

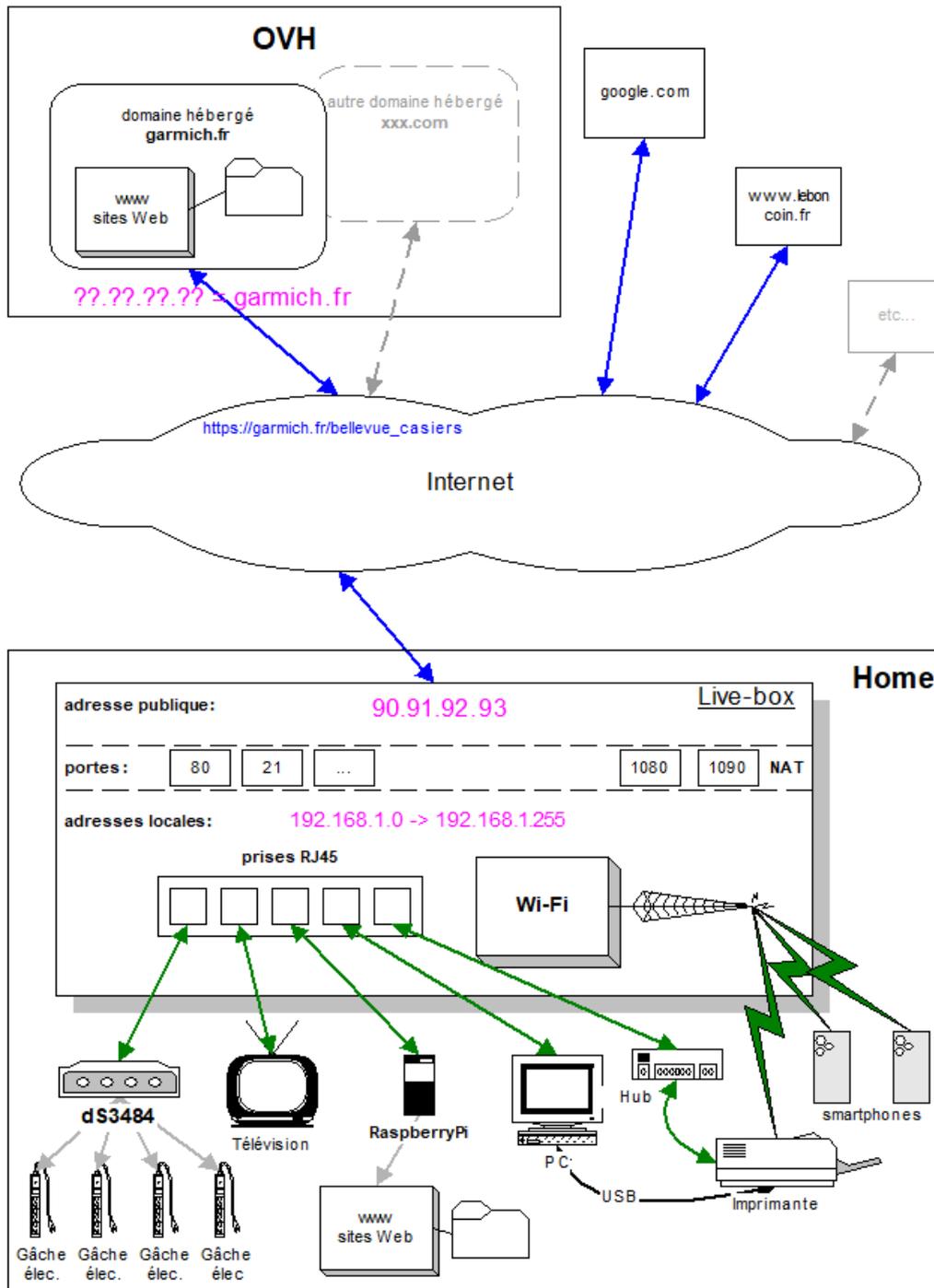
Bellevue - Casiers connectés

Adresse IP publique - Suite

Comme je l'ai expliqué dans le précédente document, il est essentiel de connaître l'adresse IP publique de la **Live-Box** sur laquelle est connecté le module **dSxxxx** et/ou le **RaspberryPi**.

Schéma de principe

Pour bien comprendre les besoins, voici le schéma d'une installation connectée à internet :



■ la partie du haut, le cadre **OVH**, symbolise un domaine hébergé chez un prestataire extérieur. Cet abonnement comprend généralement :

- un nom de domaine : ici **garmich.fr**. Les serveurs DNS du monde entier connaissent ce nom et peuvent le traduire en adresse IP réelle (celle du système **OVH**). On peut connaître cette IP lorsque, par exemple, on accède à un site WEB du **RaspberryPi** depuis un site sur **garmich.fr** chez **OVH** (en utilisant une porte **NAT**). Mais cette information n'a que peu d'intérêt face à la praticité du nom de domaine référencé mondialement !!
- un espace **www** : c'est une zone dans laquelle on place les différents fichiers d'un site WEB (**bellevue_casiers** par exemple). On y accède généralement par **FTP**, voire par **SSH** si l'abonnement le permet. **OVH** fournit un nom d'utilisateur et un mot de passe à cet effet ;
- un groupe d'adresses **mél** : mon abonnement m'en propose 10.

Bien d'autres acteurs du WEB sont connectés (**google**, **leboncoin** etc...)

■ la partie du centre symbolise le réseau mondial internet. Chaque système connecté dessus dispose d'une adresse IP unique au monde (l'explosion d'internet rend l'adressage IPv4 presque saturé, raison pour laquelle IPv6 est de plus en plus poussé...). Ce sont des adresses dites publiques, car les serveurs DNS (**Domain Name System**) les exposent à tout le monde.

■ la partie du bas, le cadre **Home**, symbolise un foyer disposant d'une connexion Internet. On y trouve :

- un routeur internet : ici la **live-box**. Elle est connecté au réseau mondial via une ligne téléphonique (ADSL) ou désormais une fibre optique. Le prestataire (**Orange** par exemple) effectue le câblage physique jusque dans l'habitation du client chez lequel il pose un boîtier mural. La box est ensuite reliée au boîtier avec la liaison idoine (câble RJ11 ou fibre optique).

Chaque box dispose d'une adresse publique unique, symbolisée ici par **90.91.92.93**. Lorsqu'elle démarre, la box va demander au prestataire (**Orange** par exemple) une adresse IP unique. Celle-ci peut être invariable (fixe) chez certains prestataires, mais elle est variable chez **Orange**.

Vu le nombre faramineux de systèmes connectés (PC, portables, télévisions, imprimantes, caméras de surveillance, arrosages connectés, ...), il est techniquement impossible de les relier tous directement au réseau mondial ! La box propose donc un mini-réseau, dit local ou privé, sur lequel nos systèmes personnels sont connectés, soit par des câbles RJ45, soit de plus en plus fréquemment par Wi-Fi. Ces réseaux locaux utilisent un jeu d'adresses IP qui leur est spécialement réservé : généralement **192.168.1.xxx** avec **xxx** de **0** à **255** (en fait, c'est plus compliqué, mais sans intérêt pour nous...).

Tous les systèmes locaux communiquent avec internet dans les deux sens en utilisant des « portes » ; ce sont les fameux ports, dont les plus connus sont :

- 21 : FTP
- 22 : SSH
- 25 : SMTP (pour les méls)
- 80 : HTTP (pour les sites WEB)
- 110 : POP3 (pour les méls)
- 115 : SFTP

(voir la liste en https://fr.wikipedia.org/wiki/Liste_de_ports_logiciels)

Lorsque l'on ouvre une nouvelle porte pour accéder à un module **dSxxxx**, on utilise un port externe inutilisé (par exemple **1090**), et on indique la cible : `<ip_ds>` et `<port_interne_ds>`. Lorsque de l'extérieur, on «frappe à la porte» avec `<ip_box>:<port_externe>`, la box va traduire ça en `<ip_ds>:<port_interne_ds>`. D'où le **T** de **NAT** : *translation*, traduction.

- des systèmes connectés : ce sont tous nos appareils : PC, téléviseurs, smartphones... Ils sont en liaison avec la box, soit par les prises **RJ45** pour les systèmes fixes, soit par **Wi-Fi** pour nos appareils mobiles. Les connexions filaires sont plus rapides et stables que le Wi-Fi ; elles sont à préférer pour les PC non portables, les téléviseurs, les modules **dSxxx**, les RaspberryPi, les imprimantes. Pour ces dernières, elles pourront également être liées par USB à un PC, et désormais souvent liées en Wi-Fi à la box, permettant ainsi aux smartphones d'y imprimer leurs documents. D'aucuns reprochent au Wi-Fi d'émettre des ondes nocives, au même titre que les smartphones. Étant dans le doute, j'essaie de limiter ces effets en utilisant des écouteurs tant que faire se peut...
- module dSxxx : c'est un élément essentiel pour le projet qui nous occupe, et qui permet d'agir électriquement sur d'autres dispositifs non connectés comme des gâches électriques. On peut communiquer avec lui en TCP-IP, en mode ASCII (voir documents précédents).
- RaspberryPi : ce pico-ordinateur est un moyen idéal pour héberger un site WEB local, que l'on peut rendre accessible de l'extérieur via une **NAT**. Équipé de disques SSD, il peut contenir nos vidéos, photos, musiques et autres fichiers... Avec ce système, toutes nos données sont « à la maison », loin des GAFAM !

Les chemins

Dans la configuration décrite ci-dessus, les données transitent dans les deux sens selon ce chemin :

Pour un module **dSxxx** piloté depuis l'extérieur :

site distant :	(OVH site Web)
IP OVH :	garmich.fr
réseau mondial :	(internet)
ip box :	90.91.92.93
porte box :	:1090
ip module dSxxx :	192.168.1.152
port module :	:17123
module :	(dSxxx)

On voit que le seul paramètre qui est potentiellement variable est l'IP de la box.

La solution actuelle

Je souhaitais construire mon propre site WEB local. J'ai donc installé un **RaspberryPi**, qui est un pico-ordinateur conçu avec des composants de téléphone portable. D'un coût très réduit (environ 70€), on peut lui installer l'OS Linux avec interface graphique, clavier, souris, écran, disques USB SSD. Il se connecte au réseau local via une connexion Ethernet en RJ45. L'OS dispose d'applications pré-installées comme un navigateur WEB, **LibreOffice**, **Python**, **VLC**, un visionneur PDF... On pourrait très bien y installer **NetBeans** et développer dessus !

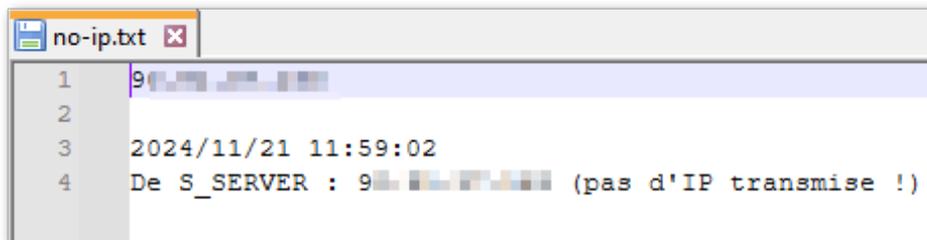
Pour faire un site WEB, je lui est installé **Apache** pour disposer de l'environnement nécessaire, et **PHP** pour concevoir les scripts de mes sites. Je lui ai aussi connecté en USB 4 disques SSD (2To, 1To, 500Go et 500Go) sur lesquels j'ai mis mes ressources personnelles (vidéo, musique). Le site WEB a été développé sur mon PC, avec **NetBeans** et une synchronisation automatique. J'ai créé une règle **NAT** me permettant d'y accéder depuis l'extérieur, depuis **garmich.fr**. Une simple redirection renvoie sur mon site local.

Disposant donc d'un **RaspberryPi** dans le réseau local, je l'ai également utilisé pour récupérer automatiquement l'IP publique de la box du réseau local :

- installation d'une tâche **cron** avec deux commandes à exécuter tous les jours à 11:00 et 19:00 heures : **set_ip_public.py**.
- en substance, le script **Python** va solliciter l'URL **http://garmich.fr/domus/setIp.php?no-ip=meudon** et récupérer la réponse, qui est l'IP du demandeur, c'est à dire la box.

Du côté OVH, le script **setIp.php** va effectuer les opérations suivantes :

- lecture de la variable **\$_SERVER[]** et recherche de l'IP du demandeur (la box) ;
- stockage du résultat dans un sous-répertoire au nom du paramètre (ici, **meudon**) et dans un fichier ASCII nommé **no-ip.txt** :



```
no-ip.txt x
1 9[redacted]
2
3 2024/11/21 11:59:02
4 De S_SERVER : 9[redacted] (pas d'IP transmise !)
```

Lorsque, depuis l'extérieur, je lance mon site depuis **garmich.fr**, je vais lire l'IP publique dans le fichier **no-ip.txt** du bon répertoire, puis je redirige la demande vers la version locale, via l'IP publique. Et le tour est joué !

Le principal avantage est que je maîtrise entièrement la chaîne et que je peux rajouter autant de boxes que je veux. J'utilise cette astuce depuis des années maintenant, sans problèmes.

Mais l'inconvenient est de taille : il nécessite un appareil de plus dans chaque réseau local (ici, des **RaspberryPi**'s), automate qu'il faut installer et programmer en conséquence.

Une autre méthode ?

Je vais réfléchir à une autre méthode n'utilisant qu'un seul système :

- soit un unique module **dSxxx** qui serait capable de solliciter l'URL **http://garmich.fr/domus/setIp.php?no-ip=meudon** ;
- soit un unique **RaspberryPi** complété avec un shield comportant le nombre de relais nécessaires.

■ Michel GARREAU