

Mémoire technique

Du 4/01 au 15/04/2022

Stage de 1^{ère} année du diplôme d'ingénieur en spécialisation
informatique

Développeur logiciel



Leviathan
Dynamics

1 centre commercial La Tour chez pépinière d'entreprises,
93120 LA COURNEUVE

1. Remerciement

Je remercie avant tout Karino Kang pour l'opportunité qu'il m'a offerte ainsi que pour sa disponibilité au long de mon stage, son aide et sa tolérance.

Je remercie l'ensemble de l'équipe de Leviathan Dynamics pour son accueil, sa bonne humeur ainsi que pour sa bienveillance.

Je remercie spécialement l'équipe de ping pong ainsi que l'équipe Pizza de Léviathan Dynamics pour les bons moments de relaxation sur la pause de midi.

Pour finir je remercie l'entreprise Leviathan Dynamics de donner la possibilité à un projet de développement durable et écologique de voir le jour ainsi que pour m'avoir permis d'y participer.

Table des matières

1. Remerciement	2
2. L'entreprise	4
2.1. Présentation	4
2.2. La vision de l'entreprise	4
2.3. Histoire	5
2.4. L'équipe	6
2.5. Les projets	6
3. Ma position dans l'entreprise	8
3.1. Ma mission	8
3.1.1. Module connectivité	9
3.1.1.1. Analyse de l'algorithme existant	9
3.1.1.2. Communication entre programmes	10
3.1.1.3. Compression des données	10
3.1.1.4. Communication avec le serveur	11
3.1.1.5. Sécurisation des communications	12
3.1.2. Bug fix module connectivité.	12
3.1.3. Module 4G	12
3.1.3.1. SixFab	13
3.1.3.2. Waveshare	13
3.1.3.3. Test	13
3.1.3.4. Conclusion	13
3.1.4. Dashboard web	14
3.1.4.1. Navbar	14
3.1.4.2. Page Dashboard	15
3.1.4.3. Page Analyse	16
	2

3.1.4.4.	Page Gestion	16
3.2.	Les problèmes / surprises	17
3.2.1.	Test	17
3.2.1.	Choix	17
3.2.2.	SQL	18
3.2.3.	Design	19
3.3.	Dans une start-up	19
3.4.	Apport du stage	19
4.	La suite du stage et l'après	19
5.	Conclusion	20
6.	Glossaire	20
7.	Annexe	20

2. L'entreprise

Mon stage se déroule dans une microentreprise de moins de 15 employés, situé à la Courneuve.

Dans cette partie je vais vous présenter l'entreprise dans laquelle j'ai fait mon stage, sa vision et ses projets.

2.1. Présentation

Léviathan Dynamics est une entreprise de recherche et développement industriel.

L'entreprise est spécialiste des machines thermodynamiques basées sur la Compression Mécanique de Vapeur et possède une expertise spéciale dans les domaines du chauffage / refroidissement des procédés et du traitement des eaux usées.

Les différents projets de l'entreprise se basent sur une technologie unique qui rend possible la compression mécanique de vapeur d'eau sous vide poussé (en dessous de 50 mbars). Cette technologie est la combinaison :

- D'une construction brevetée de compresseur haute vitesse qui rend possible l'utilisation de la CMV⁽¹⁾ sans avoir besoin d'utiliser des compresseurs encombrants et onéreux.
- D'une compréhension profonde de l'environnement de vide humide qui nous permet de concevoir des systèmes de gestion de vide abordables.
- De modèles numériques d'évaporation et de condensation développés en interne qui sont nécessaires pour concevoir des échangeurs thermiques performants.

2.2. La vision de l'entreprise

Léviathan Dynamics a débuté avec une seule idée en tête : remplacer les réfrigérants dangereux et polluants par de l'Eau.

Depuis 2016, l'entreprise a travaillé à lever la plus grande barrière technologique : l'indisponibilité de compresseur compact et abordable fonctionnant sous vide.

Ils ont développé leur propre technologie de compresseur

Cette technologie permet maintenant de réaliser un grand débit volumique tout en assurant un taux de compression important : elle peut être utilisée dans les groupes froids et les pompes à chaleur. La recherche et développement permettant d'améliorer la robustesse et les performances pour les rendre conforme aux standards de l'industrie de la réfrigération sont encore en cours.

Dans le même temps, parce qu'il y a de nombreuses applications qui bénéficieraient du niveau actuel de leurs avancées technologiques, ils ont décidé d'en explorer quelque-unes.

Le produit est parfait pour le traitement des effluents industriels.

Cela les a menés à commercialiser Turbevap®, un évaporateur d'effluents industriels de faible capacité qui permet de séparer efficacement les 5% de déchets liquides des 95% d'eau pures présentes dans les effluents...

Vision de Léviathan Dynamics sur les Responsabilité Sociale des Entreprises :

"L'Énergie et l'Environnement sont au centre des préoccupations de l'humanité. En tant qu'acteur dans les deux domaines, Leviathan Dynamics aide les industriels à réduire leurs consommations énergétiques et leurs déchets. En faisant cela, nous travaillons à rendre la société plus respectueuse de l'environnement.

Nous pensons également que les entreprises doivent montrer l'exemple à leurs employés et à leur environnement social. Nous sommes engagés pour rendre la société plus juste. Nous avons installé notre entreprise dans une banlieue de Paris, La Courneuve, en espérant que dans l'avenir, nous pourrions contribuer au dynamisme de cette zone."

2.3. Histoire

06/02/2016 : Intégration de l'incubateur des Arts&Métiers Paris Tech

10/10/2016 : Naissance de l'entreprise en tant que Start Up

2017 : Déploiement du site WEB -> <https://lnkd.in/dSnFCfZ>

2020 : partenariat avec Ariane Group

2020 : 1^{ère} maquette du Turbevap, un projet de traitement des effluents industriels

2021 : Mise sur le marché du Turbevap

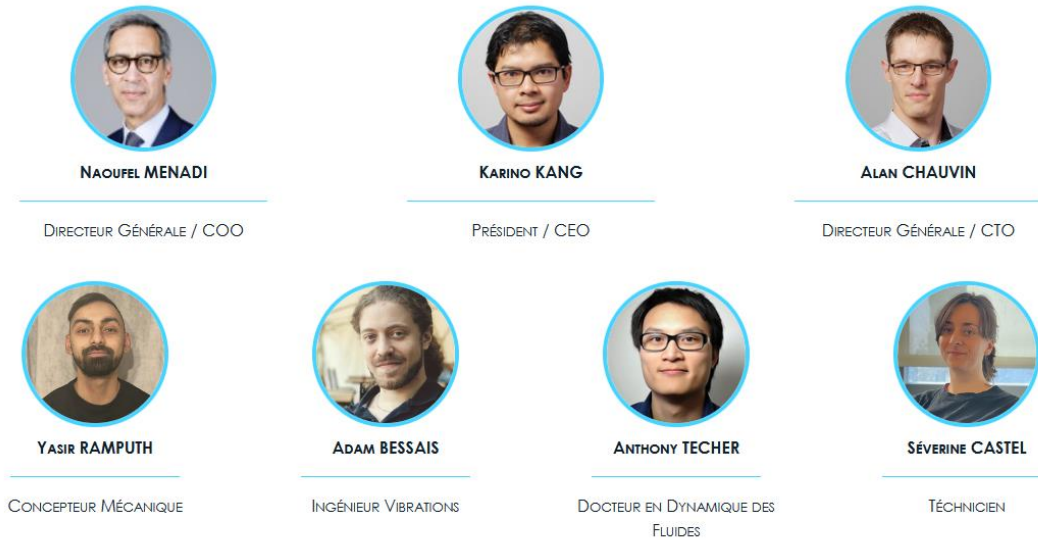
Mai 2021 : Levée de fond avec Happy Capital

Septembre 2021 : partenariat Time for the Planet

Janvier 2022 : Embauche d'un stagiaire prometteur

2.4. L'équipe

NOTRE EQUIPE



Lors de mon arrivée chez Léviathan Dynamics l'entreprise comptait 7 employés :

- Karino Kang, CEO
- ALAN Chauvin, CTO
- Naoufel Menadi, COO
- Yasir Ramputh, Concepteur Mécanique
- Adam Bessais, Ingénieur Vibrations
- Anthony Techer, Docteur en Dynamique des fluides
- Séverine Castel, Technicienne

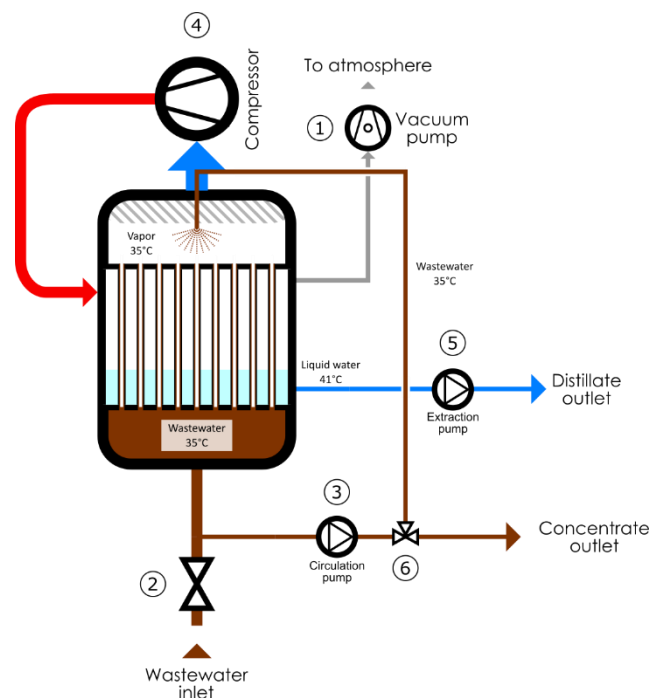
2.5. Les projets

Turbevap : Evapo-concentrateur pour le traitement des effluents industriels.

Turbevap est un projet de traitement des effluents industriels déjà commercialisé. L'entreprise travaille aujourd'hui à la conception physique des machines vendues et au développement de versions plus avancées.

Fonctionnement de l'évaporateur d'effluent à CMV :

1. Le système est mis sous vide poussé.
2. L'effluent est aspiré dans l'évaporateur.
3. L'effluent est circulé en utilisant une pompe de circulation. Il est pulvérisé en fines gouttes dans la partie supérieure de l'évaporateur.
4. Le compresseur est démarré. La dépression causée par le compresseur force l'eau à s'évaporer. La vapeur est alors comprimée pour être rejetée dans le condenseur.
5. La vapeur, en contact thermique avec l'effluent froid à travers les tubes de l'échangeur thermique, est condensée. Quand le niveau est suffisamment élevé, l'eau distillée, ou distillat, est extrait.
6. Dès qu'il n'y a plus d'eau à évaporer dans l'effluent, l'effluent concentré, ou concentrat, est extrait. Il peut alors être traité en utilisant les moyens usuels.



L'innovation réside dans l'amélioration de la construction traditionnelle des évaporateurs à CMV par l'intégration de notre technologie de compresseur. L'étanchéité parfaite de notre compresseur permet un fonctionnement sous vide poussé. Quand les évaporateurs à CMV traditionnels évaporent l'eau à 80°C, notre construction permet une évaporation à 35°C.

Technologie de chauffage et de refroidissement :

Les technologies de chauffage et de refroidissement couvrent un très large spectre d'équipements. En fonction du besoin, plusieurs solutions peuvent être proposées.

Le domaine d'expertise de Léviathan Dynamics concerne **les pompes à chaleur et les groupes froids.**

Pompe à chaleur

Une pompe à chaleur est un dispositif dont l'objectif principal est de transférer la chaleur présente dans un réservoir thermique, tel que l'environnement, à basse température vers un procédé qui nécessite un chauffage à plus haute température. La chaleur est transportée par un fluide caloporteur de la pompe à chaleur vers le procédé. L'utilisation principale des pompes à chaleur est le chauffage de confort. Cependant, les pompes à chaleur sont de plus en plus utilisées dans l'industrie où la chaleur est traditionnellement produite par des chaudières à combustible fossile.

Groupe froid

Un groupe froid est l'inverse d'une pompe à chaleur. Son objectif principal est de transférer la chaleur du procédé à une température relativement faible vers un réservoir thermique tel que l'environnement.

La chaleur est transportée par un fluide caloporteur du procédé vers le groupe froid. Les groupes froids sont essentiellement utilisés dans l'industrie pour refroidir les procédés. Ils sont aussi souvent utilisés pour le froid de confort lorsque la climatisation est centralisée.

Le projet serenade, est un projet de groupe froid. C'est le prochain projet que Léviathan Dynamics compte commercialiser. Un système de climatisation qui remplace les gaz fluorés (autant d'émissions que l'aviation mondiale) par de l'eau (quelques litres en circuit fermé) et consomme 30% moins d'électricité que les climatiseurs actuels.

Le projet de transformation des réfrigérants polluants actuellement utilisé dans la majorité des industries mondiales pour un réfrigérant à base d'eau est en fait le but de départ de l'entreprise Leviathan Dynamics.

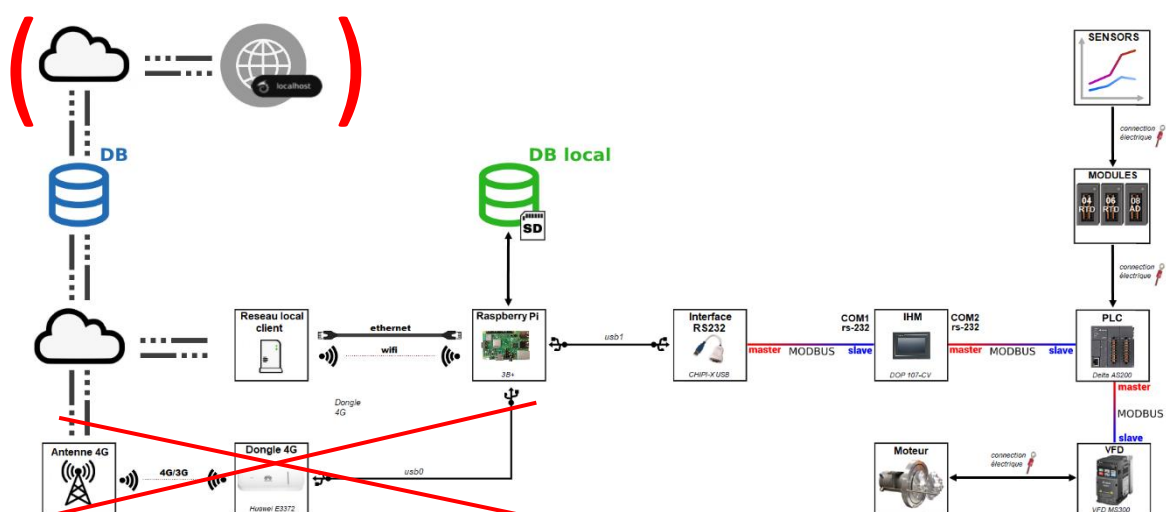
3. Ma position dans l'entreprise

3.1. Ma mission

Mon stage se déroule en tant que technicien informatique. Je m'occupe de la connectivité entre les machines et une base de données. Durant mon stage j'effectue ma mission tout en suivant les directives de Karino Kang, mon maître de stage.

Je travaille sur toutes les machines de l'entreprise car le programme d'envoi des données est commun pour toutes les machines.

Architecture à mon arrivée dans l'entreprise :



The diagram illustrates the data flow architecture for the Machine Leviathan Dynamics. It shows a central 'Serveur en ligne' (Online Server) connected to a 'DB' (Database) and a 'DB local' (Local Database). The 'DB' is connected to a 'cloud' icon, which is also connected to a 'localhost' icon. The 'DB local' is connected to a 'Raspberry Pi 3B+'. The 'Raspberry Pi 3B+' is connected to a 'Dongle 4G' (Huawei E3372) via a 'usb0' connection. The 'Dongle 4G' is connected to an 'Antenne 4G' via a '4G/3G' connection. The 'Antenne 4G' is connected to the 'Serveur en ligne' via a 'Décompression des données' (Data Decompression) process. The 'Raspberry Pi 3B+' is also connected to a 'DB local' via a 'Compression des données' (Data Compression) process. The 'Raspberry Pi 3B+' is connected to a 'Machine Leviathan Dynamics' via a 'usb1' connection. The 'Machine Leviathan Dynamics' is a white, rack-mounted unit with a display and buttons.

3.1.1. Module connectivité

L'algorithme d'envoi des données est programmé en python.

L'algorithme, codé avec Python⁽²⁾, en place à mon arrivée possédait trois parties majeures :

- 9

- La partie MAIN permettant de lancer un thread⁽³⁾ de récupération des données en les plaçant dans une queue et un autre thread permettant de vider la queue en envoyant les données si trouvant au serveur base de données.

J'ai d'abord pris le temps de comprendre en profondeur la partie MAIN et la partie envoi qui sont les deux parties dont j'allais être amené à modifier durant ma mission.

3.1.1.2. Communication entre programmes

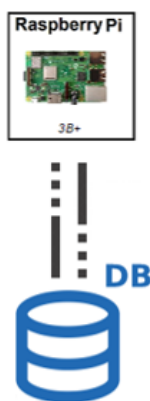
Mon objectif étant de limiter la taille des données envoyées depuis la machine il m'est tout de suite venu à l'idée de compresser les données que j'envoie. Néanmoins un serveur SQL⁽⁴⁾ n'est pas fait pour lire des messages compressés, il fallait donc que je crée un programme intermédiaire permettant de recevoir les données pour les retransmettre ensuite en clair au serveur SQL. Évidemment la première étape à mettre au clair avant de traiter les données était de savoir comment mes programmes allaient communiquer.

Étant une technique que je connaissais déjà et facile à mettre en place en Python, j'ai opté pour l'utilisation de socket.

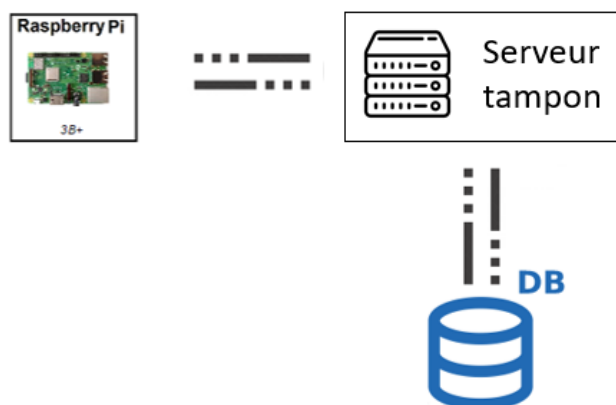
J'ai donc créé un programme serveur permettant de récupérer la requête SQL et l'envoyant au serveur BDD (base de données). De plus j'ai modifié la partie envoi du programme de la RPI (Raspberry PI) pour que la requête soit envoyée sous forme de 'string python' à mon serveur, et non au serveur BDD.

Cela mis en place je pouvais compresser mes données comme je voulais si je créais le programme inverse sur mon serveur tampon.

Envoi de la requête SQL complète au serveur BDD:



Envoi de la requête SQL transformé au serveur tampon pour déchiffrement et envoi au serveur BDD:



3.1.1.3. Compression des données

Tout d'abord pour limiter la taille des données envoyées il était évident qu'il fallait **supprimer la syntaxe imposée par MySQL**. Ainsi :

- "INSERT INTO machine1 (temp, pression, time) VALUES ('500','49.9','04/04/2022 12:12:12')"
 - Devenait : [1] * [500,49.9,04,04,2022,12,12,12]
- *Numéro de la machine envoyé une seule fois au début de la communication.

Nous pouvons voir que le nombre de données envoyé est nettement réduit.

Ensuite je me suis tournée vers des bibliothèques de compressions. Avec Pickle je pouvais réduire légèrement la taille de mes messages les plus longs mais les messages court étaient parfois plus longs qu'avant compression. J'ai compris que cela était dû au fait que chaque donnée possédait avant elle un code permettant de savoir la taille de la donnée suivante (en bit) et le type (string, int, float).

Sachant que le serveur pouvait connaître le type des données qui allait être transmis à partir du numéro de machine envoyé en début de communication je pouvais **simplifier l'algorithme de compression en retirant les indicateurs de types**.

En suivant les types de la base de données MySQL j'ai pu faire une table contenant le nom de la donnée son type et sa position dans la chaîne d'envoi. Avec cela je pouvais envoyer les données en limitant le nombre de bits utilisés en fonction du type de la donnée.

- [1] [500,49.9,04,04,2022,12,12,12] (16x[32bits])
- Devenait: [1] [16 bits] [32bits] [8bits] [8bits] [16 bits] [8bits] [8bits] [8bits]

(Par la suite j'ai remarqué que certaines données pouvaient dépasser la limite de 16bits des int, il a donc fallu passer les int sous 32bits.)

Après cela j'ai remarqué qu'une grande partie des données était identique d'un envoi à l'autre, j'ai donc ajouté un système permettant de **supprimer les données redondantes sur plusieurs envoies**.

- [500,49.9,04,04,2022,12,12,12], [500,49.8,04,04,2022,12,12,13]
 - Devenait : [1,1,1,1,1,1,1] * [500,49.9,04,04,2022,12,12,12], [0,1,0,0,0,0,1] [49.8,13]
- *Tableau de BOOL définissant si la donnée est envoyée ou non (le tableau prend 1bit par données max envoyé par la machine. Dans l'exemple : 8 données -> tableau de 8 bits sur tous les envois.)

En envoyant en amont un tableau informant le serveur que la donnée a changé ou non j'ai réduit drastiquement le nombre de données envoyées. Nous pouvons remarquer que le premier envoi contient toutes les données et que c'est seulement au deuxième envoi que la méthode est utile.

Tous ces systèmes compressant les données sont inversés sur le serveur tampon permettant de recréer la requête SQL de base à partir des données et des paramètres de la machine, connu par le serveur.

3.1.1.4. Communication avec le serveur

Jusqu'alors je faisais mes tests en local en mettant le serveur et le client sur la même machine pour simplifier le développement du module de compression. Néanmoins sur le terrain les machines n'hébergeront pas le serveur et seront de plus sur un réseau différent du serveur. Ne connaissant pas l'adresse IP global du serveur j'ai d'abord mis en place un système de tunneling⁽⁶⁾ ssh⁽⁵⁾ et de redirection de port permettant d'envoyer les données au serveur avec son nom de domaine et la clef ssh.

Cette solution n'étant pas viable, car le fait de fournir à tous nos clients la possibilité de se connecter par ssh à notre serveur est bien trop dangereux, je suis retourné sur la solution d'une simple connexion par sockets. Après quelques recherches j'ai pu obtenir l'adresse globale de notre serveur tampon et la communication était possible.

3.1.1.5. Sécurisation des communications

Un problème logique apparaissant lors de la création de communications entre différents réseaux est la sécurisation de cette communication. Pour la sécurité de la communication un tunnel ssh était une très bonne solution car les communications ssh sont, de base, chiffrés et donc illisible par un tiers néanmoins comme dit précédemment cette méthode nécessite de donner trop de pouvoir à chacun de nos clients.

La solution était simplement d'utiliser une librairie python permettant de transformer notre communication en clair par socket en une communication chiffrée grâce à TLS/SSL par socket.

Pour cela il fallait de plus fournir un certificat à notre serveur contenant une clef privée ainsi que de fournir la partie publique à notre client. Ainsi chaque client peut communiquer de manière chiffrée avec notre serveur tout en étant certain de bien s'adresser, sans détour, au serveur.

De plus j'ai ajouté une connexion par mot de passe pour le client permettant de limiter les machines sur lesquels il peut envoyer des données.

3.1.2. Bug fix module connectivité.

Durant l'analyse des données mon supérieur Karino Kang a remarqué que certaines données étaient enregistrées à la même date.

Les données enregistrées à la même date étaient par groupes de 40 (étendues sur environ 40 secondes) et l'espacement entre deux bugs était de plusieurs heures. J'ai remarqué que les données étaient à chaque fois datées à la même heure que le démarrage du logiciel (inscrit dans les logs).

Le problème se déroulait donc au démarrage. Grâce à quelques tests j'ai pu comprendre que dans son utilisation habituelle la Raspberry pi était éteinte en la coupant du courant. Cela entraînant des problèmes car l'horloge n'a pas le temps de sauvegarder la date et lors du prochain démarrage la date sera très en retard. De plus la RPI ne possédant pas d'horloge fonctionnant lorsqu'elle est éteinte même en éteignant la RPI proprement la date au démarrage serait la même qu'au dernier arrêt et donc tout de même en retard.

La solution mise en place fut donc de mettre à jour la date dans le logiciel avant toute autre action en la récupérant sur internet.

3.1.3. Module 4G

Le module 4G permet à la Raspberry pi de se connecter à internet sans utiliser le réseau du client. La possibilité de se connecter à internet sans utiliser un réseau WI-FI externe est un atout important du point de vue du client, grâce à cela aucun expert en sécurité réseaux n'a besoin de vérifier si la machine est dangereuse car elle ne peut pas l'être si elle ne fait pas partie du réseau.

Après analyse des différentes solutions permettant de donner à notre Raspberry pi la possibilité de se connecter aux réseaux 4G nous avons fait le choix d'utiliser un HAT pour Raspberry pi.

Un HAT est une carte électronique à brancher directement sur la Raspberry pi, les HAT peuvent avoir de nombreuses utilités et peuvent se mettre en série sur la Raspberry pi.

3.1.3.1. SixFab

Le hat 4G proposé par SixFab est fourni avec un kit contenant une antenne pour les réseaux cellulaires / 4G, une antenne GPS et une carte Sim.

Il est normalement très simple de mettre en marche le module. Après avoir branché les composants entre eux et le HAT sur la Raspberry pi il faut simplement exécutée une commande d'installation sur la RPI. Quand l'installation est terminée un redémarrage est exécuté et la connexion 4G est disponible.

Lors de mon installation j'ai eu un problème de configuration sur ma carte RPI ce qui à rallonger anormalement l'installation du module et a nécessité l'aide du support SixFab.

3.1.3.2. Waveshare

Waveshare est un autre constructeur de hat pour Raspberry pi, il propose des solutions plus spécifiques et moins cher. Nous avons testé deux hats, un 2G et un 3G/4G.

L'installation de ces modules commence aussi par le branchement des différentes antennes au hats et à la RPI puis il est nécessaire de faire une installation plus longue pour rendre le réseau cellulaire utilisable.

Après l'installation du module 4G il peut être remplacé par le 2G sans installation supplémentaire.

3.1.3.3. Test

Pour les tests j'ai configuré deux cartes SD, une pour chaque type de hat, puis j'ai vérifié que la communication fonctionne en utilisant seulement la connexion cellulaire. J'ai aussi pu voir que lors d'une perte de connexion avec le serveur la communication reprend sans problème dès que le serveur est à nouveau disponible.

3.1.3.4. Conclusion

Pour la communication 4G la solution la plus facile à mettre en place et a industrialisée est le hat de SixFab qui ne demande que le branchement et l'exécution d'une seule ligne de configuration pour être mis en service.

Le hat Waveshare a tout de même l'avantage d'être moins chère (2x moins pour le 4G et 4x moins pour le 2G) mais son temps d'installation est trop long pour être une solution dans l'état.

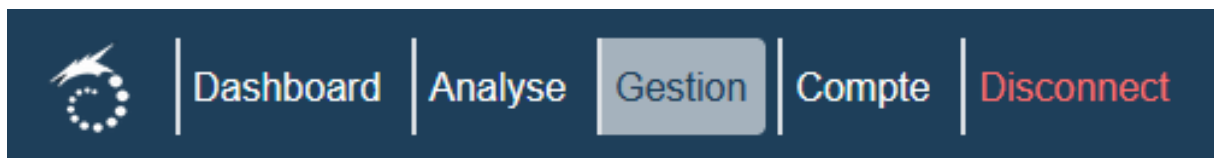
3.1.4. Dashboard web

Le Dashboard web est un site disponible pour les clients permettant de vérifier le bon fonctionnement des machines. Le site doit donc posséder un système d'authentification et des pages de suivi avec des graphiques et les différentes données les plus importantes accessibles rapidement.

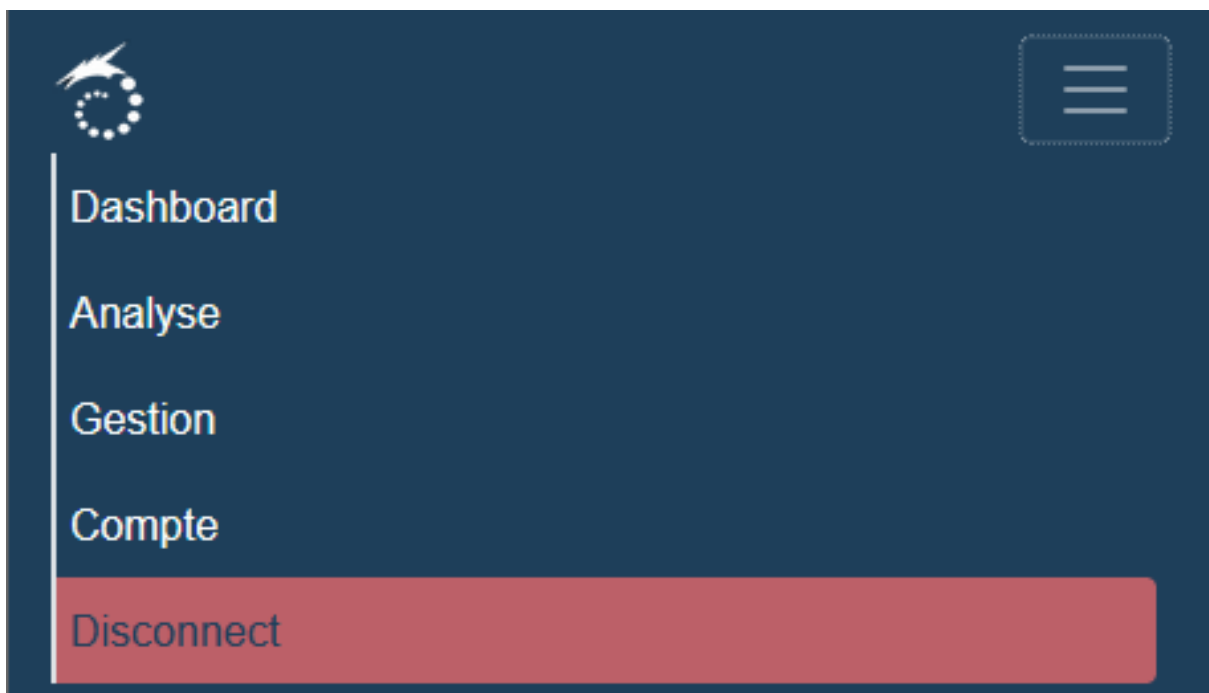
L'ensemble du site est programmé en HTML, CSS, JS⁽⁷⁾ et PHP⁽⁸⁾. Il communique avec une base de données stockée sur un autre serveur qui contient les utilisateurs, leurs paramètres et les données des machines.

3.1.4.1. Navbar

Le site est composé d'une navbar permettant de naviguer parmi les différentes pages. La navbar évolue en fonction des droits de l'utilisateur connecté.

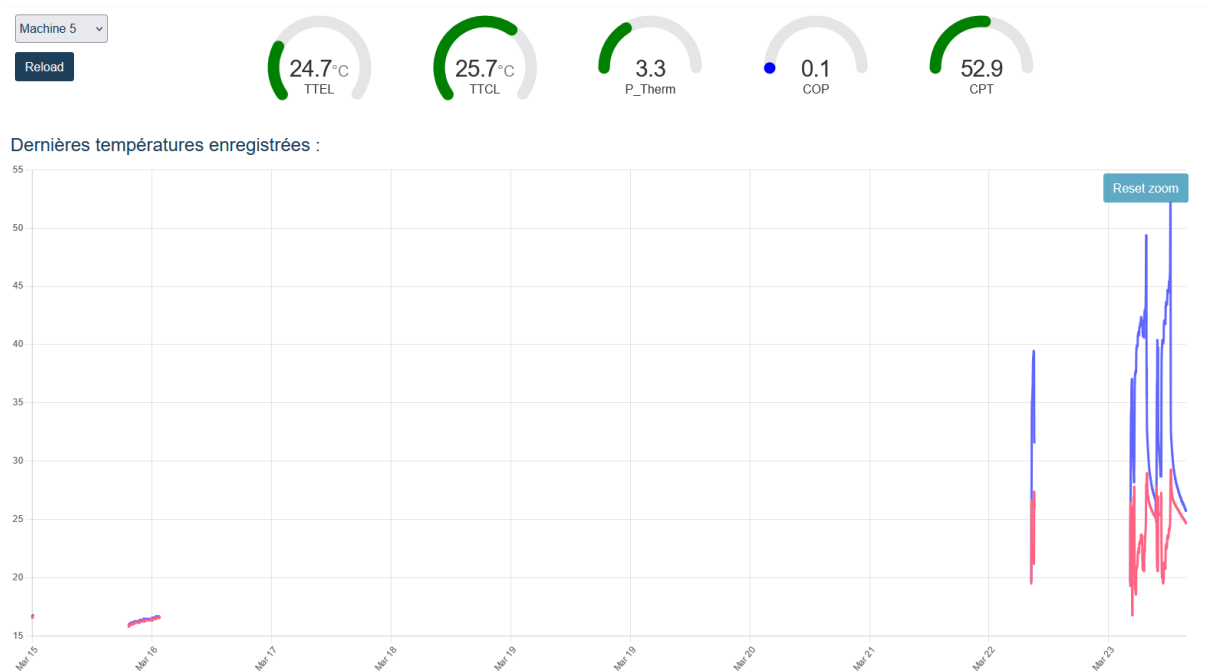


Faite avec Bootstrap celle-ci est responsive et se réduit pour les petits écrans.



3.1.4.2. Page Dashboard

La page Dashboard est la page principale du site web, sur cette page le client peut voir les données les plus importantes de ses machines et un graphique des températures sur les dernières semaines.



Pour créer des jauges j'ai utilisé Gaugemeter.js et pour le graphique j'ai utilisé chart.js, deux librairies open source.

Les jauges

Les jauges permettent de voir rapidement la valeur de certaines mesures particulièrement importantes, elles permettent aussi de comprendre en 1 coup d'œil si la valeur se rapporte à un fonctionnement classique ou non.

Les jauges sont créées à une taille précise en fonction de la taille de l'écran pour permettre une meilleure lisibilité sur des petits écrans.

Le graphique

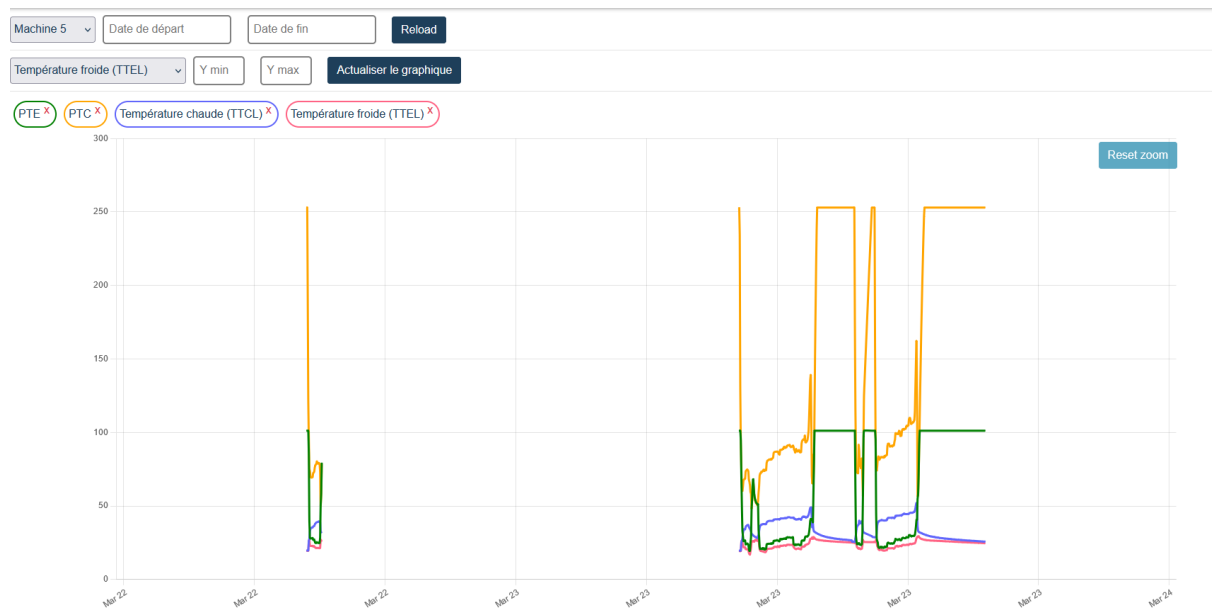
Le choix de chart.js a été favorisé car cette librairie permet de créer différents graphiques responsifs et personnalisables. L'atout majeur de cette librairie est qu'elle n'est ni trop grande ni trop complexe et est gratuite.

Sur tous les graphiques j'ai ajouté l'option d'avoir un axe X linéaire en fonction de la date et l'option permettant de zoomer.

De plus j'ai fait en sorte de couper les courbes lorsque des données sont espacées de plus d'une heure pour faciliter la lecture.

3.1.4.3. Page Analyse

La page analyse est la page la plus complexe, elle permet de voir toutes les données de ses machines et de les comparer sur un graphique. Il est possible de changer les bornes des axes X et Y.



3.1.4.4. Page Gestion

La page gestion montre l'ensemble des machines et des utilisateurs enregistrés sur le site et permet d'en ajouter, d'en supprimer et de modifier les existant en fonction des droits de l'utilisateur connecté. Cette page est disponible uniquement pour les utilisateurs avec le rôle Admin ou super Admin.

User

Search

Créer un nouvel utilisateur.

Ajouter une machine.

admin : Super Admin

Machines liés : 5 4 3 8

Password1 : User

Machines liés : 3 5

Tom : Admin

Machines liés : 4

Machine

Search

Créer un nouvel utilisateur.

Ajouter une machine.

n° : 3

Utilisateurs liés : admin Password1

n° : 4

Utilisateurs liés : admin Tom

n° : 5

Utilisateurs liés : admin Password1

n° : 8

Utilisateurs liés : admin

3.2. Les problèmes / surprises

Durant mon stage de nombreux problèmes et obstacle dans le développement mon permis d'en apprendre plus sur les outils que j'utilisais. Dans cette partie je vais présenter les différents problèmes que j'ai rencontrés, leur résolution et leurs apports.

3.2.1. Test

Le premier problème que j'ai rencontré est apparu après que j'ai modifié le programme actif sur la machine de test.

Pour résoudre un bug enregistrant des données à une mauvaise date j'avais ajouté une étape au démarrage du programme. Après avoir vérifié que mon ajout marche je l'ai mis sur la machine. Néanmoins lorsque le programme était démarré après un shut down forcé l'ajout faisait planter l'algorithme et aucune donnée n'était envoyée. Je n'avais pas assez testé ma modification et cela a entraîné une perte de données sur une demi-journée. Bien que résoudre le problème était simple et rapide j'ai appris qu'il était primordial de tester ses modifications, surtout lorsqu'on ne travaille pas dans une simulation.

3.2.1. Choix

Le deuxième problème que j'ai rencontré est un problème dû à la liberté qui m'était accordée.

L'entreprise me laissant le choix des technologies, langages et librairies que j'utilisais j'ai eu de nombreuses fois à choisir quoi utiliser pour faire ma tâche. Le choix le plus long que j'ai eu à faire fut sur les graphiques du site web. De nombreuses librairies permettent de dessiner des graphiques.

J'ai d'abord essayé plot.js, un Framework répandu pour cette tâche. Après avoir réussi à dessiner mes premiers graphiques j'ai tout de suite trouvé les résultats très laids.

J'ai donc analysé les différents Framework dérivé de jqplot.js. Trois Framework sont sorti du lot : Highcharts, D3.js et chart.js.

Highcharts étant très cher et D3 trop complexe je les ai abandonnés. J'ai finalement choisi chart.js car celui-ci est facile, beau, flexible et gratuit.

Ce genre de choix sur les outils que j'allais utiliser est souvent apparu et m'ont fait comprendre l'intérêt d'être libre sur la technologie qu'on utilise mais aussi le temps nécessaire à trouver les outils qui conviennent bien.

3.2.2. SQL

Lors de la création des graphiques sur le site internet j'ai rapidement été confronté à des problèmes dûs à la quantité de données à afficher. Sur un graphique, chaque donnée étant un point avec des événements au survol, les graphiques avec un très grand nombre de données ralentissaient le navigateur.

J'ai d'abord choisi de limiter le nombre de données afficher à 1000 en les triant avec Javascript après les avoir reçus. Pour limiter les données à 1000 en gardant des données représentatives j'ai suivi la procédure suivante :

- Diviser le nombre de données en entrée par 1000 pour obtenir 'ratio' (la distance entre 2 données).
- Prendre toutes les données séparées par le ratio et supprimer les autres.
 - o `If (i % data/1000 == 0) {result.push(dataResult[i]) }`

Cette méthode fonctionne, néanmoins j'envoie un nombre de données pouvant être extrêmement élevé pour n'en utiliser que 1000. J'ai donc essayé de faire le tri du côté du serveur SQL. Cela grâce à une requête SQL permettant de récupérer 1000 données sur une plage de temps donné :

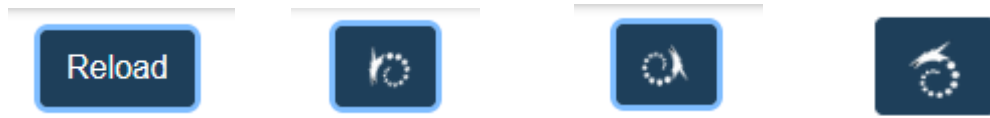
- D'abord j'initialise une variable @row_num à -1 pour numéroter mes lignes différemment de l'ID : `'SET @row_num = -1;'`
- Ensuite je trie les données en fonction de l'intervalle de temps demandé et les range dans une table temporaire :
`'CREATE TEMPORARY TABLE temp_time_machine SELECT (@row_num : =@row_num +1) AS rowNum, ID FROM machine5 WHERE id_machine = (?) AND TIME > (?) AND TIME < (?) ORDER by TIME desc;'`
- Je récupère seulement 1000 données de la même manière qu'avec javascript :
`'SELECT * FROM machine5 WHERE ID IN (SELECT ID FROM temp_time_machine WHERE rowNum%@ratio = 0) ;'`
- Pour finir je supprime la table temporaire : `'DROP TEMPORARY TABLE temp_time_machine;'`

De cette manière je n'envoie que les données que je vais afficher à l'utilisateur. Le problème avec cette solution est que sur de très grandes tables le temps de calcul pour sélectionner les données peut prendre plusieurs secondes. La solution mise en place pour faire attendre l'utilisateur est un écran de chargement.

3.2.3. Design

Le design fut un problème important lors de la conception du site web. D'abord il a fallu trouver de jolis graphiques et après il a fallu que je design le site web pour qu'il soit beau et intuitif. N'étant pas très bon en arts la plupart des choix que je faisais n'était pas convaincant. Pour réussir à faire un joli site j'ai été chercher des designs dans les sites web que j'avais faits précédemment et sur le site web de Léviathan Dynamics. De plus j'ai utilisé Bootstrap pour simplifier et harmoniser le design des objets communs (bouton, formulaire).

Ma meilleure réalisation en design est sûrement à attribuer au bouton pour recharger les données qui affiche le logo de Leviathan Dynamics en rotation durant la récupération des nouvelles données :



3.3. Dans une start-up

Grâce à ce stage j'ai pu découvrir l'univers des start-up. Le travail dans une start-up m'a permis de développer mes compétences dans de nombreux domaines tout en apprenant chacun des problèmes survenant tout au long d'un développement. J'ai apprécié la satisfaction de faire un projet seul en comprenant l'ensemble des éléments, des fonctionnalités et des modules résultants.

J'ai aussi apprécié la proximité des différents métiers de l'entreprise, permettant de découvrir des problèmes qui ne nous touchent pas mais que l'on peut aider à résoudre et permettant de comprendre encore plus profondément le projet global de l'entreprise.

3.4. Apport du stage

Durant mon stage j'ai pu développer majoritairement mes compétences techniques en programmation web et python. De plus j'ai appris de nouvelles notions de sécurité réseau en les utilisant en situation réelle et concrétisé mes connaissances en SQL.

Le fait d'avoir travaillé sur un projet dans sa globalité en touchant à tous les domaines connexes a été le plus formateur durant mon stage. J'ai ici appris à comprendre tous les enjeux de chacune de mes décisions et les liaisons entre les différents domaines informatiques.

Du côté des soft-skills j'ai appris à écouter les retours de mon tuteur de stage ainsi que de mes collègues pour améliorer les designs et l'ergonomie de mon travail.

4. La suite du stage et l'après

Durant les deux dernières semaines de stage je vais tester les différentes fonctionnalités des rendus pour vérifier leur bon fonctionnement ensemble et pour régler les différents bugs qui pourraient survenir.

Je vais aussi ajouter des fonctionnalités sur le logiciel de traitement de données et sur le site internet. Le prochain ajout sera un système d'alerte par mail lors du dépassement des seuils sur certaines données.

Après mon stage le programme python sera utilisé sur les futures machines. Le site web est déjà en ligne et sera utilisé par les clients et par l'équipe Léviathan pour obtenir rapidement un bilan sur les machines en service. Une modification de la base de données sera nécessaire pour regrouper l'ensemble des machines dans la même table et il faudra donc modifier la configuration du serveur tampon et du site.

5. Conclusion

Pour conclure, ce stage de 15 semaines chez Leviathan dynamics m'a permis de découvrir le monde du travail dans une start-up. Améliorant mon autonomie et mes compétences dans différents domaines en les appliquant à un projet concret et motivant car proche de mes valeurs. J'ai découvert ici l'apport de travailler pour un projet m'intéressant et pour lequel j'ai vraiment envie de travailler. J'ai travaillé chez Léviathan Dynamics en voulant faire avancer le projet et non seulement en voulant développer mes compétences. Je remercie encore Karino Kang et toute l'équipe de Leviathan Dynamics pour leur aide et leur travail permettant à cette entreprise d'exister, je les remercie aussi de m'avoir choisi pour y participer.

6. Glossaire

CMV⁽¹⁾ (anglais pour VMC) : système de ventilation mécanique contrôlée, incluse au bâtiment.

Python⁽²⁾ : Langage de programmation orienté objet.

Thread⁽³⁾ : partie d'un programme informatique exécuté en parallèle d'autres processus.

SQL⁽⁴⁾ : langage informatique normalisé servant à exploiter des bases de données.

SSH⁽⁵⁾ : protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite toute la communication est chiffrée.

Html, CSS, JS⁽⁷⁾ : Langage de programmation WEB, frontend⁽⁹⁾.

PHP⁽⁸⁾ : Langage de programmation, backend⁽¹⁰⁾.

Frontend⁽⁹⁾ : Partie d'un programme informatique responsable uniquement de l'interface utilisateur, permettant à l'utilisateur d'interagir avec un backend.

Backend⁽¹⁰⁾ : Partie d'un site web invisible par l'utilisateur, souvent lié à la base de données.

7. Annexe

Site web: <https://monitoring.leviathan-dynamics.com/>