

# Photomaton !

(Python)Photomaton

Simon CHOLLET



- Les principes
- Les options
- Déroulement
- Un peu de Web, de réseau
- Intégration
- Mémo !



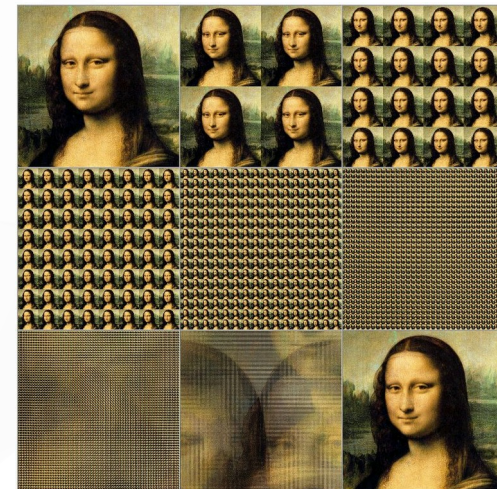
# Python pour photos

- Les principes
- Les options
- A l'attaque



# Les principes ...

- Avoir un système **autonome** :
  - Démarrage **automatique**
  - Alimentation simple
  - Aucune intervention de l'extérieur
- Fonctionnalités principales :
  - **Prendre** une photo sans avoir de photographe
  - **Visualiser** la photo
  - **Choisir** la bonne photo
  - **Obtenir** la ou les photos
- Doit être « **UPUPIDES compatible** » :
  - Maximum de choses récupérées
  - Minimum de matériel
  - Apprendre de nouvelles technologies





# Options possibles

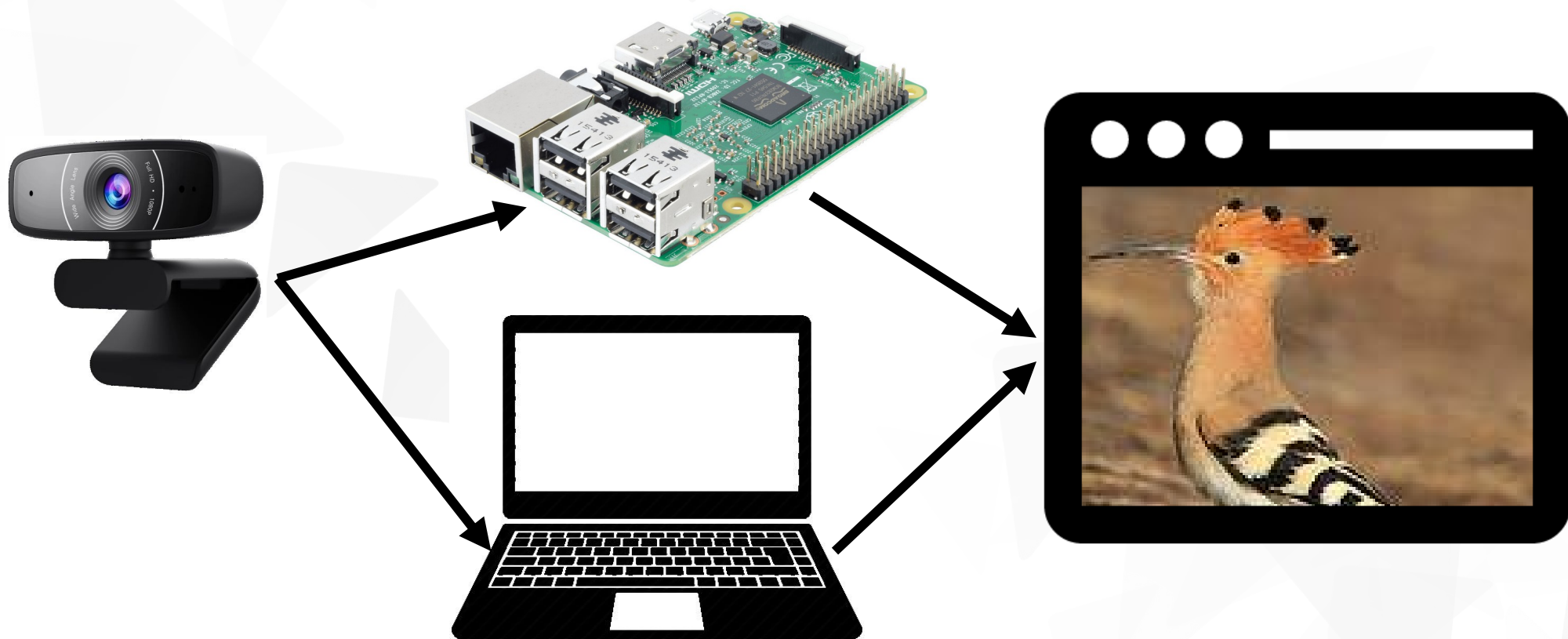
- **Monétiser** la prise de photos :
  - Conserver la photo **originale**
  - **Dégrader** de la photo visible
- Envoyer sur un **cloud** les photos :
  - Demande un serveur **extérieur**
  - Besoin d'une connexion **Internet**
- **Sécuriser** l'accès aux photos :
  - Toute machine sur Internet demande une **protection**
  - **Accès unique** de l'utilisateur à ces clichés
  - **Cryptage** de la zone de stockage
- Tests réguliers sur la **qualité** des images
- **Sauvegarde** automatique des photos





# Au commencement ...

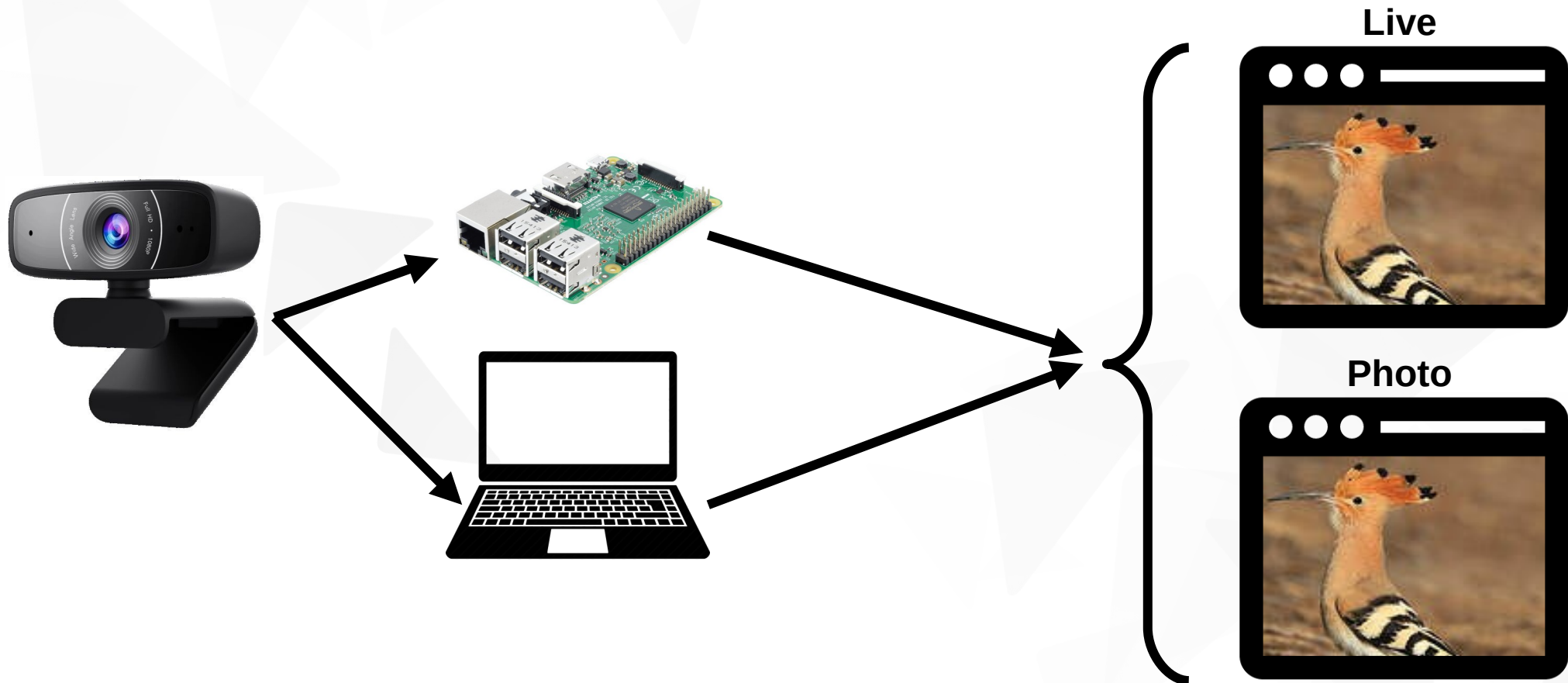
- Prise de photo avec une **Webcam** :
  - Utilisation de la librairie **OpenCV**
  - Acquisition des images en « **live** »
  - Affichage de l'image sur une **fenêtre**





# On code ... on code ...

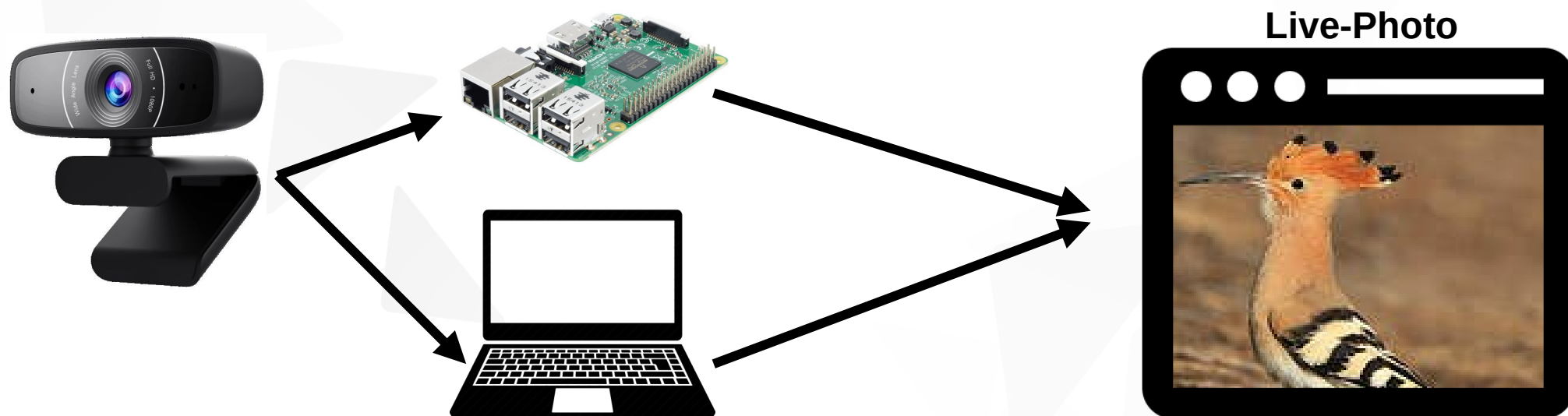
- Gestion des **touches** pour prendre une photo
- Affichage de la photo sur une **autre** fenêtre





# On continue !

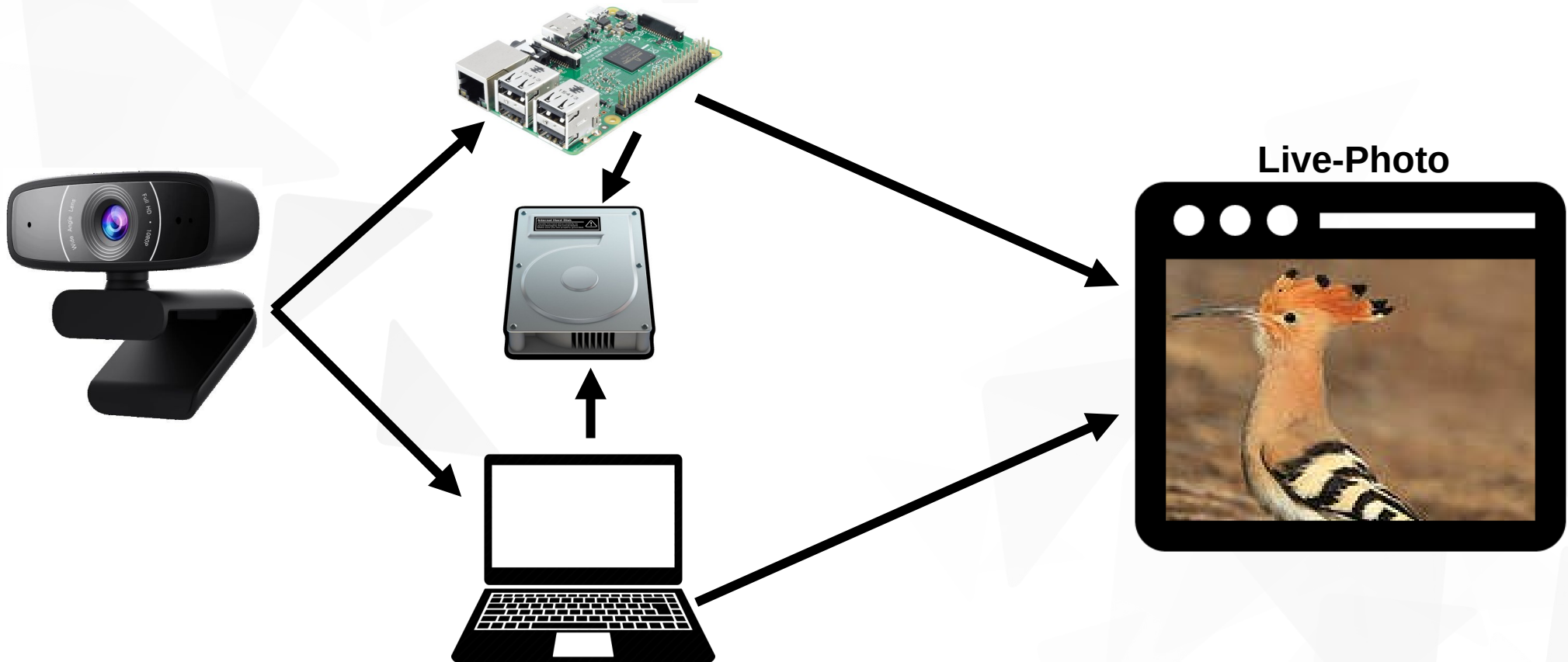
- Affichage photo sur la **même fenêtre** pendant un certain temps





# Une étape de plus !

- **Enregistrement** de la photo
- Construction d'un **nom de répertoire** de stockage
- Construction d'un **nom de fichier** photo
- Utilisation des modules **os.path** et **datetime**

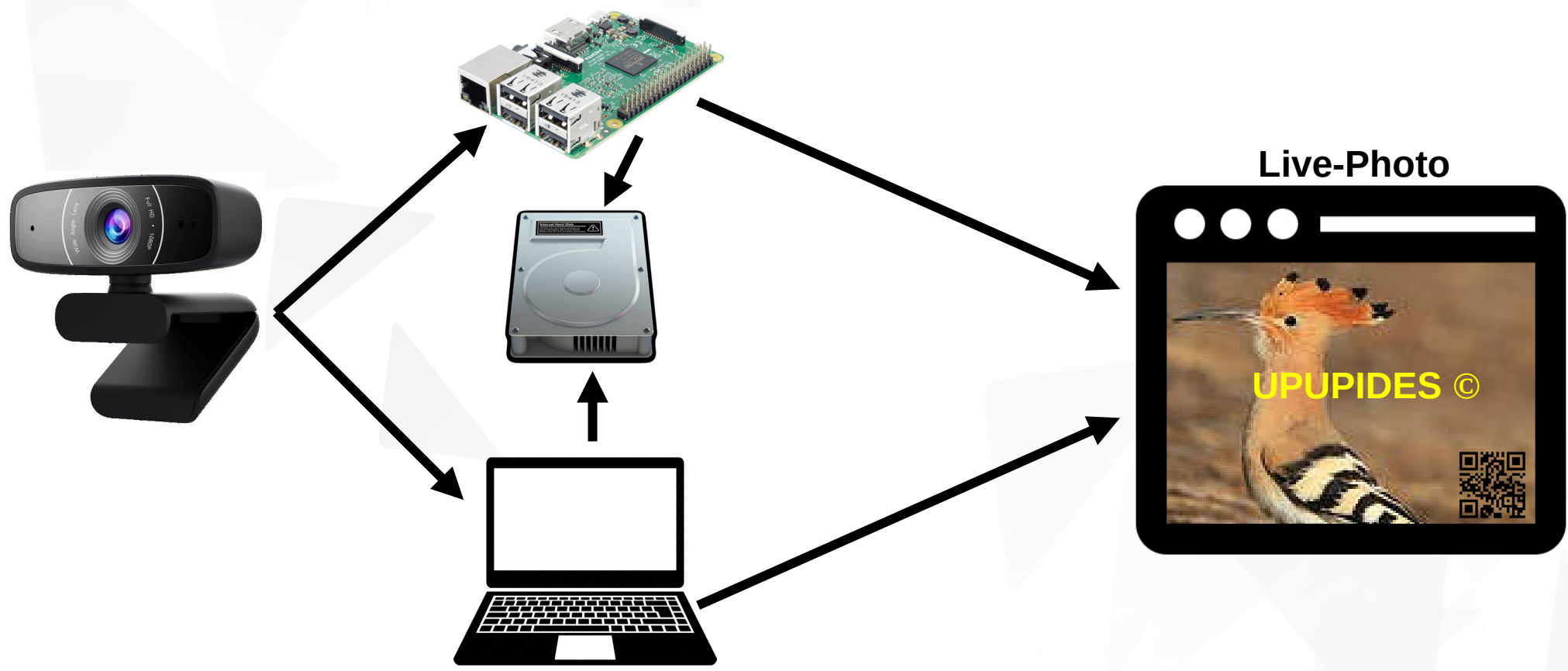






# Ajout d'éléments sur la photo !

- **Texte** : la date, un texte libre, etc.
- **QRCode** : module supplémentaire nécessaire



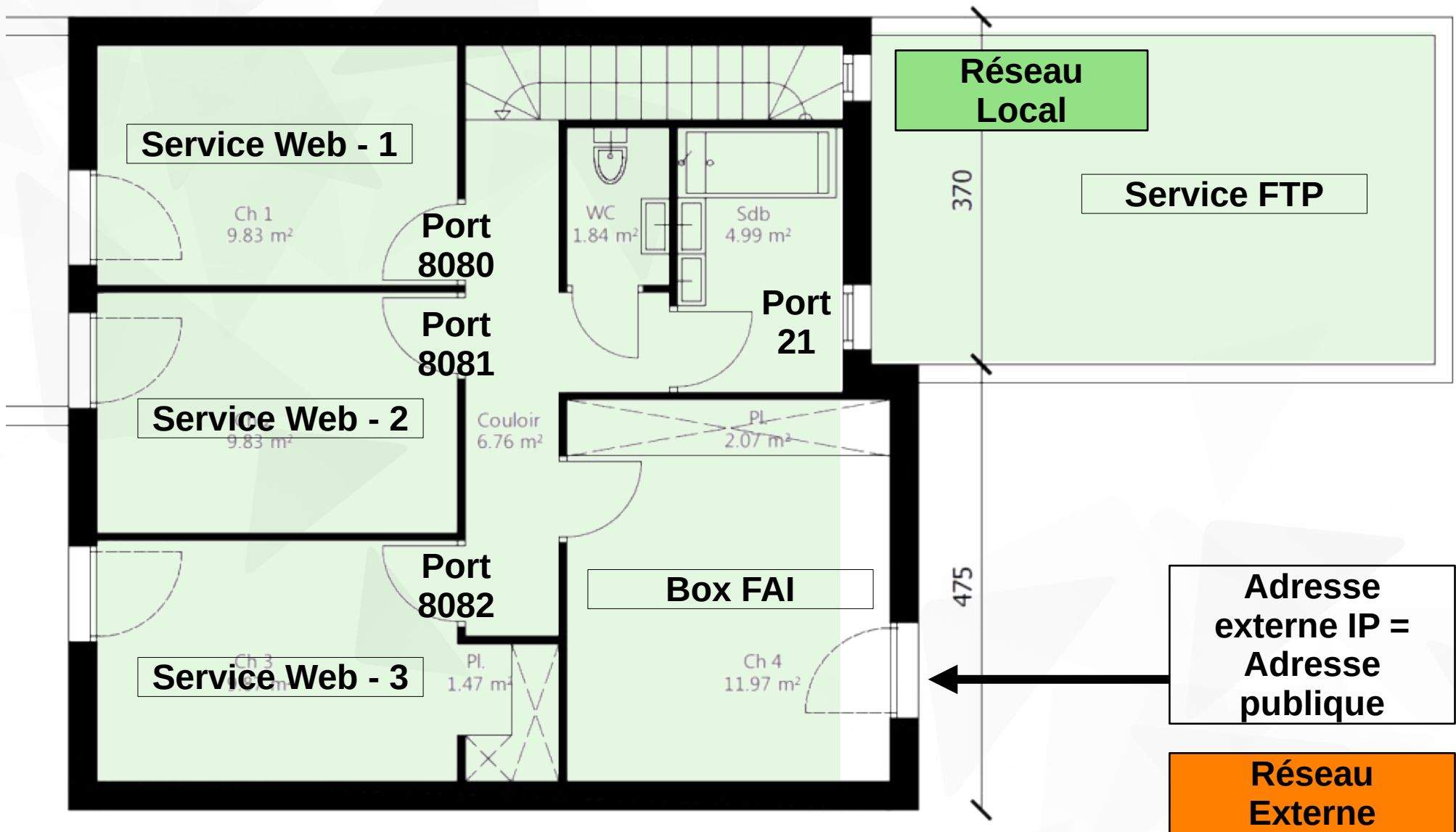


## Un peu de Web ... un peu de réseau

- Adresses IP – Ports
- Python et le Web (HTTP)



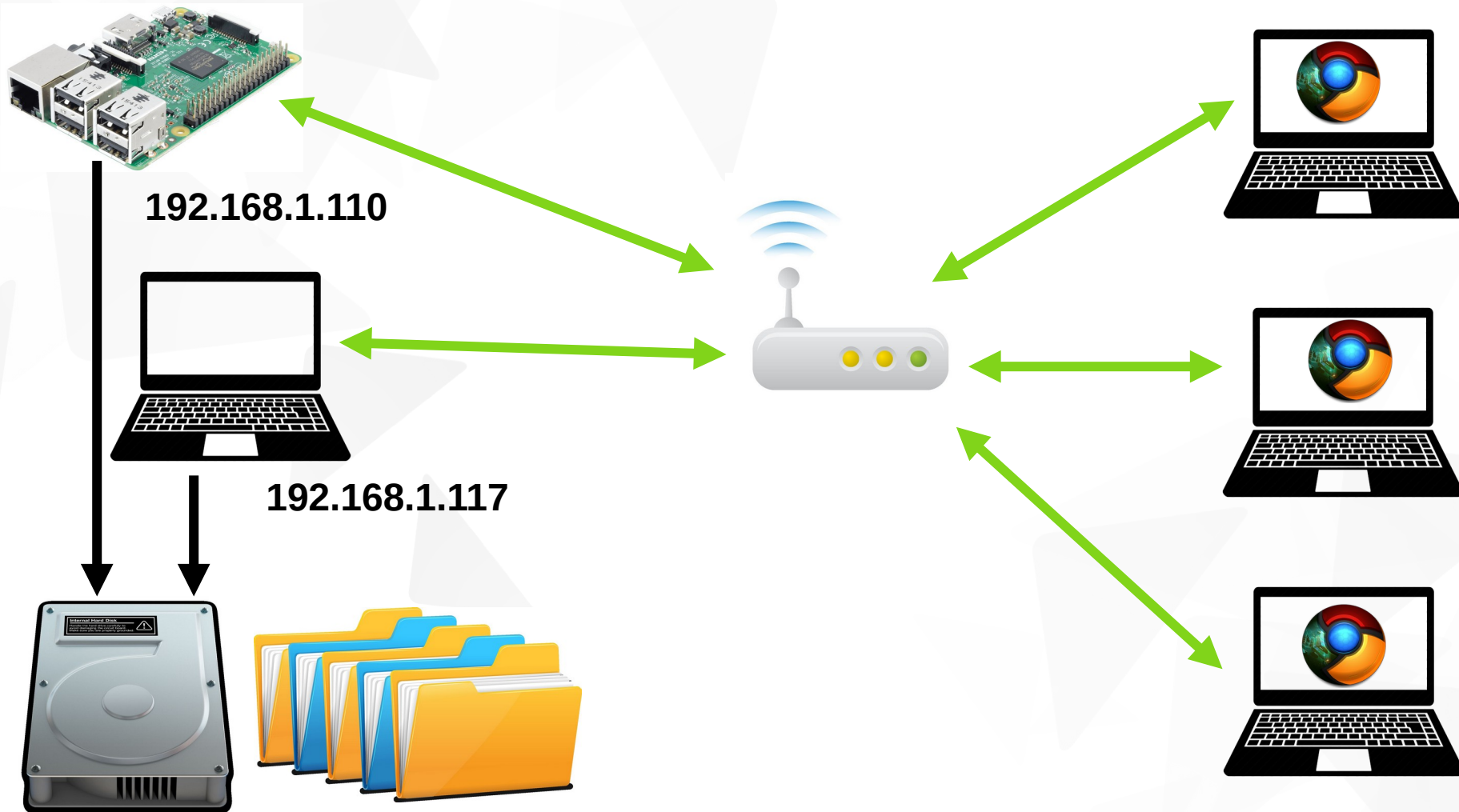
# Adresses IP – Ports de communication





# Python et le Web

- Création d'un **serveur Web** sur PC / Raspberry PI
- Accès à ce serveur par un **navigateur Web** : **URL**





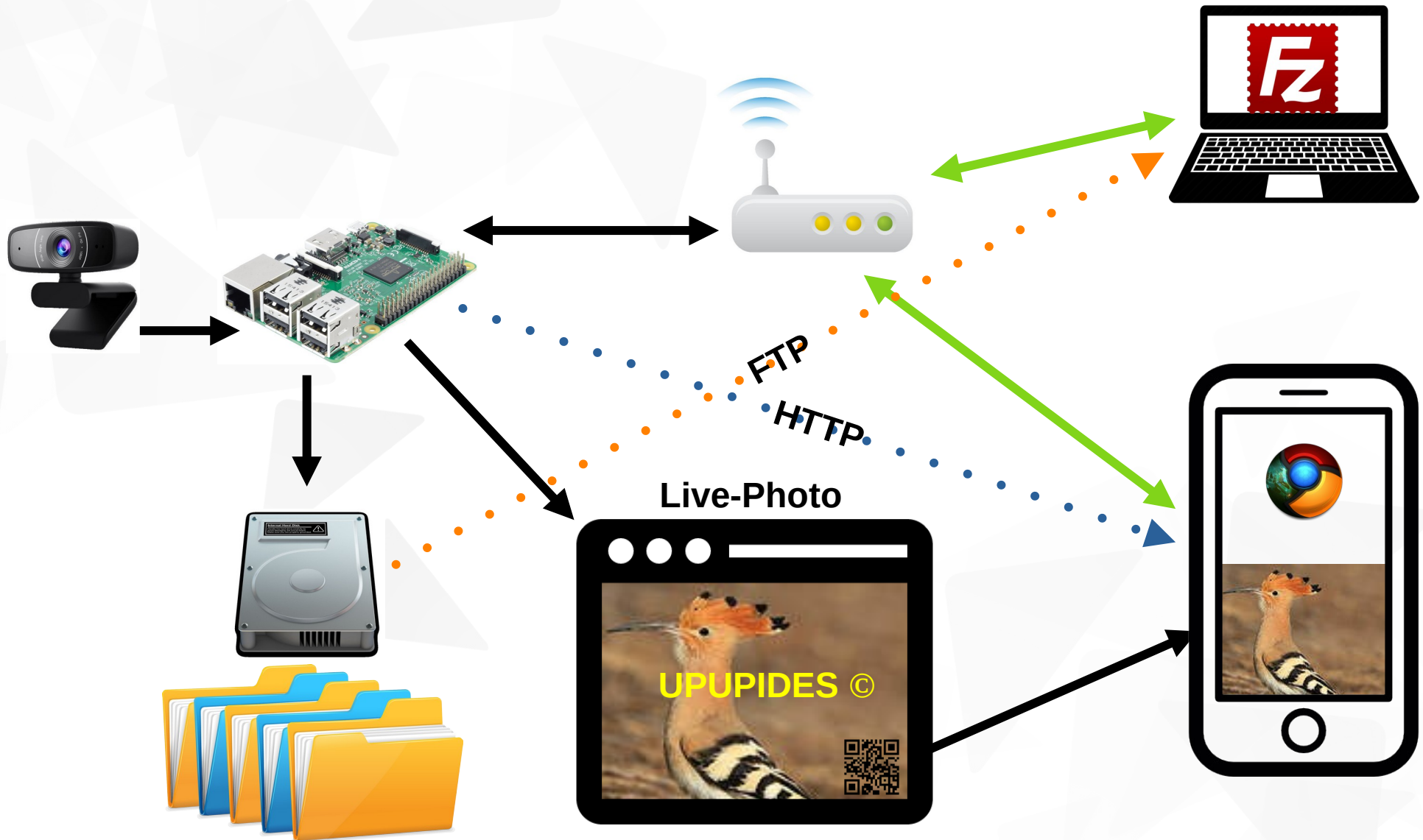
## Mixage !

- Intégration finale
- Bonus FTP
- Résumé - Mémo



# Intégration (finale ?)

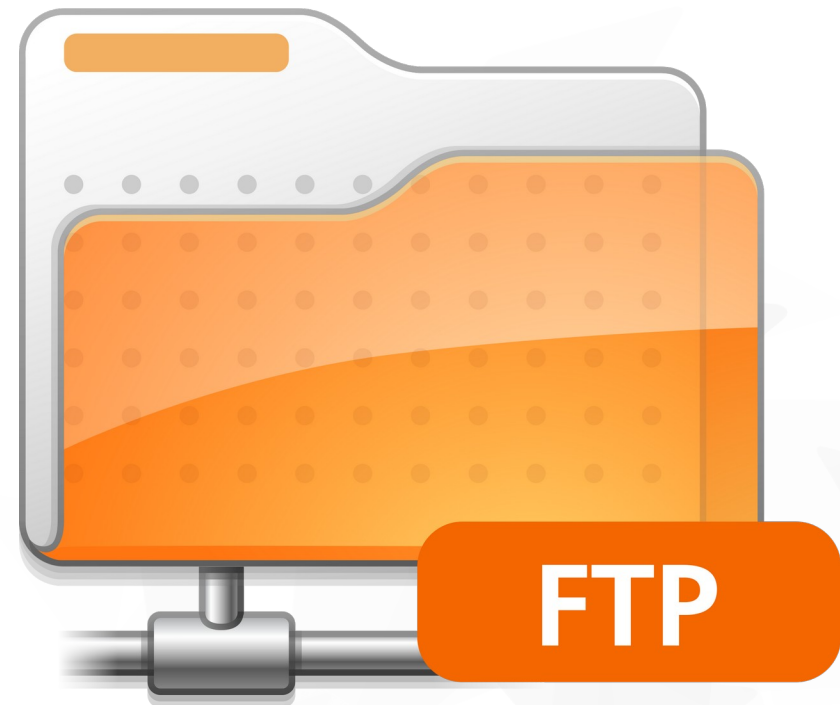
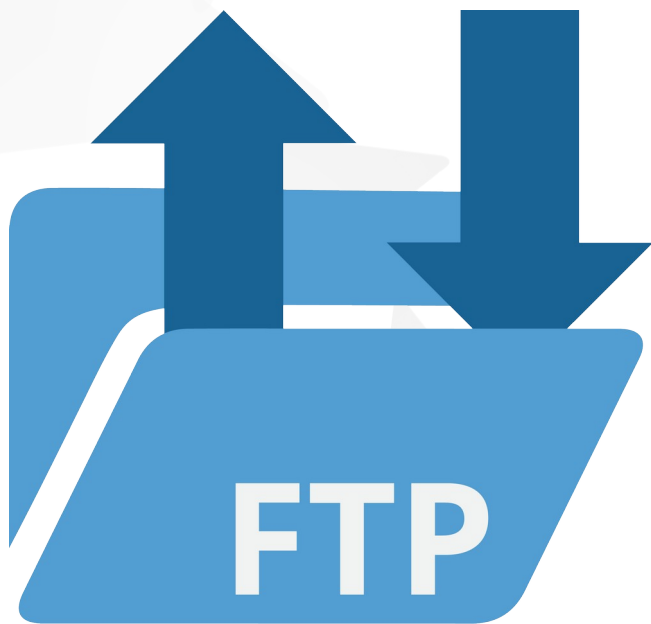
- On colle tous les **morceaux** ensemble !





# Bonus : FTP !

- Service **FTP** : **F**ile **T**ransfert **P**rotocol
- **Télécharger** les fichiers stockés = récupérer les données
- Logiciel :
  - **Client** FTP = navigateur de fichiers distants
  - Exemple : Filezilla





# Résumé – Mémo (1/2)

- Services :
  - **FTP** : **F**ile **T**ransfert **P**rotocol
    - Transférer des fichiers d'une machine à une autre
  - **HTTP** : **H**yper**T**ext **T**ransfer **P**rotocol
    - Demander des pages Web
  - **SMTP** : **S**imple **M**ail **T**ransfer **P**rotocol
    - Envoyer des emails
  - **POP** : **P**ost **O**ffice **P**rotocol
    - Recevoir des emails avec destruction sur serveur
  - **IMAP** : **I**nternet **M**essage **A**ccess **P**rotocol
    - Recevoir des emails avec copie sur serveur
- Rajouter un '**S**' pour la partie **sécurisée** = transfert d'informations **cryptées**, ex. : HTTPS, SFTP/FTPS, etc

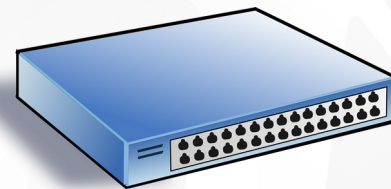
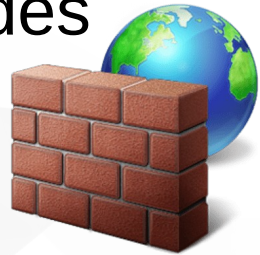
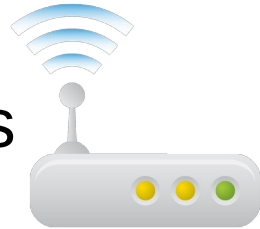






# Résumé – Mémo (2/2)

- **Routeur** = Passerelle :
  - Une machine connectée à un réseau demande à un routeur comment communiquer avec une autre machine
  - Permet de tracer des '**routes**' entre différents éléments connectés au réseau
  - Votre **BOX** à la maison : c'est son rôle !
- **Firewall** = Pare-feu :
  - Un élément qui **filtre** les communications : protection des ports ouverts ou fermés
  - Votre **BOX** à la maison : c'est son rôle !
- **Switch** = Multiprise :
  - Un élément qui permet de **connecter** plusieurs machines sur le même réseau
  - Votre **BOX** à la maison : c'est son rôle !



# FIN !

**Merci ...**

- ... pour avoir tenu jusqu'ici,
- ... de ne pas hésiter à poser des questions !

