujets nous pouvons créer un site beaucoup plus simple mais surtout sans impératifs à livrer en dehors du cours, cependant je me suis engagés et en conséquences je me dois de me renseigner sur tous les aspects de Symfony et de l'attente du client. Ce qui augmente significativement mes compétences en développement web.

Cependant, quelque chose de nouveau se dessine à l'horizon, un monde de l'environnement connecté où les données seront difficilement falsifiables de par la blockchain, la signature numérique et les smarts contracts. Quelle est l'application des objets connectés à une échelle industrielle ? Que se passera-t-il si l'on combine IOT et blockchain, pouvons-nous implémenter une IA directement dans la blockchain ? Je n'ai pas la réponse à ces questions mais je commence déjà à me renseigner sur l'assemblage d'un objet connecté, les différents types de puces comme la STM32 F7 à basse consommation, les réseaux LoRaWan et Sigfox.

Je pense que nous sommes plus qu'au lancement d'un grand cycle de transformation digitale qui commence déjà à bouleverser nos habitudes tant au niveau technologique qu'écologique. C'est plus qu'une croyance c'est une conviction, une vision . Si nous voulons que les prochaines générations puissent survivre et c'est peu dire qu'il est de notre responsabilité de contrôler en amont les ressources de la planète et de créer des villes autosuffisantes et écologiques.

3.2.2 - Grasaan GRATION



Ce projet commun était une expérience assez unique. En effet, je n'avais jamais été aussi occupé par tant de projets simultanément de toute ma vie. Tout ce stress et cette pression m'a fait penser à beaucoup de décisions qui m'ont toujours mené vers la désorganisation et le stress. C'est pourquoi j'ai décidé d'appeler l'objet de ma réflexion : « La procrastination ».

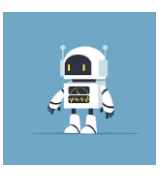
Au début, le projet était un peu flou. J'avais les réponses du pourquoi, mais aucune idée du comment. Ce flou s'était peu à peu dissipé à chaque explication de mes camarades ainsi que de Mr Herbaut, mais il ne disparaissait pas complètement car je n'étais toujours pas capable d'expliquer comment nous allions atteindre cet objectif. Ce flou était aussi la cause de ma réticence à l'égard de ce projet au début de celui-ci qui m'a mené dans un cercle vicieux. Je n'arrivais pas à travailler et à m'organiser à cause de ce brouillard et vice versa. J'étais oisif à l'idée de se renseigner afin de dissiper ce flou.

Je savais que ce n'était pas un comportement logique de vouloir se mettre des bâtons dans les roues juste à cause d'un simple souci de compréhension. Surtout à une époque où l'accès à tout type d'information se fait sur le bout des doigts.
C'est ici que j'ai revu mon plus grand ennemi à ce jour qui n'était ni plus ni moins que la procrastination.

Elle était la cause de tant de stress et de pression qui ont mené vers des travaux qui auraient pu être effectués avec plus de soin et de qualité . Mais elle ne s'est pas arrêtée là. Nombreuses sont les occasions ratées qui m'auraient permis d'être le meilleur de moi-même en matière de parcours scolaire mais aussi humainement parlant.

Ce qui fait la force de la procrastination, c'est qu'elle fait partie de nous-même et de notre train de vie. Pour la combattre, il faut combattre nos habitudes , notre façon de travailler mais aussi d'être. Et pour ce faire, il faut chercher comment s'y prendre et la commence un vrai cercle vicieux. Celle-ci s'était encore plus empirer avec l'arrivée du confinement qui à laisser libre cour aux différentes distraction.
C'est avec ce projet et le contexte assez particulier qui m'ont permis de connaître les principaux points faibles de mon ennemie.

Tout d'abord, comme je l'ai dit précédemment, la méconnaissance autour d'un sujet enlève l'envie et ainsi la motivation de travailler qui sont nos seules armes ici. C'est pourquoi, il est primordial de bien se renseigner et connaître le sujet ainsi que les enjeux du projet dès le début de celui-ci afin de dissiper le flou et avoir une vision claire du chemin à prendre pour pouvoir s'organiser.
Ensuite, le deuxième point faible de la procrastination est l'environnement de travail. Celui-ci est très important surtout pendant le télétravail car les tentation de faire tout sauf travailler sont nombreuses. Pour créer un



environnement de travail, il n'y a pas de secret, il faut séparer le lieu de travail de celui où l'on passe la plupart de son temps.

Ces résolutions ont été très dur à mettre en place durant le projet surtout à cause de la complexité de celui-ci qui créait encore une fois un flou dans la manière de procéder. En effet, cette même complexité résidait dans le fait que nous utilisions des logiciels ainsi que des langages de programmation que je n'avais jamais utilisés auparavant.

3.2.3 - Nasser HARTI

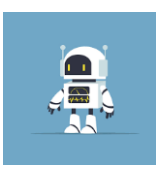
Pendant le début du projet je me suis énormément questionner sur notre choix concernant notre matériel. Nous allons appeler cet événement : "La décision".

En effet, comme expliqué plus haut dans le rapport nous avions dû au tout début du projet émettre une décision qui avait comme sujet le matériel que nous allions utiliser durant tout le projet. Pour se faire on à dû se documenter ainsi que faire des tests sur nos propres machines.

La problématique principale à laquelle j'étais principalement confronté est que je n'avais aucune connaissance sur les nano-ordinateurs. De ce fait, j'ai dû m'atteler à énormément de recherches. Ces recherches étaient obligatoires car notre projet en dépendait. Effectivement l'enjeu était de taille. C'est notamment car si nous avions du matériel inadéquat, le projet était infaisable. C'était donc une étape pilier, une des plus importantes du projet.

La raison pour laquelle je n'étais pas serein est que même si notre documentation était de taille rien ne nous disait que cela fonctionnerait vraiment. En effet, je suis une personne qui est adepte d'une phrase connue qui est : "il faut le voir pour le croire". C'est vrai, si on ne le voit pas comment être sûr à cent pourcent de son efficacité ?

Tout au long de cette expérience j'ai pris le temps de me remettre en question et réfléchir à plusieurs problématiques liées à ce que je viens d'évoquer. Il y en a une qui m'a particulièrement interpellé. Celle-ci n'était pas des moindres car elle était en relation directe avec le monde du travail et ainsi également mon avenir. "Si je ne suis pas capable de faire des comptes rendus détaillés dont je suis sûr pour évaluer une technologie inconnue, comment je ferai quand on me le demandera en entreprise". La partie la plus importante de ce questionnement est "dont je suis sûr". Effectivement, toute la documentation existante ne pourrait pas me convaincre tant que je n'aurais pas vu le produit à l'action sous mes yeux.



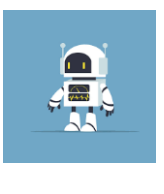
Pour tester le produit il faut des ressources. Détenir le produit, c'est échanger de la monnaie en contrepartie. J'insiste sur ce point qui paraît des plus triviaux car ce fut l'élément qui a contribué à mon blocage et qui à nuit à mon assurance. En effet, pour moi notre commanditaire nous donnait l'opportunité de nous équiper pour notre projet avec du matériel dont les frais ne nous seraient pas imputés. Se tromper sur cette technologie c'était lui nuire. C'est en tout cas ce que je pensais.

En effet, c'est ici qu'a débuté ma grande remise en question qui s'ensuivit une décision qui est très importante à mes yeux. Cela a pris sens après m'être énormément documenté sur le sujet et m'être souvenu d'un témoignage d'un de mes professeurs de base de données avancées que j'avais eu pendant mon cursus du diplôme universitaire de technologie dans l'informatique.

Effectivement, ce que je prenais pour un enjeu en était en fait un mais pas dans le sens que je pensais. Au début, pour moi c'était un enjeu car aucune faute n'était permise. Fauter c'était rater, c'était échouer. Cette pensée découle du fait de tous les éléments que je vous ai expliqués ci-avant. Toutefois, je pense désormais que je me trompais. De fait, je peux me tromper mais pas fauter tout comme je peux réussir mais fauter.

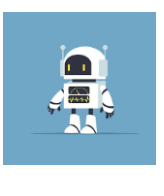
Si je peux réussir en fautant c'est parce que j'aurais bien étudié mon sujet et c'est ce qui me paraissait le plus viable et ce après moultes réflexions associées à un panel de discussions argumentées par des recherches solides. Prendre du matériel, c'est prendre un risque et si finalement en recevant le matériel on s'aperçoit qu'il n'est pas bénéfique, ce n'est pas une fin en soi. En effet, cette injection de ressources financières a permis d'écarter un élément qui paraissait des plus plausibles et ouvre in fine la voie à d'autres possibilités concurrentielles intéressantes à creuser. À contrario, si les recherches ne sont pas de qualités mais que le matériel réussi à faire son travail avec plus ou moins de problèmes : cela peut être délicat. Effectivement, il se peut que d'autres offres sur le marché correspondent plus à notre besoin et ce n'est donc pas le plus propice au projet actuel. Le véritable enjeu finalement selon moi c'était de prendre une décision juste seulement avec tous les éléments portés à ma connaissance. Par conséquent, savoir se documenter convenablement est essentiel que ce soit dans un projet ou dans les métiers accessibles via mon parcours MIAGE.

Pour en revenir à ce que m'avait expliqué mon professeur à l'époque ; c'était qu'il avait dû, pour son entreprise, établir si une technologie que cette dernière venait d'acquérir était viable ou non. Après une argumentation étayée par une formidable documentation, il leur à démontrer que cet achat pourrait baisser les performances de l'entreprise et par extension ses bénéfices. Cet achat qui leur a coûté en ressources à finalement était mis de côté pour le bien de cette enseigne. De cette



manière, j’ai compris qu’un échec, s’il en est un, n’est pas une fin en soi mais un nouveau départ.

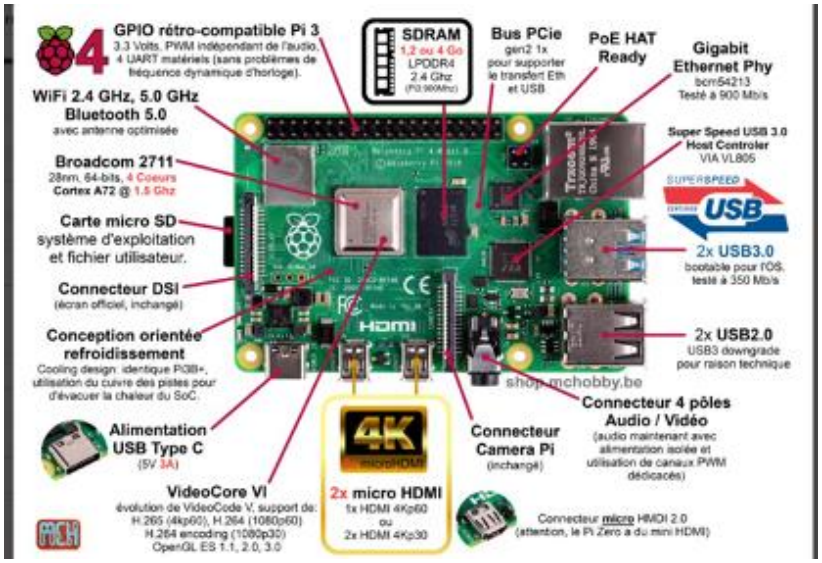
De ce fait, j’ai pris une décision : me documenter le maximum possible et ainsi éviter de manière équivalente ce que j’appelle les vrais échecs.



4 - Annexe

4.1 - Références des matériels requis

4.1.1 - Micro-ordinateur



Micro-ordinateur choisi : Raspberry Pi 4 Modèle B

2GB ARM Cortex-A72 Quad-Core 1.5 GHz RAM 2 Go - micro HDMI - USB 3.0 - USB 2.0 - USB-C - Gigabit Ethernet - Wi-Fi - Bluetooth 5.0

Prix : 64€94 TTC

Lien du produit : https://www.ldlc-pro.com/fiche/PB00273915.html

4.1.2 - Chargeur

Chargeur choisi : Adaptateur secteur officiel compatible Raspberry Pi 4B

Prix : 12€50 TTC

Lien du produit : https://www.ldlc-pro.com/fiche/PB00273953.html

4.1.3 - Carte mémoire

Carte mémoire choisi : Carte mémoire avec système d'exploitation pré-chargé pour Raspberry Pi 4B

Référence : RB-NOOBS-32GB-PI4

Prix : 19€94 TTC

Lien du produit : https://www.ldlcpro.com/fiche/PB00273936.html

4.1.4 - Boîtier

Boitier choisi: Support d'écran officiel 7" pour Raspberry PI

Prix : 19€94 TTC

Lien du produit : https://www.ldlc-pro.com/fiche/PB00250994.html

4.1.5 - Ecran

Écran choisi : Écran tactile 7'' - 800 x 480 pixels - compatible Raspberry Pi

Prix : 69€95 TTC

Lien du produit : https://www.ldlc-pro.com/fiche/PB00194184.html