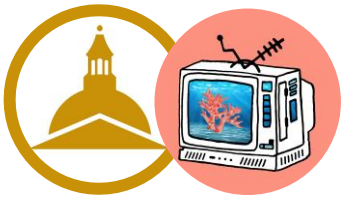


OceanStream et la MIAGE Sorbonne s'associent pour créer

L'OceanBox

Un projet AGILE et DevOps écoconçu

PARTIE ANALYSE

Abdel Benamara | Julien Doujet | David Ekchajzer | Mathieu Ridet

20/01/2020

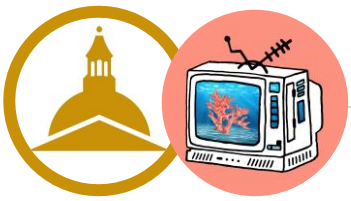


Table des matières

Introduction

On dit souvent que les océans sont les poumons de la terre. En effet ils produisent à eux seuls la moitié de l'oxygène que nous respirons[1]. Les écosystèmes marins sont aujourd'hui au cœur des problématiques environnementales dues au réchauffement climatique et aux pollutions humaines comme les déchets plastiques. Animées par les recherches scientifiques, notamment celles de l'ONG TARA, de nombreuses associations essaient de faire prendre conscience de l'importance pour la planète de préserver ces écosystèmes.

OceanStream est une association loi 1901 fondée en 2018 part des amoureux des milieux marins. Leur objectif est d'éduquer la population à la complexité de l'équilibre des Océans, mers et cours d'eau. Ils souhaitent que les populations qui ne sont pas en contact avec ces milieux, puissent se rendre compte de leur beauté et de leur fragilité. Pour ce faire, ils conduisent des projets de médiation scientifique, culturelles et artistiques à destination de différents types de populations. Un de leurs projets majeurs vise à utiliser la vidéo pour que la mer devienne un environnement familier en diffusant, par exemple, des images de la grande barrière de corail. OceanStream souhaite s'inscrire dans le mouvement de la slow TV qui a pour objectif de montrer des plans fixes et lents pour s'adapter à la vitesse lente de la nature.

Certains des membres d'OceanStream sont des anciens étudiants de Paris 1. Ils ont pu s'appuyer sur le soutien de La Sorbonne pour développer leurs projets. C'est donc tout naturellement que le président de l'association : Adrien Landa s'est tourné vers la MIAGE Sorbonne pour développer un POC de boitier électronique visant à diffuser le flux vidéo généré par leurs caméras.

Nous avons de suite adhéré au projet. Celui-ci nous permettant de travailler en collaboration avec une association sur un projet innovant. C'est une opportunité unique qui ne se représentera pas forcement dans notre vie professionnelle, et plus spécialement sur des questions d'écologie désespérément d'actualité. Constatant que le numérique a un impact très important et grandissant sur l'environnement, ce projet était pour nous l'occasion d'apprendre à développer des projets informatiques en y intégrant l'écoconception. Les compétences que nous acquerrons au cours de ce projet pourront potentiellement nous permettre de faire évoluer les pratiques dans les entreprises où nous travaillerons plus tard. Notre deuxième motivation réside dans les problématiques techniques auxquelles nous confronte le projet. Nous pourrons utiliser et développer des compétences en informatique hardware, programmation système, réseaux, administration serveur, etc…

1

Préanalyse

Il existe aujourd'hui une très grande communauté qui s'articule autour du streaming de flux provenant de webcam à différents endroits dans le monde. Les spectateurs accèdent à des flux depuis des sites spécialisés, comme EarthCam[2] par exemple, et peuvent échanger via une section « commentaire ». Certains de ces sites proposent des vidéos provenant de milieux marins.

Cette implémentation pose plusieurs limites. Premièrement le récepteur de la vidéo doit connaître ces sites et savoir chercher parmi des millions de flux webcam. Il doit donc faire la démarche d'aller vers cette vidéo. Cette démarche est réalisée par des individus déjà très mobilisés et informés sur la question d'écologie en milieu marin. De plus, ces sites sont réservés à une communauté très spécifique qui ont ce hobby et n'est donc pas adapté à la médiation scientifique qui doit viser la population la plus diversifiée possible.

Créer un objet, une OceanBox, est un moyen de rendre ce service plus concret et plus quotidien. Cela permettrait aussi plus facilement de monétiser le service fourni par l'association qui se servira de l'argent pour leurs activités de dépollution marine. La box peut même devenir un objet de décoration. Si le cahier des charges est respecté, la solution sera Plug and Play et facilitera l'utilisation pour toutes les personnes peu importe leurs connaissances en informatique.

On peut également considérer les aquariums comme un existant. En effet, ils permettent d'avoir un contact avec un milieu aquatique au quotidien mais ils nécessitent un entretien, et ne sont pas adaptés aux espaces publiques. De plus ils donnent l'image d'une nature contrôlée par l'homme alors qu'OceanStream souhaite montrer la réalité des espaces marins.

L'OceanBox pourra être implémentée facilement dans les espaces publiques pour remplacer totalement ou partiellement la publicité. Enfin, elle permettra de rapprocher les récepteurs de lieux très exotiques comme la grande barrière de corail.

Nous avons tout de même trouvé des limites à la création de l'OceanBox qu'il convient d'expliciter. Son utilisation va utiliser de l'énergie pour filmer, transmettre et diffuser le flux sans compter l'énergie et les ressources nécessaires à la production du boitier en lui-même. C'est pourquoi nous avons décidé d'essayer de réduire au maximum l'impact environnemental de l'OceanBox. Par rapport à nos choix qui, en plus des justifications techniques et financières, devront être justifiés écologiquement. C'est ainsi que nous souhaitons ce produit Eco-conçu. Le Maître d'ouvrage et le président d'OceanStream, Adrien Landa, a validé la proposition car il partage avec nous l'envie de réduire l'impact des projets numériques sur la planète et spécialement ceux de son association.

Après avoir parlé avec Adrien Landa ainsi qu'à des clients potentiels particuliers ou entrepreneurs de notre entourage, nous avons identifié plusieurs personas[3] qui nous permettront d'adapter notre projet aux différents profils d'utilisateurs amenés à utiliser le produit :

Le client/receveur particulier

[2] https://www.earthcam.com/

[3] En UX design, un persona est une personne type qui représente une catégorie d'utilisateur

Il achète le boîtier pour soutenir les actions de l'association. Il a simplement à le brancher à sa télévision (même si elle n'est pas connectée) pour recevoir et retransmettre le flux vidéo proposé par OceanStream en continue. Il ne veut pas se soucier de devoir l'éteindre et la rallumer régulièrement mais souhaite économiser le plus d'énergie possible. Le boitier reconnaît quand quelqu'un est présent grâce au capteur de mouvement et s'allume.

Le client professionnel

Le client professionnel souhaite soutenir les actions de l'association et le faire savoir. Il utilise le boîtier pour streamer sur des écrans dans les lieux qu'il détient (salle d'attentes, gares, magasins, salons, bureaux, évènements…). L'installation et la désinstallation de la box doivent être facile s'il souhaite la changer régulièrement d'emplacement et/ou en installer plusieurs.

Le receveur en lieu public

Le receveur dans un espace public a accès aux vidéos ponctuellement ou quotidiennement si la box diffuse sur son lieu de travail ou sur un chemin emprunté régulièrement comme une gare ou une station de métro. Il peut découvrir les actions d'OceanStream par ce biais, soutenir leurs actions et éventuellement acheter sa propre box.

L'équipe de développement

L'équipe de développement souhaite que les mises à jour des box soient faciles et ne nécessitent pas de rapatrier toutes les box pour les mettre manuellement à jour. C'est pourquoi elle souhaite qu'une chaine DevOps soit mise en place. Elle est constituée de bénévoles. Elle souhaite que les flux soient automatisés pour ne pas avoir à gérer quotidiennement l'administration des box.

Il est important pour nous de définir les persona car nous avons opté pour une méthode de développement projet proche des utilisateurs.

Méthodologie Projet

Un projet AGILE

Nous avons choisi la méthodologie de développement projet AGILE. Elle repose sur quatre principes fondamentaux énoncés par les dix-sept experts du développement de projets informatiques rédacteurs du « manifeste agile » :

« Nous découvrons de meilleures approches du développement logiciel en le pratiquant et en aidant les autres à le pratiquer. Ce travail nous a amené à accorder de l'importance :

- Aux individus et leurs interactions plutôt qu'aux processus et aux outils ;
- À un logiciel fonctionnel plutôt qu'à une documentation exhaustive ;
- À la collaboration avec les clients plutôt qu'à la négociation contractuelle ;
- À l'adaptation au changement plutôt qu'à l'exécution d'un plan. »[4]

Les fonctionnalités que nous allons développer devront être compréhensibles par les utilisateurs pour qu'ils puissent nous faire un retour éclairé. Le découpage des fonctionnalités sera donc fait en priorité en fonction des utilisateurs et non en fonction des implémentations techniques. OceanStream et ses membres seront pour nous d'abord des collaborateurs avant d'être des clients ce qui, dans la situation contractuelle où nous sommes, est très facile à concevoir. Enfin, la méthode AGILE nous donnera une souplesse dans le développement. En effet, certaines fonctionnalités (User-Stories), ou implémentations de ces fonctionnalités, pourront être ajoutées, enlevées ou modifiées en cours de projet en fonction des retours utilisateurs ou de problématiques techniques. Cela est un réel avantage puisque notre faible expérience pourrait nous mener vers de mauvais choix. Ainsi, nous pourrons capitaliser plus facilement sur nos échecs. Enfin, cela nous permettra de continuer à développer des fonctionnalités tant qu'il nous reste du temps.

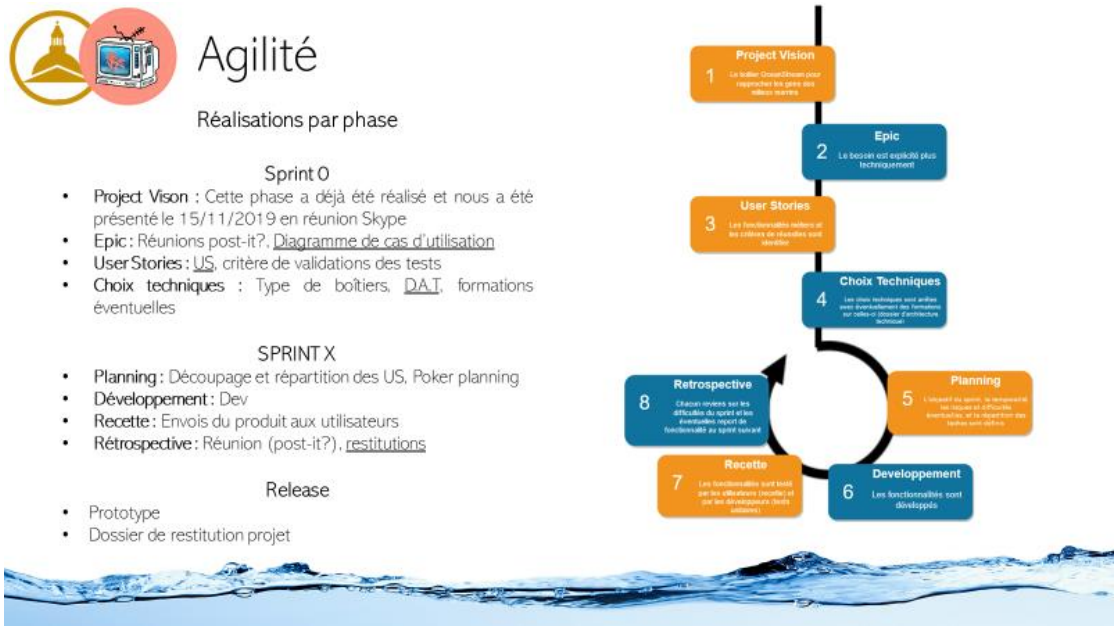
Nous allons adapter ces principes à notre projet de petite taille. Nous allons utiliser l'implémentation Scrum en utilisant la plupart de ses principes en nous accordant une certaine liberté pour ne pas nous enfermer dans une méthodologie trop lourde par rapport à l'équipe réduite et peu diversifiée qu'est la nôtre. Nous allons découper notre projet en plusieurs sprints correspondant à plusieurs fonctionnalités.

Notre workflow pour un sprint est le suivant :

- Ecriture des User-Stories : En collaboration avec la MOA/le PO (Maîtrise d'ouvrage/Product Owner), nous décrivons une fonctionnalité du point de vue utilisateur. Un résumé de l'US est fait sous la forme « je fais…grâce à…pour… ». Les critères de validation sont également énoncés du point de vue utilisateur. Cela nous aidera dans notre développement de tests.
- Sprint Backlog : Nous choisissons les US à développer lors de ce sprint en prenant en compte les dépendances entre les US et la volonté du PO.
- Poker Planning : Pour nous répartir les US, nous organiserons un poker planning à chaque sprint. Cette réunion consiste à noter la difficulté technique d'implémentation d'une US part des notes suivant la suite de Fibonacci. Ainsi, nous nous mettons d'accord sur les difficultés que nous pourrions rencontrées et nous nous répartissons équitablement le développement des US. Enfin, nous espérons que le PO pourra être présent à ces réunions pour qu'il se rende compte de la difficulté de développement des différentes fonctionnalités même s'il n'a pas de profil technique. Il comprendra ainsi mieux les délais et les potentiels échecs.

[4] Extrait du « Manifeste pour le développement Agile de logiciels - 2001

- Phase de développement : Nous développons les fonctionnalités en suivant les instructions de l'US avec ses tests associés. Si besoin, nous nous formons à des compétences nécessaires au développement de l'US.
- Test Utilisateur : Nous aurons une box chez un utilisateur pour qu'il puisse nous faire des retours sur les nouvelles fonctionnalités.
- Rétrospectives : Nous revenons avec toute l'équipe sur les résultats du sprint, les difficultés rencontrées et les améliorations à apporter au sprint suivant.



Dans la méthode Scrum, tout le monde est au même niveau d'égalité. Ceci est une bonne chose car tout le monde peut intervenir et proposer des améliorations. Dans notre cas, nous avions prévu dès le début de ne pas instaurer de hiérarchie dans notre groupe. Ainsi, nous aurons plus de fluidité. Cependant tout le monde doit être responsable pour que cela fonctionne. C'est pourquoi Scrum définit des rôles avec des prérogatives précises que nous nous sommes appropriés :

- Product Owner : Dans notre cas il s'agit également de la maîtrise d'ouvrage. Il s'agit de notre contact à OceanStream : Adrien Landa. Il a pour rôle de représenter les commanditaires du projet. Il sera chargé, en collaboration avec nous, de définir les User-stories. Il gérera également la partie budgétaire.
- Utilisateur : Il est l'élément central de notre projet. Il doit tester les fonctionnalités à la fin de chaque sprint et faire ses retours au PO.

- Scrum master : Il a pour rôle d'organiser les Sprint, de rappeler les risques et est garant de l'harmonie du groupe. Il aura la charge d'administrer le tableau Trello que nous avons mis en place pour représenter et répartir les User-Stories.
- Maitrise d'œuvre : Il est le contact privilégié du PO/MOA. Il a pour rôle de justifier les choix techniques et d'organiser les modélisations du système.

Un projet DevOps

Le mot DevOps vient de la contraction entre Développement et Opérations. C'est un mouvement visant à rapprocher, voire fusionner, le développement avec la production. Cela passe par la fusion partielle ou totale des équipes, des environnements, des workflows et des objectifs de développement et des opérations de production. Cette méthode permet une mise en production très rapide et se marie très bien avec la méthode AGILE. Elle part du principe qu'une application peut être testée en intégration pendant des mois mais rien ne vaudra un test grandeur nature avec un déploiement et des correctifs rapides à mettre en œuvre à posteriori. Pour ce faire, une chaine DevOps industrialisée doit être mise en place depuis le développeur jusqu'aux utilisateurs sans ou avec peu d'interventions humaines entre les deux. Ainsi, un correctif mineur peut être effectué en quelques minutes/heures.

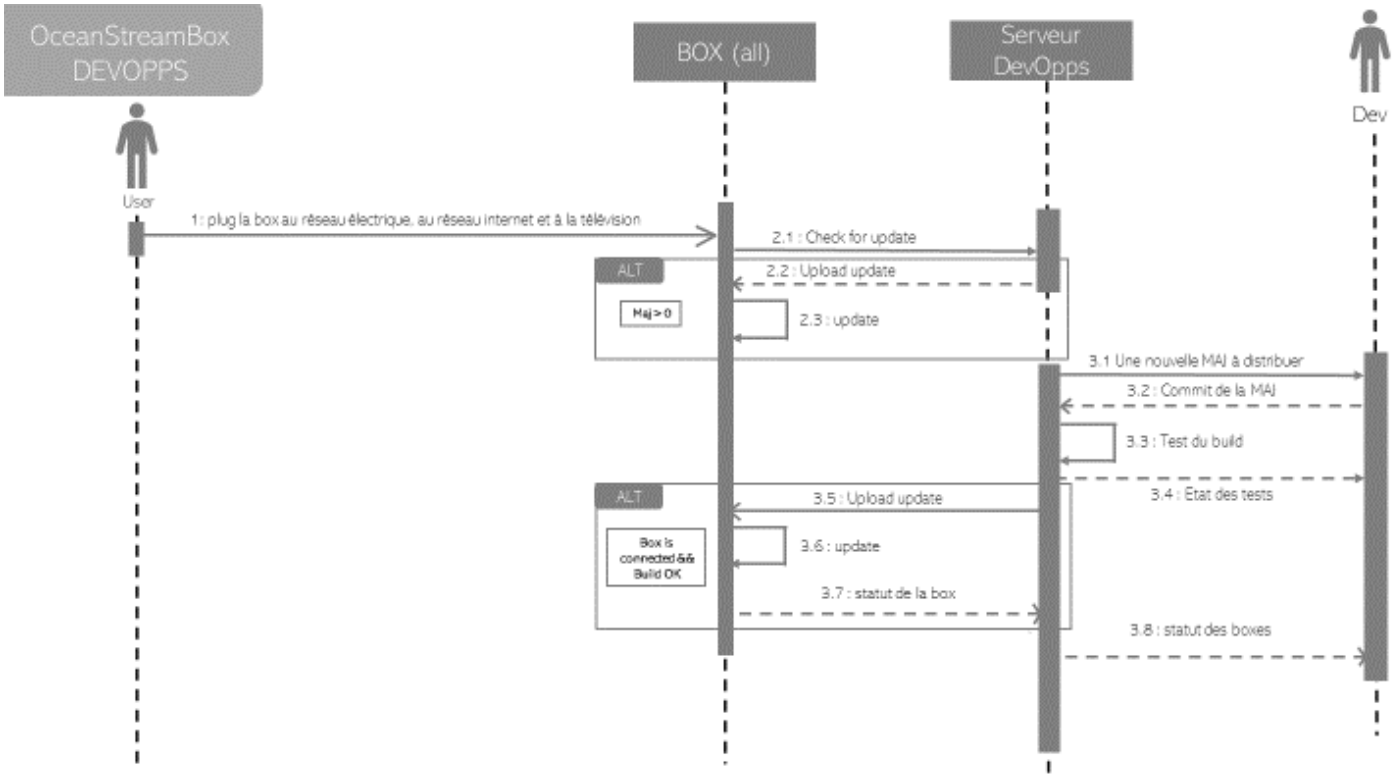
Cette méthode est intéressante car elle nous permettra, une fois la chaine DevOps mise en place, de développer des micro-fonctionnalités et de la mettre en production facilement.

Le Chief DevOps Officer aura pour rôle d'administrer cette chaine DevOps ainsi que la partie production du projet (numéros de release, retours de bugs, serveurs de production…)

Nous allons utiliser GitHub et sa capacité à faire des Workflow pour monter notre chaîne DevOps. Il y a deux manières pour une box de se mettre à jour :

- Quand la box est branchée, elle recherche les dernières mises à jour via un « *git pull* » du clone du répertoire de production qui est également le répertoire de l'application sur les box.

- Quand un développeur commit une mise à jour, elle est pull par le serveur DevOps (une box avec une architecture comparable aux box de production). Le serveur DevOps lance une batterie de tests. S'ils sont concluants, il ordonne via SSH à toutes les box de *pull* les dernières mises à jour.

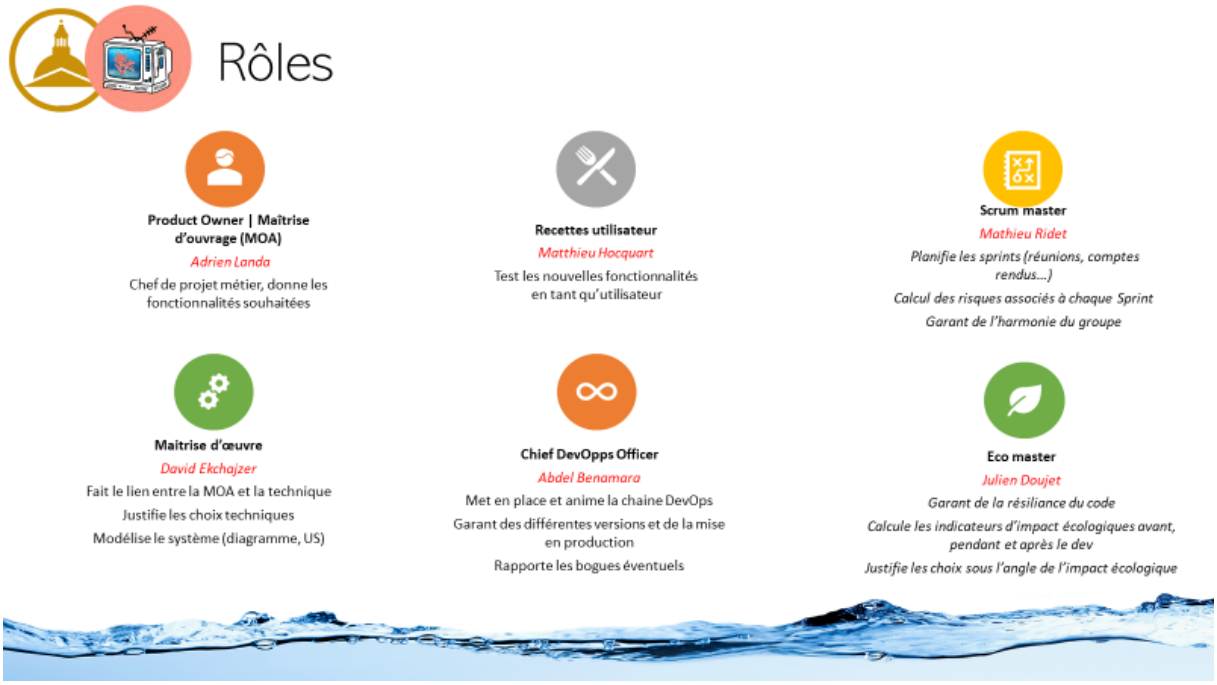


[5] https://www.greenpeace.fr/la-pollution-numerique/

Répartition des taches

Même si chacun aura un rôle défini, les prérogatives pourront être partagées mais devront être déléguées par la personne responsable du pôle concernée pour que chacun se sente responsable d'une partie du projet.

- Adrien Landa en tant que commanditaire du projet sera PO/MOA.
- Mathieu Hocquart, membre d'OceanStream, sera notre utilisateur. Il aura une carte connectée à notre chaine DevOps chez lui. Il a l'avantage d'être freelance en informatique et sera capable d'effectuer des manipulations si nous lui demandons. Par ailleurs ses retours auront l'avantage de pouvoir être techniques.
- Mathieu Ridet sera Scrum Master.
- David Ekchajzer sera MOE.
- Abdel Benamara sera Chief DevOps Officer.
- Julien Doujet sera Eco master. Ayant une formation de biologiste, il pourra plus facilement lire et comprendre les données et articles environnementaux.
- Nous nous partagerons tous les quatre la partie développement.



Rendu attendu

Comme vu avec Adrien Landa, nous souhaitons lui rendre un projet qui pourra ensuite être facilement repris par une autre équipe. Nous avons donc en tête que notre rendu doit être assez précis avec de la documentation pour assurer la réversibilité.

Le premier des éléments rendu est l'architecture techniques de nos box dans le dossier d'architecture technique (DAT). Il y sera détaillé les choix Hardware, les branchements, les choix d'architecture des flux vidéo, les fichiers exécutables, les dépendances…

Ensuite, une ou plusieurs preuve(s) de concept (POC) du boîtier seront également rendus pour montrer les fonctionnalités développées. Nous aimerions travailler avec des étudiants en design pour design le boîtier imprimable en 3D à partir de plastique recyclé provenant des océans.

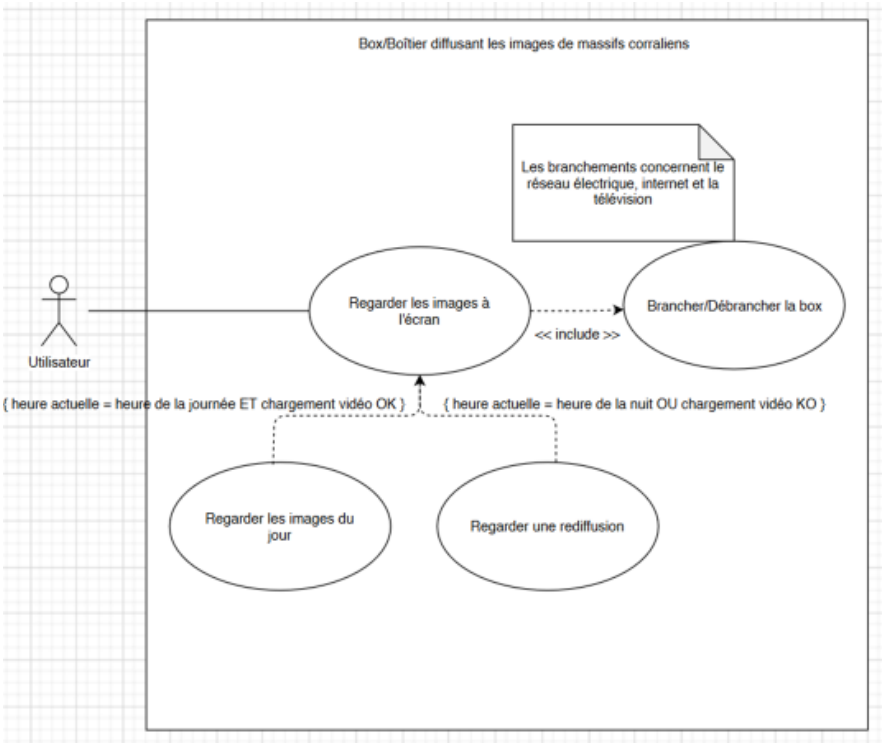
Enfin, nous aimerions qu'une chaine DevOps complète puisse être mis en place pour que la prochaine équipe de développement puisse continuer à implémenter de nouvelles fonctionnalités sans avoir à « rapatrier » les box. Cela facilitera grandement leurs phases développement.

Enfin tous les documents et présentations pptx seront rendus comme livrables pour documenter le plus précisément notre projet.

Système

Le système se veut extrêmement simple du point de vue de l'utilisateur. En effet, nous souhaitons qu'il ait simplement à brancher la box à interner (via câble Ethernet qui consomme moins que le WiFi et a un meilleur rendement), au réseau électrique et a un écran en HDMI (norme la plus populaire actuellement dans les pays occidentaux).

Il peut ensuite regarder les images de la vidéo de la webcam d'OceanStream à l'écran quand il est présent dans la pièce.

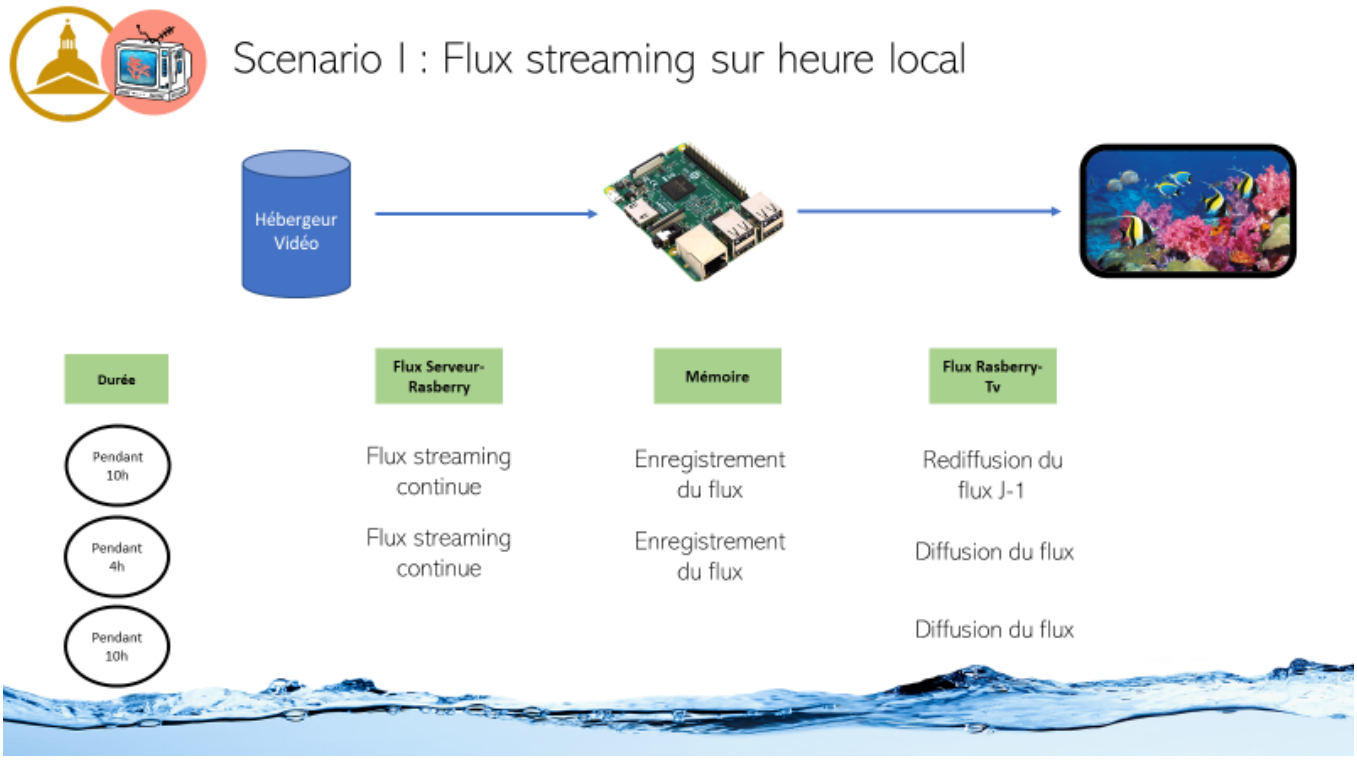


Il est important de noter qu'il y a une problématique importante de coordination temporelle. En effet, les webcam d'OceanStream filment pendant 14h (quand la lumière du jour est disponible). Nous devons donc implémenter une rediffusion pendant 10h.

De plus, les utilisateurs souhaitent voir les images filmées à l'heure x locale de la webcam à l'heure x de leur fuseau horaire. Un utilisateur qui se réveille à 10h voudra voir les images du lever du soleil et non du coucher. En d'autres termes, on doit supprimer les effets du décalage horaire pour les utilisateurs.

La solution proposée dépendra du type de flux choisi pour faire parvenir la vidéo. Nous avons défini 2 scénarios. Un avec un flux FTP et un avec un Flux Streaming. Nous faisons l'hypothèse la plus probable d'une caméra dans la grande barrière de corail (Australie +10h) et que l'architecture du flux est organisée sur 24h. Enfin la vidéo doit être diffusée en 4k.

Flux Streaming



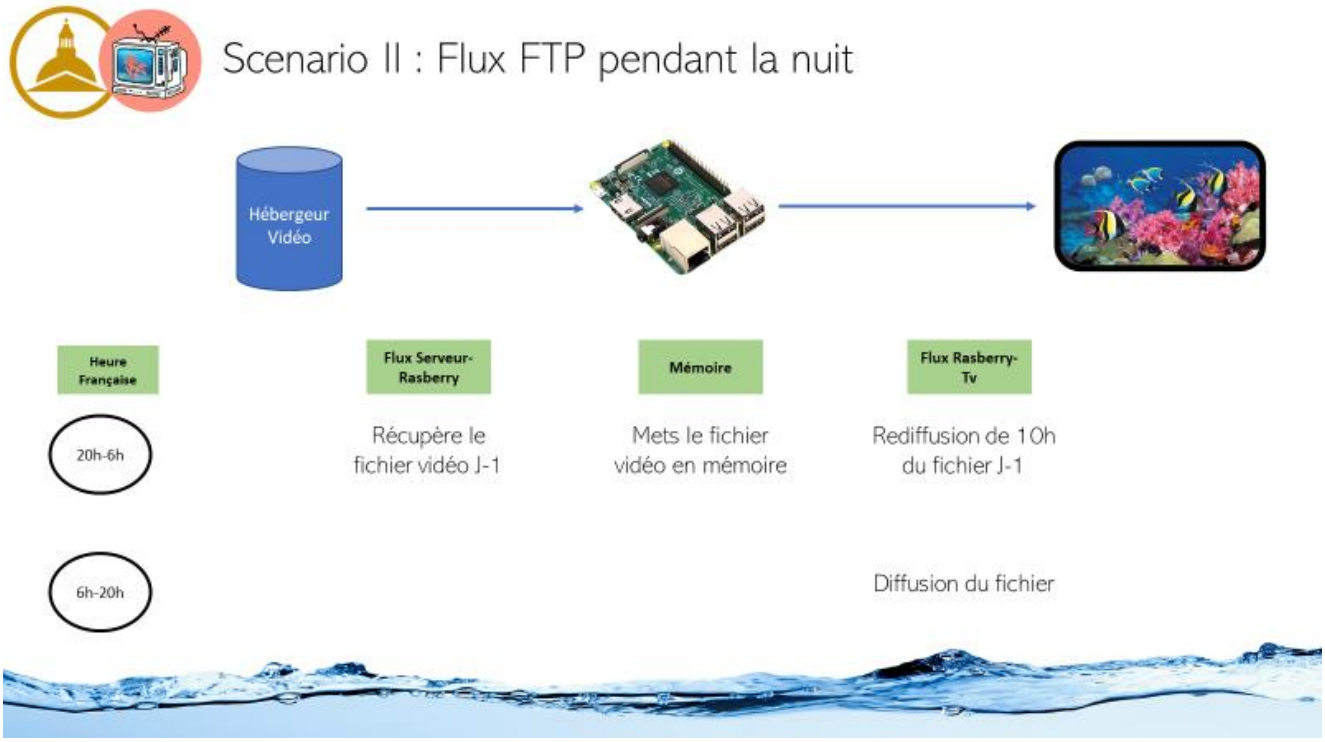
Pendant 10h, nous utilisons dans ce scénario un flux streaming pour récupérer la vidéo en continue et nous l'enregistrons. Puis, la vidéo enregistré le jour d'avant est rediffusée. Pendant 4h nous continuons de récupérer le flux streaming et diffusons le début du flux récupéré en streaming. Enfin, pendant 10h nous ne récupérons plus de vidéo et nous diffusons la fin de la vidéo récupérée en streaming.

Beaucoup d'éléments ont fait que nous n'avons pas choisi cette solution. Déjà d'un point de vue écologique. En effet, les flux streaming occupent en continue les infrastructures réseaux. Il est donc responsable de l'utilisation de beaucoup d'énergie en continue. De plus le Box devra également consommer plus pendant 14h pour récupérer le flux en continue. Il faudrait stocker 2 vidéos en 4k ce qui représente des cartes ou des disques couteux en ressources rares et polluant à la fabrication, à l'utilisation et au recyclage (quand il est fait).

Ensuite techniquement, cette implémentation pose plusieurs limites. Déjà, elle nécessite une bonne connexion internet pour récupérer sur 14h une vidéo en 4k. Des coupures réseaux, électriques ou simplement des drops de débit entraineront une qualité moindre sur certaine partie de la vidéo voir des parties de vidéo manquante qu'il faudrait gérer. Le fait d'avoir une qualité différente entre chaque box et différente dans le temps n'est pas acceptable pour OceanStream.

Enfin, d'un point de vue budgétaire, le(s) serveur(s) faisant transiter le flux à l'ensemble des box devront être puissant et nombreux pour un coup énergétique et donc financier important.

Flux FTP



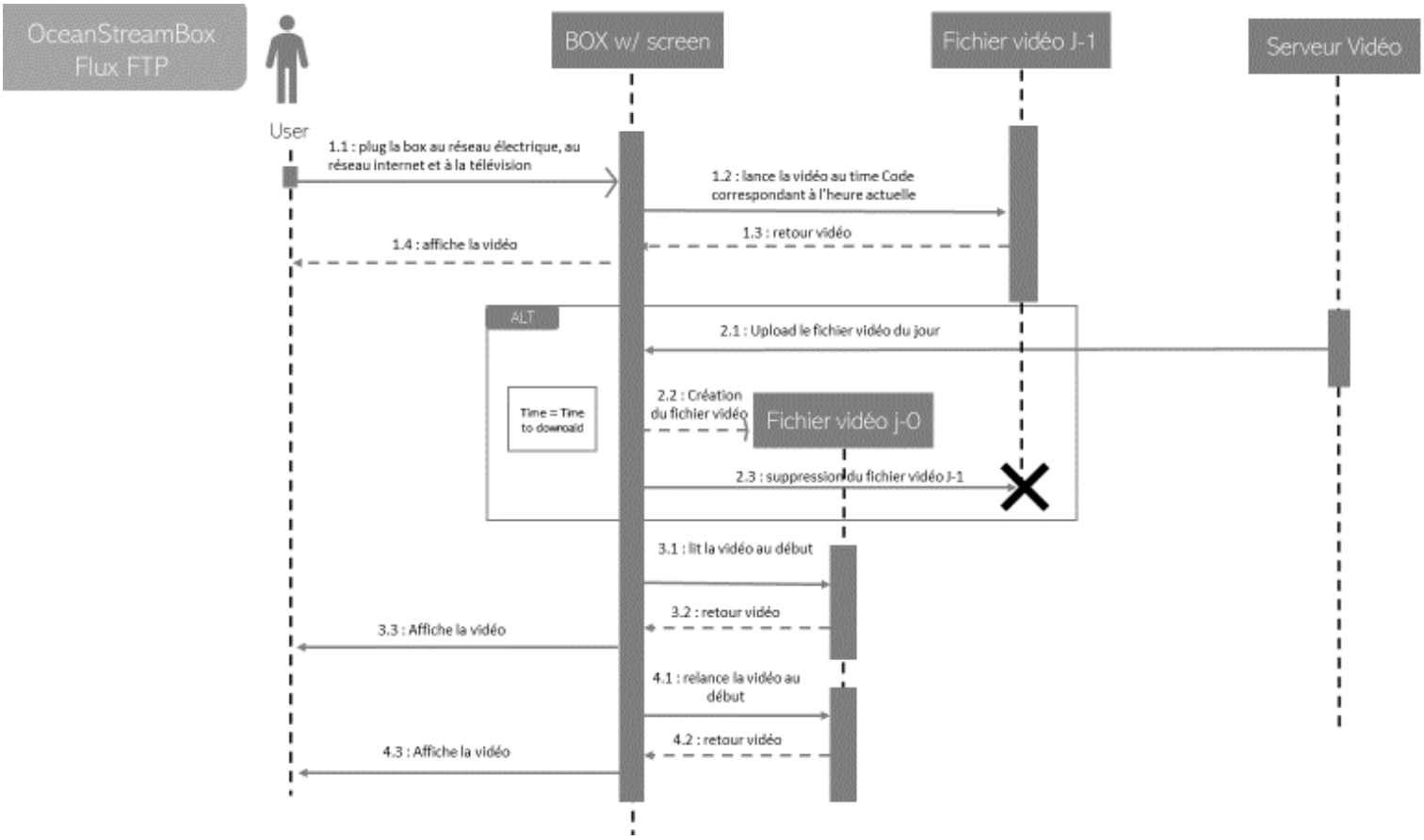
Dans ce scénario, nous récupérons le flux vidéo sous forme de fichier entre 20h et 6h (heure française) pendant que nous rediffusons la vidéo de la veille. Entre 6h et 20h, nous diffusons le fichier vidéo récupéré la nuit via FTP.

Le fait de récupérer un fichier vidéo nous permet de le compresser pour qu'il prenne moins de place sur les box. Moins de matériaux électroniques seront nécessaires. L'image étant fixe avec peu de mouvement, le fichier se prête très bien à la compression. Des tests devront être fait pour choisir le moteur et le format de compression. La récupération du fichier sera moins longue qu'avec un flux streaming et occupera moins les infrastructures réseau. De plus, récupérer le fichier dans la nuit est un choix judicieux du point de vue de l'écoconception car c'est le moment où l'énergie électrique ainsi que les infrastructures réseaux sont les moins utilisées. Nous pouvons même choisir l'horaire précise où elles sont le moins utilisées pour lancer le téléchargement FTP.

La qualité du fichier vidéo sera constante contrairement au streaming. Le téléchargement pourra être relancé en cas de coupure réseau ou électrique.

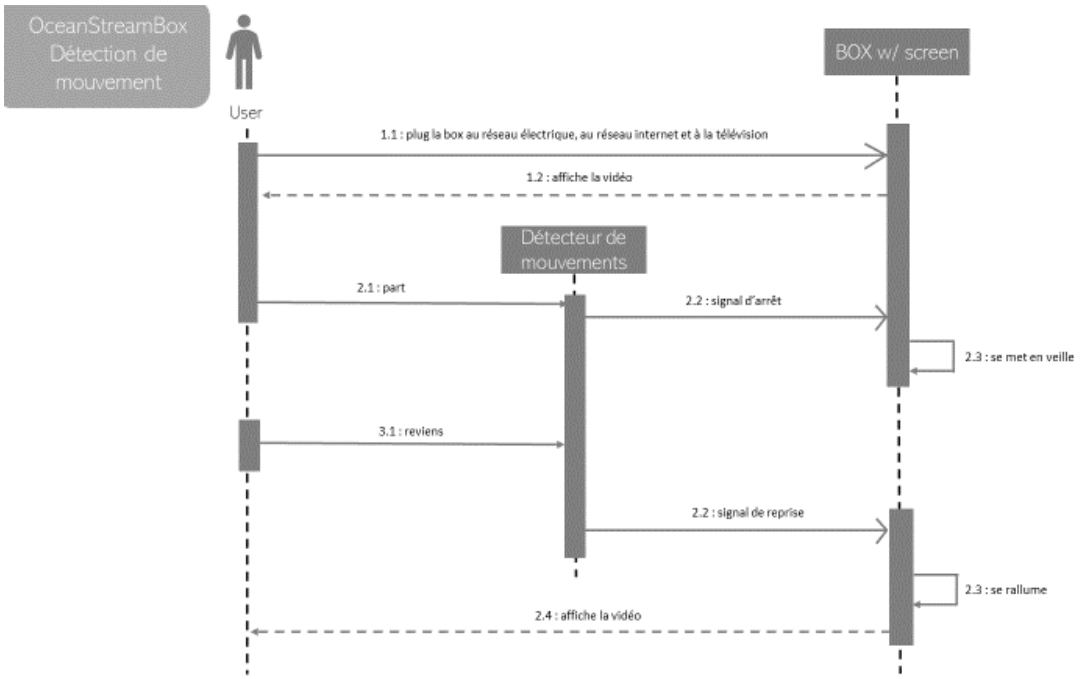
C'est pour toutes ces raisons que nous avons choisi d'utiliser un flux FTP.

Le diagramme de séquence qui décrit la récupération du fichier via un flux FTP est donc le suivant.



Détecteur de mouvement

Comme énoncé plus haut, par soucis d'économie d'énergie, nous souhaitons implémenter un détecteur de mouvement. Il sera responsable du signal de mise en veille et de la sortie de veille. Le diagramme de séquence suivant décrit son utilisation.



Choix Hardware

Pour choisir la technologie de carte mise dans notre box nous nous sommes fixés les critères suivants :

- La carte doit consommer peu d'électricité notamment via un système d'exploitation optimisé pour sa configuration hardware.
- La carte doit avoir une bonne capacité réseau pour récupérer assez rapidement le fichier vidéo quotidiennement. Elle doit disposer d'un port Ethernet pour que notre boîtier soit Plug and Play. Par ailleurs la technologie Ethernet offre un rendement débit/consommation d'énergie bien meilleur que le WiFi.
- La carte doit utiliser le moins de ressources possibles pour sa construction (métaux rares, matières premières…) pour réduire l'impact environnementale de sa construction ainsi que son recyclage.
- La carte doit pouvoir supporter un système d'exploitation supporté par git (noyaux Unix ou Microsoft) pour pouvoir mettre en place notre chaîne DevOps

Marque

Deux marques ont été confrontés. Il s'agit de Arduino et Raspberry leaders sur le marché des cartes programmables non industrielles.

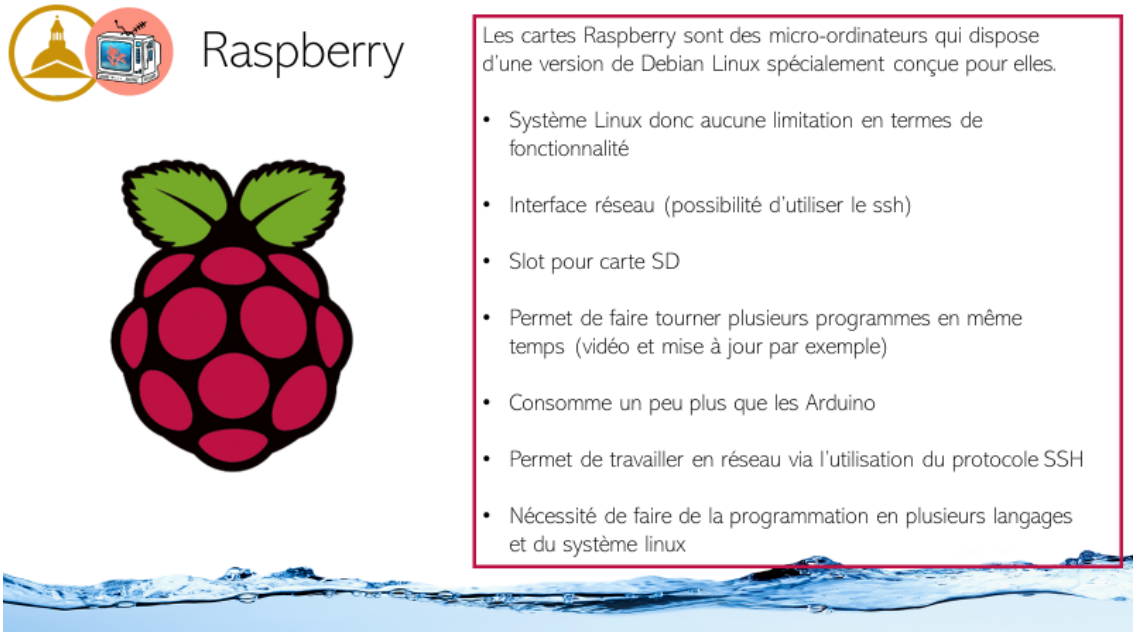
Nous avons présélectionné ces deux marques car il existe beaucoup de documentation et de forums sur leurs utilisations. Il existe également de nombreux modules complémentaires que nous pourrons utiliser.

Arduino

ARDUINO™

Les cartes Arduino sont des micros-contrôleurs open-source disposant de leur propre software et permettant de faire tourner un programme qui gère les entrées et lessorties.

- Le software d'Arduino est assez simple d'utilisation (proche du python) et offre un haut niveau d'abstraction.
- Idéal pour les projets avec peu de fonctionnalités.
- Entreprise Européenne (Italy).
- Dispose de son propre environnement de développement.
- Pas de slot carte SD de base
- Mise en réseau assez complexe
- Ce n'est pas un ordinateur donc impossible d'utiliser des programme tiers (git, VLC…)



Les Raspberry PI semblent plus convenir à nos besoins du fait de la multiplicité des fonctionnalités de l'OceanBox, de notre volonté de mettre toutes les cartes en réseaux, de notre connaissance en programmation système et de notre envie de mettre en production cette box.

Choix de la gamme[6]

Raspberry propose plusieurs gammes de cartes avec chacune leurs spécificités.

En termes de prix, la PI zéro se démarque du lot. En effet à 5€, elle représente un prix de revient pour OceanStream assez intéressant. Les B/B+ et A/A+ sont nettement plus chères car plus puissantes et proposant plus de ports. La PI ZeroW offre une interface Wifi par rapport à la PI Zero pour 5€ supplémentaires.

[6] Nous utilisons le tableau communautaire de ThePiHub : https://thepihut.com/blogs/raspberry-pi-roundup/raspberry-pi-comparison-table pour comparer les gammes. Vous pouvez retrouver ce tableau en annexe

Comparaison en terme de consommation électrique

B/B+

A/A+

Zero

ZeroW

B/B+	A/A+	Zero	ZeroW
450mA-1.2A	200-300mA	100-350mA	100-350mA

Encore une fois, c'est la PI Zero et ZeroW qui se démarquent avec une consommation en moyenne plus faible que les autres cartes.

Comparaison en terme d'impact environnementale du Hardware

B/B+

A/A+

Zero

ZeroW

B/B+	A/A+	Zero	ZeroW
C'est la carte qui utilise le plus de métaux notamment à cause des nombreux ports	Un peu moins consommatrice en ressource	Etants les plus petites carte Raspberry sur le marché, elle utilisent asse peu de métaux et de ressources dans leur fabrication	

Enfin, en termes d'utilisation des ressources à la construction, la PI Zero et ZeroW se démarquent encore dues à leurs petites tailles et leur faible nombre de ports.

PI A

- Moins cher de 10€ que le modèle B+
- Moins consommatrice en électricité
- Plus petite
- Toujours une quantité de port exaltable (USB, SATA, pins, HDMI)

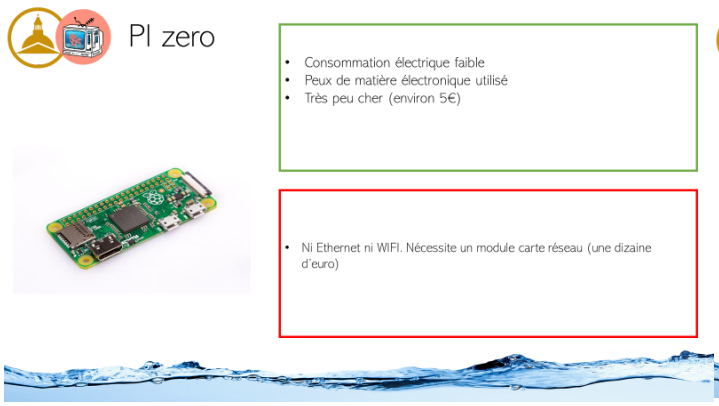
- Pas de port Ethernet. Nécessite une module Ethernet ce qui rends la carte plus cher et plus gros que le modèle B+

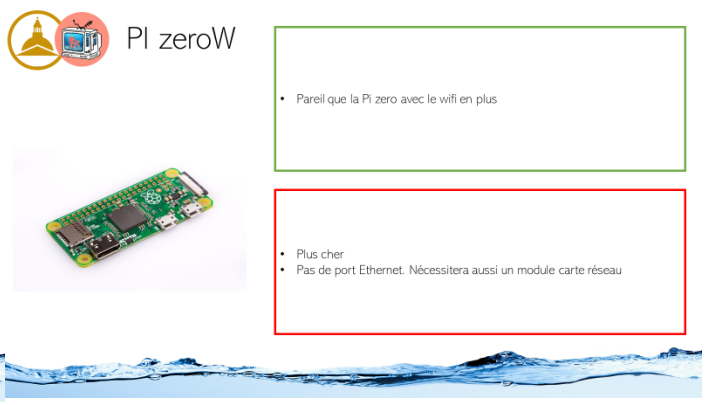
PI B+

- Carte la plus puissante
- Choix de la quantité de RAM (1, 2, 3 ou 4GO)
- Nombreux ports (USB 2.0, USB 3.0, Ethernet, HDMI, SATA, Pins…)
- Port Ethernet nécessaire à notre projet

- Carte la plus consommatrice en électricité
- Beaucoup de ports inutiles dans notre cas
- Puissance très importante par rapport à nos besoins
- Prix de 40€ assez conséquent

Nous ne choisirons donc pas la PI A/A+ et B/B+ qui sont trop puissantes et donc trop consommatrices en ressources et en énergie par rapport à nos besoins.

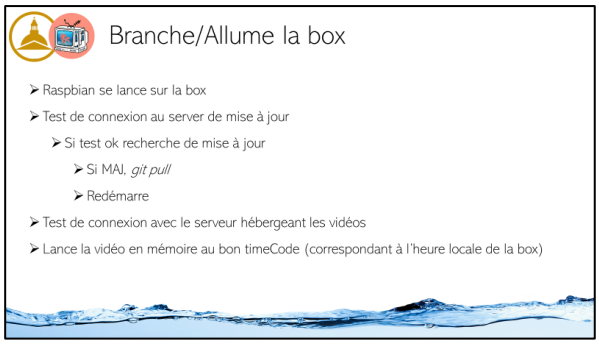


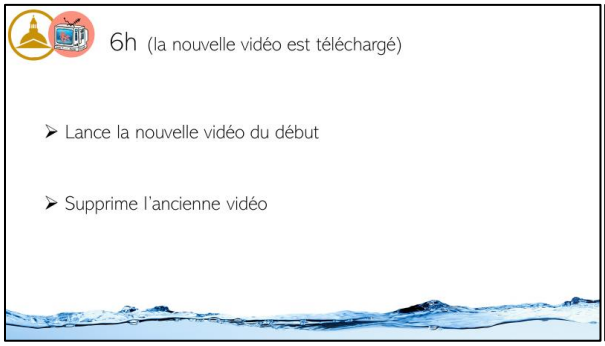


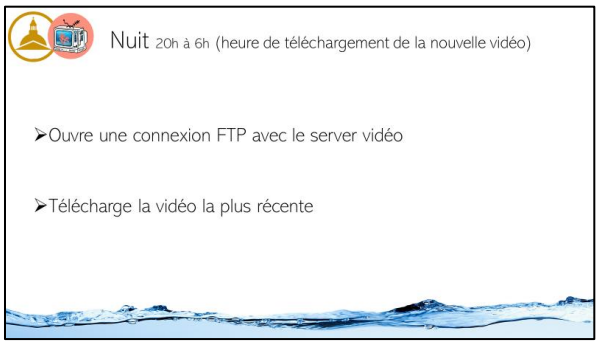
Reste donc à choisir entre la PI Zero et sa sœur la PI ZeroW. Il se trouve que les deux cartes ne disposent pas de port Ethernet. Il faudra donc leur ajouter un module Ethernet (prix d'environ 10€). Il n'est donc pas nécessaire d'avoir une interface Wifi. Nous choisissons donc la Raspberry avec la plus petite configuration : la PI Zéro

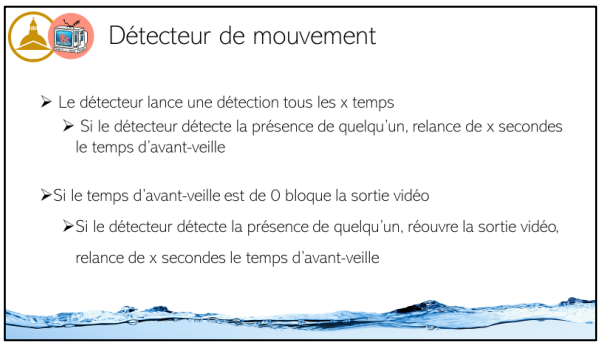
Annexe

Etapes d'une box

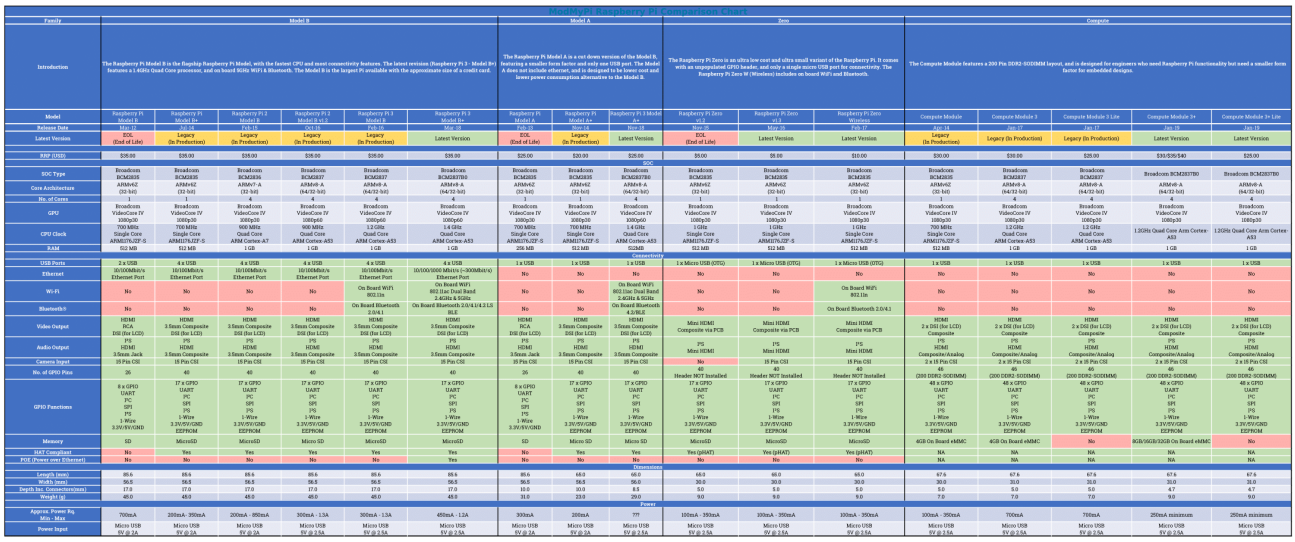








Figure



Modify!!! Raspberry Pi Comparison Chart																
Family	Model B						Model A			Zero			Compute				
Introduction	The Raspberry Pi Model B is the flagship Raspberry Pi Model, with the fastest CPU and most connectivity features. The latest revision (Raspberry Pi 3 - Model B+) features a 1.4GHz Quad Core processor, and on board 802.11 WiFi & Bluetooth. The Model B is the largest Pi available with the approximate size of a credit card.						The Raspberry Pi Model A is a cut down version of the Model B, featuring a smaller form factor and only one USB port. The Model A does not include ethernet, and is designed to be lower cost and lower power consumption alternative to the Model B.			The Raspberry Pi Zero is an ultra low cost and ultra small variant of the Raspberry Pi. It comes with an unpopulated GPIO header, and only a single micro USB port for connectivity. The Raspberry Pi Zero W (Wireless) includes on board WiFi and Bluetooth.			The Compute Module features a 200 Pin DDR2-SODIMM layout, and is designed for engineers who need Raspberry Pi functionality but need a smaller form factor for embedded designs.				
Model	Raspberry Pi Model B	Raspberry Pi Model B+	Raspberry Pi 2 Model B	Raspberry Pi 2 Model B v1.2	Raspberry Pi 3 Model B	Raspberry Pi 3 Model B+	Raspberry Pi Model A	Raspberry Pi Model A+	Raspberry Pi 3 Model A+	Raspberry Pi Zero v1.3	Raspberry Pi Zero v1.3	Raspberry Pi Zero Wireless	Compute Module	Compute Module 3	Compute Module 3 Lite	Compute Module 3+	Compute Module 3+ Lite
Release Date	May-11	Jul-14	Feb-15	Oct-16	Feb-16	Mar-18	Feb-13	Nov-14	Nov-18	Nov-15	May-16	Feb-17	Jun-14	Jan-17	Jan-17	Jan-19	Jan-19
Latest Version	EOL (End of Life)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Latest Version	EOL (End of Life)	Legacy (In Production)	Latest Version	EOL (End of Life)	Latest Version	Latest Version	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Latest Version	Latest Version
RRP (USD)	$35.00	$35.00	$35.00	$35.00	$35.00	$35.00	$25.00	$20.00	$25.00	$5.00	$5.00	$10.00	$30.00	$30.00	$25.00	$35/$35/$40	$25.00
SOC																	
SOC Type	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2837	Broadcom BCM2837B0	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2837B0	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2837	Broadcom BCM2837	Broadcom BCM2837B0	Broadcom BCM2837B0
Core Architecture	ARMv6Z	ARMv6Z	ARMv7-A	ARMv8-A	ARMv8-A	ARMv8-A	ARMv6Z	ARMv6Z	ARMv8-A	ARMv6Z	ARMv6Z	ARMv6Z	ARMv6Z	ARMv8-A	ARMv8-A	ARMv8-A	ARMv8-A
No. of Cores	1	1	4	4	4	4	1	1	4	1	1	1	1	4	4	4	4
GPU	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV
CPU Clock	700 MHz Single Core ARM1176JZF-S	700 MHz Single Core ARM1176JZF-S	900 MHz Quad Core ARM Cortex-A7	900 MHz Quad Core ARM Cortex-A53	1.2 GHz Quad Core ARM Cortex-A53	1.4 GHz Quad Core ARM Cortex-A53	700 MHz Single Core ARM1176JZF-S	700 MHz Single Core ARM1176JZF-S	1.4 GHz Quad Core ARM Cortex-A53	1 GHz Single Core ARM1176JZF-S	1 GHz Single Core ARM1176JZF-S	1 GHz Single Core ARM1176JZF-S	700 MHz Single Core ARM1176JZF-S	1.2 GHz Quad Core ARM Cortex-A53	1.2 GHz Quad Core ARM Cortex-A53	1.2GHz Quad Core Arm Cortex-A53	1.2GHz Quad Core Arm Cortex-A53
RAM	512 MB	512 MB	1 GB	1 GB	1 GB	1 GB	256 MB	512 MB	512MB	512 MB	512 MB	512 MB	512 MB	1 GB	1 GB	1 GB	1 GB
Connectivity																	
USB Ports	2 x USB	4 x USB	4 x USB	4 x USB	4 x USB	4 x USB	1 x USB	1 x USB	1 x USB	1 x Micro USB (OTG)	1 x Micro USB (OTG)	1 x Micro USB (OTG)	1 x USB	1 x USB	1 x USB	1 x USB	1 x USB
Ethernet	10/100Mbit/s Ethernet Port	10/100Mbit/s Ethernet Port	10/100Mbit/s Ethernet Port	10/100Mbit/s Ethernet Port	10/100Mbit/s Ethernet Port	10/100/1000 Mbit/s (~300Mbit/s) Ethernet Port	No	No	No	No	No	No	No	No	No	No	No
Wi-Fi	No	No	No	No	On Board WiFi 802.11n	On Board WiFi 802.11ac Dual Band 2.4GHz & 5GHz	No	No	On Board WiFi 802.11n	No	No	On Board WiFi 802.11n	No	No	No	No	No
Bluetooth®	No	No	No	No	On Board Bluetooth 2.0/4.1	On Board Bluetooth 2.0/4.1/4.2 LS BLE	No	No	On Board Bluetooth 2.0/4.1	No	No	On Board Bluetooth 2.0/4.1	No	No	No	No	No
Video Output	HDMI RCA DSI (for LCD)	HDMI 3.5mm Composite DSI (for LCD)	HDMI 3.5mm Composite DSI (for LCD)	HDMI 3.5mm Composite DSI (for LCD)	HDMI 3.5mm Composite DSI (for LCD)	HDMI 3.5mm Composite DSI (for LCD)	HDMI RCA DSI (for LCD)	HDMI 3.5mm Composite DSI (for LCD)	HDMI 3.5mm Composite DSI (for LCD)	Mini HDMI Composite via PCB	Mini HDMI Composite via PCB	Mini HDMI Composite via PCB	HDMI 2 x DSI (for LCD) Composite	HDMI 2 x DSI (for LCD) Composite	HDMI 2 x DSI (for LCD) Composite	HDMI 2 x DSI (for LCD) Composite	HDMI 2 x DSI (for LCD) Composite
Audio Output	I²S HDMI 3.5mm Jack	I²S HDMI 3.5mm Composite	I²S HDMI 3.5mm Composite	I²S HDMI 3.5mm Composite	I²S HDMI 3.5mm Composite	I²S HDMI 3.5mm Composite	I²S HDMI 3.5mm Jack	I²S HDMI 3.5mm Composite	I²S HDMI 3.5mm Composite	I²S Mini HDMI	I²S Mini HDMI	I²S Mini HDMI	I²S HDMI Composite/Analog	I²S HDMI Composite/Analog	I²S HDMI Composite/Analog	I²S HDMI Composite/Analog	I²S HDMI Composite/Analog
Camera Input	15 Pin CSI	15 Pin CSI	15 Pin CSI	15 Pin CSI	15 Pin CSI	15 Pin CSI	15 Pin CSI	15 Pin CSI	15 Pin CSI	No	15 Pin CSI	15 Pin CSI	2 x 15 Pin CSI	2 x 15 Pin CSI	2 x 15 Pin CSI	2 x 15 Pin CSI	2 x 15 Pin CSI
No. of GPIO Pins	26	40	40	40	40	40	26	40	40	40 Header NOT Installed	40 Header NOT Installed	40 Header NOT Installed	46 (200 DDR2-SODIMM)	46 (200 DDR2-SODIMM)	46 (200 DDR2-SODIMM)	46 (200 DDR2-SODIMM)	46 (200 DDR2-SODIMM)
GPIO Functions	8 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	8 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	17 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	46 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	46 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	46 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	46 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM	46 x GPIO UART I²C SPI I²S 1-Wire 3.3V/5V/GND EEPROM
Memory	SD	MicroSD	MicroSD	Micro SD	MicroSD	MicroSD	SD	Micro SD	Micro SD	MicroSD	MicroSD	MicroSD	4GB On Board eMMC	4GB On Board eMMC	No	8GB/16GB/32GB On Board eMMC	No
HAT Compliant	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes (pHAT)	Yes (pHAT)	Yes (pHAT)	NA	NA	NA	NA	NA
POE (Power over Ethernet)	No	No	No	No	No	Yes	No	No	No	No	No	No	NA	NA	NA	NA	NA
Dimensions																	
Length (mm)	85.6	85.6	85.6	85.6	85.6	85.6	85.6	60.0	60.0	65.0	65.0	65.0	67.6	67.6	67.6	67.6	67.6
Width (mm)	56.5	56.5	56.5	56.5	56.5	56.5	56.5	56.5	56.5	30.0	30.0	30.0	30.0	31.0	31.0	31.0	31.0
Depth (inc. Connectors)(mm)	17.0	17.0	17.0	17.0	17.0	17.0	10.0	10.0	8.5	5.0	5.0	5.0	5.0	5.0	5.0	4.7	4.7
Weight (g)	45.0	45.0	48.0	48.0	48.0	48.0	31.0	23.0	29.0	9.0	9.0	9.0	7.0	7.0	7.0	9.0	9.0
Power																	
Approx. Power Eq. Min - Max	700mA	200mA - 350mA	200mA - 850mA	300mA - 1.3A	300mA - 1.3A	450mA - 1.2A	300mA	200mA	???	100mA - 350mA	100mA - 350mA	100mA - 350mA	100mA - 350mA	700mA	700mA	250mA minimum	250mA minimum
Power Input	Micro USB 5V @ 2A	Micro USB 5V @ 2A	Micro USB 5V @ 2A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A	Micro USB 5V @ 1A	Micro USB 5V @ 2A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A	Micro USB 5V @ 2.5A

Tableau 1 Raspberry by comparison - https://thepihut.com/blogs/raspberry-pi-roundup/raspberry-pi-comparison-table

Bibliographie partielle

EcoConception

Page Wikipédia dédiée à l'écoconception - https://fr.wikipedia.org/wiki/%C3%89coconception

Restitution du travail de Greenpeace sur la pollution numérique – 10 janvier 2017 - https://www.greenpeace.fr/la-pollution-numerique/

Site de la communauté française de la green IT - https://www.greenit.fr/

Méthode AGILE

Page Wikipédia dédiée à la méthode AGILE - https://fr.wikipedia.org/wiki/M%C3%A9thode_agile

Points problématiques de la méthode Scrum - https://blog.myagilepartner.fr/index.php/2019/12/20/le-sprint-objectif-un-scrum-implementation-point-of-failure/

How to implement Scrum - https://www.knowledgehut.com/blog/agile/implementing-scrum-in-organizations

Page Wikipédia dédiée à la méthode Scrum - https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement)

DevOps

Page Wikipédia dédié au DevOps - https://fr.wikipedia.org/wiki/Devops

Page Git de conseil en DevOps - https://gist.github.com/jpswade/4135841363e72ece8086146bd7bb5d91

Flux

Blog Raspberry concernant la création d'un flux streaming - http://www.magdiblog.fr/divers/raspberry-pi-camera-5-facons-de-faire-du-streaming/

Documentation officielle Raspberry concernant les flux ftp - https://www.raspberrypi.org/documentation/remote-access/ftp.md

notre-plantet.info - Le Streaming est-il un gouffre énergétique ? –https://www.notre-planete.info/actualites/247-streaming-emissions-CO2

Raspberry et Arduino

The PIHUB – Site de vente de produits autour de la Raspberry - https://thepihut.com/

Comparaison Arduino vs Raspberry - https://www.electronicshub.org/raspberry-pi-vs-arduino/