



# Formation adh6 – partie 1

---

VIKTOR COLAS



# Sommaire

---

- Petit historique (g pa copié je jure)
- Cahier des charges
- Les technos
- Architecture backend/frontend
- Le backend
- Le frontend
- J'espère que vous avez écouté...
- Maintenant on travail...



# Histoire : adh4

**NE PAS UTILISER CETTE VERSION POUR LES ADHERENTS (SAUF ASSOC ET  
HEBERGEMENT WEB)**

(2258 adh dont 662 ayant une chambre | 37 assoc | 0 préadh)  
(442 clefs bitdefender libres)

Accueil | Adhérent | Aide | Logout

Recherche

Pré-inscription(s) (0)

**Rechercher un adhérent**

Vous pouvez utiliser des wildcards (\*) dans la recherche.

Login S2IA :


Prenom Nom :

Chambre :

MAC :

IP :

DNS :



**Fiche adhérent de : Guillaume ROSE**

Bienvenue guillaume

Accueil | Adhérent | WEB

**Informations Adhérent :**

- Login S2IA : rose\_gui
- Nom : Guillaume ROSE
- Email : Guillaume.Rose@telecom-sudparis.eu
- Promo : TINT
- Chambre :   vider la chambre?
- Téléphone : 1005
- est membre de minet-adherents:
- Date d'adhésion : 07/09/2009
- Date de départ :  /  /
- Commentaire: no comment.

**Les Paiements :**

Moyen de paiement	Pourquoi on paie ?	Montant	Date	Actions Possibles
Liquide	Internet	60	07/09/2009	<input type="button" value="Supprimer"/>

**Les ports :**

ADH4						SWITCH					
Port switch	Vlan	Rcom	Oid	Chambre	Port (phy)	Port (log)	Descr	802.1x	DVLAN	VLAN	Client MAC
0/43	41	1	10043	1005g	<input type="button" value="Success"/>	<input type="button" value="UP"/>	Admin Bender	<input type="button" value="Red"/>	41	41	00-16-D3-54-97-90

<https://www.slideshare.net/guillaumerose/prsentation-dadh5>



## Rechercher un adhérent

Identifiant DISI

Prénom Nom

Numéro de chambre

Adresse MAC

Adresse IPv4

Adresse IPv6

Rechercher



Bienvenue Viktor Colas  
Ces pages sont faites pour administrer simplement le réseau.

[Déconnexion](#)

Si vous rencontrez des problèmes pour vous identifier, n'hésitez pas à adresser un mail à [adh5@listes.minet.net](mailto:adh5@listes.minet.net)



703 adhérents à jour.  
Nombre d'IPs restantes :  
Vlan 41 : 155

# Histoire : adh5



# Histoire : adh5

---

- Début: 2010
- LDAP -> SQL
- Plusieurs composants: gestion de la tréso, des switchs, des adhérents, ...
- Ruby (Ruby on rails)

# Histoire : plus c'est long mieux c'est

From Anthony VEREZ <netantho@minet.net> ☆  
Subject [equipe] ADH666 6/16/12, 3:07 AM  
To equipe@listes.minet.net ★

Salut,

Bon avant de lâcher le troll de l'année :  
ADH5 est déjà très bien foutu. La qualité de code est très impressionnante, tu as fais un boulot formidable Guillaume.

L'idée d'ADH6(66) vient plutôt du fait que l'équipe actuelle n'ose pas toucher au code, même pour corriger les bugs. Ce qui est très gênant sur le moyen/long terme.

J'ai essayé de corriger des bugs, apporter quelques améliorations mineures mais je n'ai clairement pas le temps à moi seul (ou alors il faut que je me décharge d'autres tâches dans le même cas ...) pour me consacrer à ADH5.

2012 – merci frazew, je n'avais pas retrouvé le mail

thread Re: [equipe] réunion ADH6



Anatole Lefort

10 Jun 2016 12:16 p.m.

Salut,

Sur ton framadate, tu proposes 3dates, dont 2 qui sont déjà passées :p

On 06/10/2016 02:11 PM, capolrik wrote:

Search this list

Les stagiaires MiNET de cet été vont commencer à bosser sur adh6. C'est un projet qui demande de la préparation et un discussion sur le cahier des charge, et d'autres aspects techniques.

Pour trouvez une date, on remplit le framadate suivant SVP:

<https://framadate.org/oQ62qZhlst5m9yn5>

Si vous avez des conseils, des erreurs à ne pas commettre, n'hésitez pas à les donner sur ce pad:

[https://imagine.minet.net/pad/p/Commentaires\\_sur\\_ADH6](https://imagine.minet.net/pad/p/Commentaires_sur_ADH6)

Merci d'avance à tous

Cordialement

Capolrik



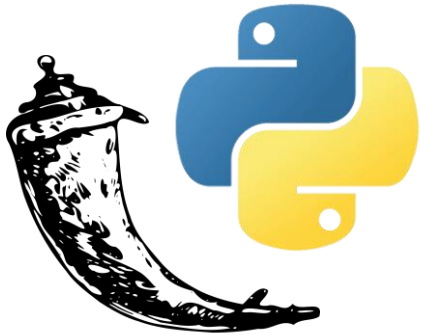
# Cahier des charges

---

- Une application ultra-modulaire
- Isolation backend-frontend : on m'a dit modularité
- Fiabilité : vous avez dit tests ?
- Simplicité : on ne réinvente pas la roue

# Technos

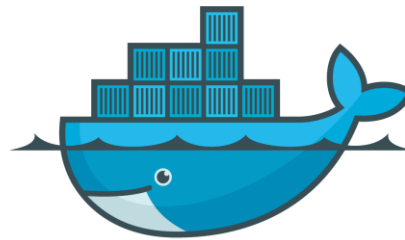
---



Flask



OPENAPI  
INITIATIVE



docker



BULMA





# Petit rappel

---

HTTP / HTTPS

Interfaces / Classes



# Petit rappel

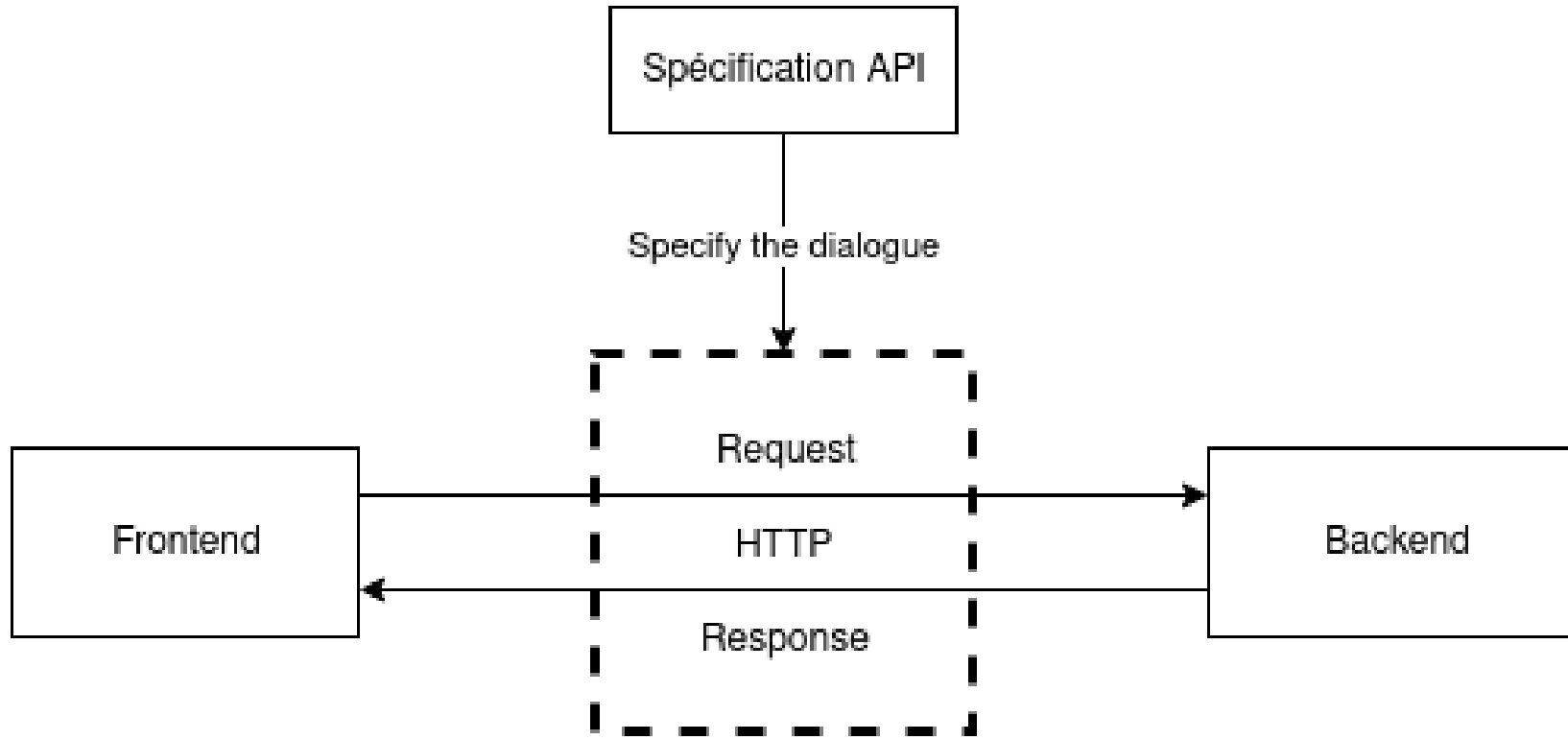
---

HTTP / HTTPS :

- Types de requêtes : POST, GET, PUT, PATCH, HEAD, DELETE
- Cookies

Interfaces / Classes





# Architecture backend/frontend



# Architecture backend/frontend

---

Plusieurs moyens de communication :

- REST : JSON (text)
- Protobuff : Binaires (Compilé)
- GraphQL : JSON/Binaires (Choix sur les valeurs à récupérer/envoyer)



JSON

```
1  [
2  {
3    "__typename": "Device",
4    "connectionType": "wired",
5    "id": 146808,
6    "ipv4Address": "157.159.47.13",
7    "ipv6Address": "2001:660:3203:407::a13",
8    "mac": "00-25-90-2A-26-92",
9    "member": {
10     "__typename": "Member",
11     "associationMode": "2021-10-27T16:25:48+00:00",
12     "departureDate": "2022-09-02",
13     "email": "viktor.colas@protonmail.com",
14     "firstName": "Viktor",
15     "id": 56920,
16     "ip": "157.159.192.235",
17     "lastName": "Colas",
18     "room": {
19       "__typename": "Room",
20       "description": "Enfer de MiNET",
21       "id": 2854,
22       "roomNumber": 666,
23       "vlan": {
24         "__typename": "Vlan",
25         "id": 12,
26         "ipv4Network": "157.159.47.0/24",
27         "ipv6Network": "2001:660:3203:407::/64",
28         "number": 47
29       }
30     },
31     "subnet": "10.42.14.160/28",
32     "username": "viktor"
33   }
34 }
35 ]
```

# Architecture backend/frontend

---



# Architecture backend/frontend

---

Principales actions :

- C
- R
- U
- D

# Architecture backend/frontend

---

Principales actions :

- Create
- R
- U
- D

# Architecture backend/frontend

---

Principales actions :

- Create
- Read
- U
- D



# Architecture backend/frontend

---

Principales actions :

- Create
- Read
- Update
- D



# Architecture backend/frontend

---

Principales actions :

- Create
- Read
- Update
- Delete



# Architecture backend/frontend

---

Principales actions :

- Create // POST
- Read // GET (HEAD)
- Update // PATCH/PUT
- Delete // DELETE



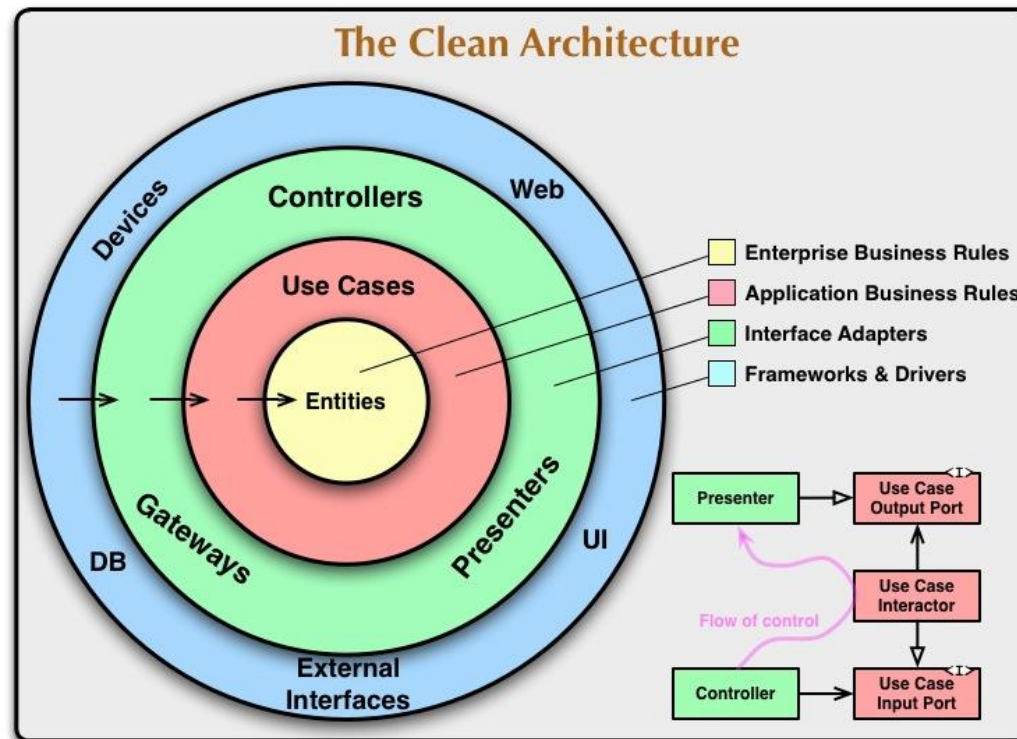
# Backend

---

- Gère les requêtes
- Souvent le plus complexe élément
- Doit pouvoir être mis à jour facilement
- Très modulaire

# Backend : Un oignon ?

---



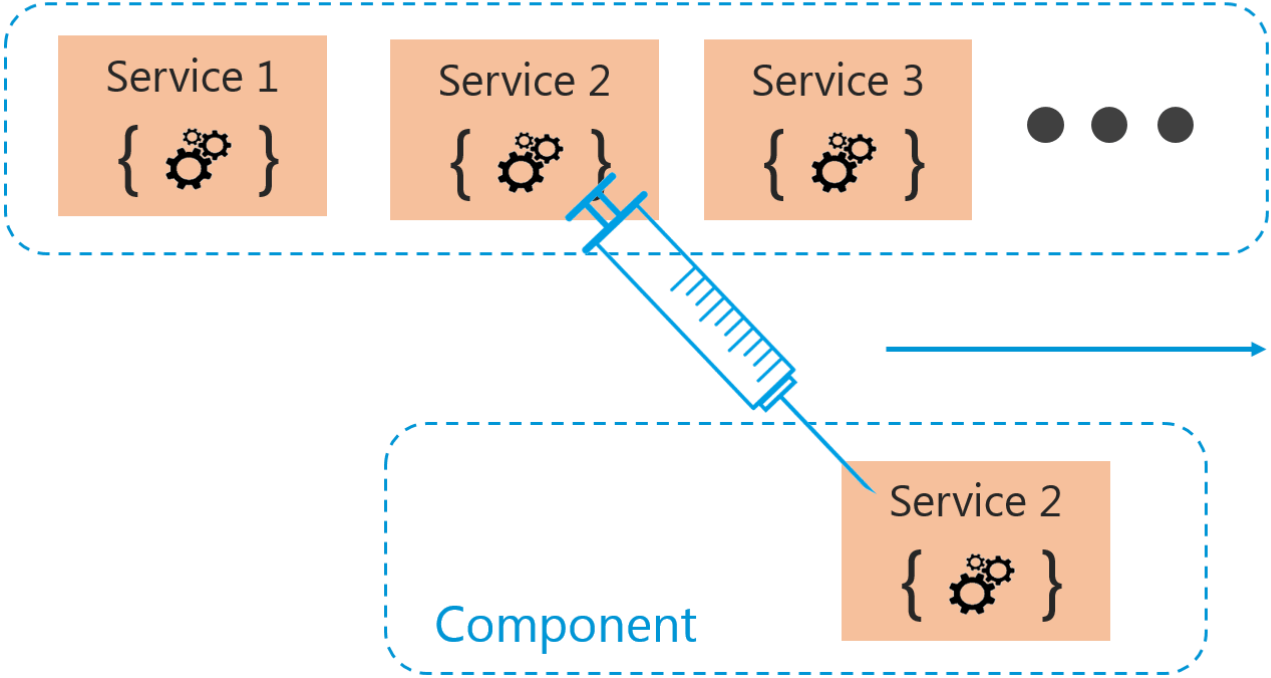
# Backend : Un oignon ?

---

```
1 from abc import ABCMeta, abstractmethod
2
3
4 class A(ABCMeta):
5     @abstractmethod
6     def test(self, t: str) -> int:
7         pass
```

- CLASSES ABSTRAITES : ABC

edureka!

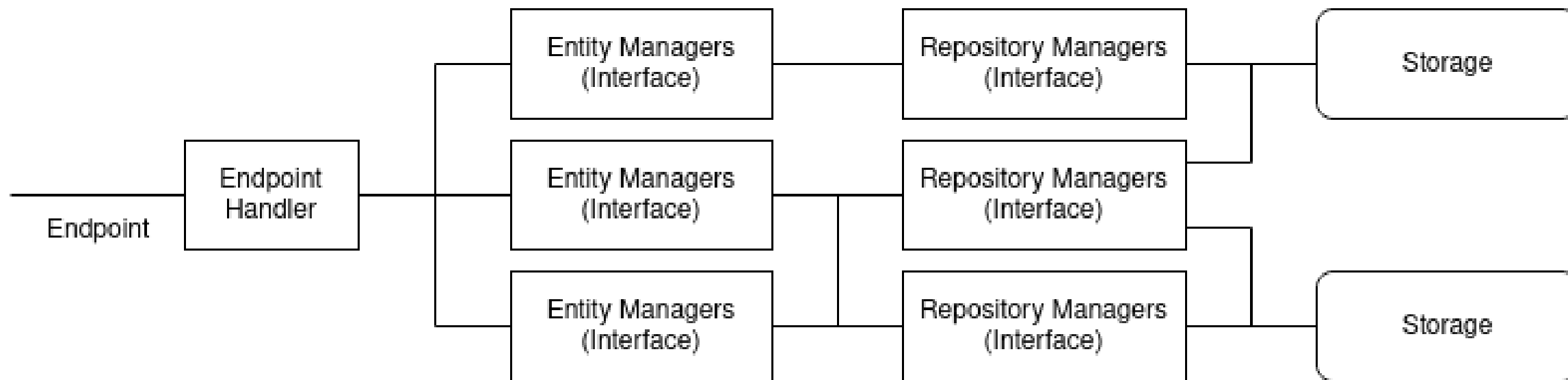


**Dependency  
Injection**

Backend

INJECTION DE  
DÉPENDENCE





Backend : Injection de  
dépendence





# Frontend

---

- Interface utilisateur
- Code le plus simple en général
- Mais le plus complexe en même temps
- Prendre en compte les actions utilisateurs

# Frontend : Vous avez dit javascript ?

---

- C'est de la merde javascript...
  - C'est vraiment pas ouf...
  - Nan mais on va pas se cacher que c'est dégueux...
  - Je crois que je suis encore trop gentil.
- => Utiliser du typescript développé par Microsoft et qui est opensource



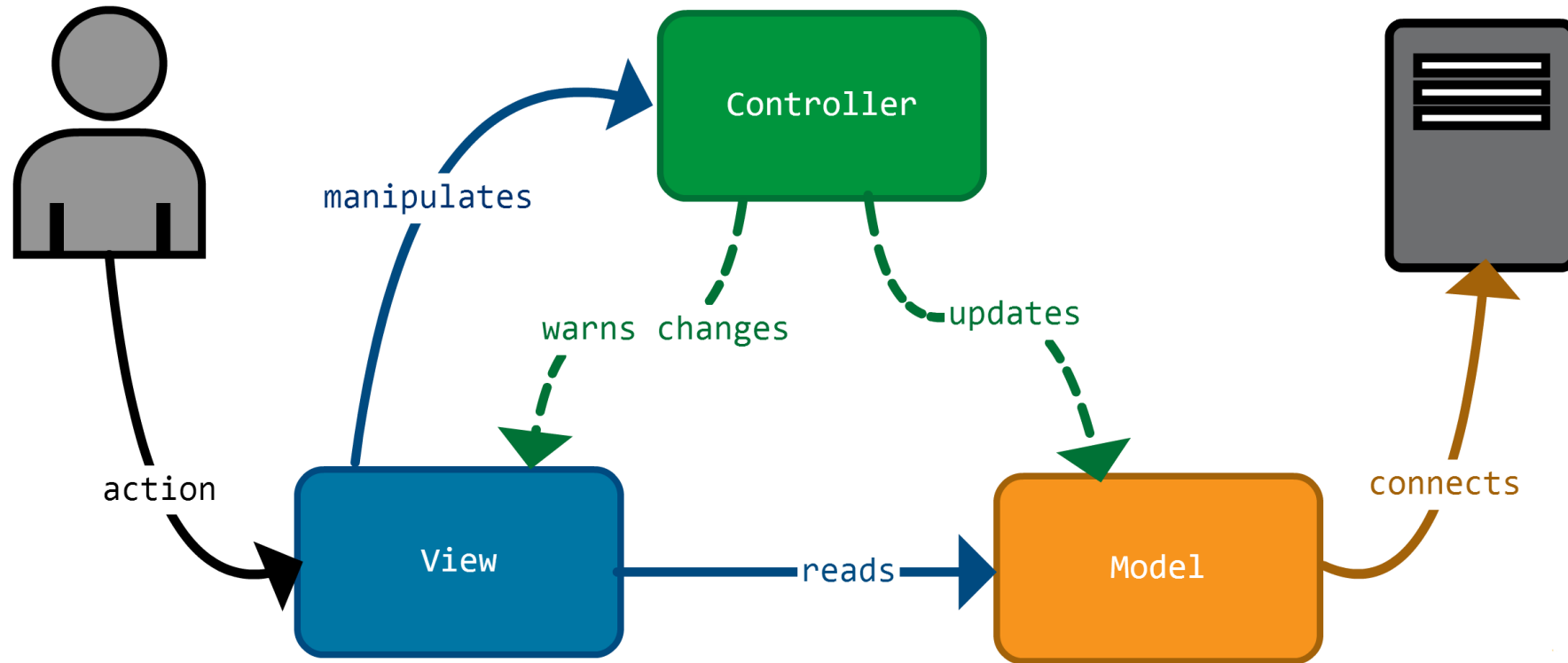
# Frontend : Typescript

---

Plusieurs frameworks:

- Angular
- React
- ViewJS





# Frontend : Modèle MVC

# Frontend : que j'eusse attendu ?

---

- L'asynchrone c'est le sang
- Il ne faut jamais une page bloquante sauf si elle est statique
- une bonne library de gestion de requêtes asynchrones : rxjs

# Frontend : RxJS

 26,793 Dependents

- 
- Observable : "Observables are lazy Push collections of multiple values"
  - Observer : "An Observer is a consumer of values delivered by an Observable"
  - Subject : "An RxJS Subject is a special type of Observable that allows values to be multicasted to many Observers"
  - Subscription : "A Subscription is an object that represents a disposable resource, usually the execution of an Observable"



# Frontend : Utilisateur

---

- C'est là que l'utilisateur interagit
- Faut que ce soit un minimum zoli
- Le premier qui met du JS pour des animations qui peuvent être fait en css je le tue
- 80% UX – 20% Code



J'espère que vous avez écouté

---



# Formation / Example

---

Tester l'injection de dépendance :

- Faire l'interface de gestion du stockage
- Définir la classe dans pour qu'elle hérite de l'interface
- En plus : Refaire de l'injection de dépendances dans un autre langage (comme Java ^^)

Tester les appels asynchrones :

- Frontend

[https://gitlab.minet.net/formation/sujets-tp/2022-adh6\\_partie1](https://gitlab.minet.net/formation/sujets-tp/2022-adh6_partie1)

