1 Exercice 1 : Projet TODOLIST

1 Création du projet

Créer un virtualenv, installer django et créer un projet nommé TODOLIST
(slide 9)

2 Création d'une première application

Créer l'application todo, le modèle Tache (description, date limite …) et activer l'application

(slide 14 → 19)

3 Activer le modèle tache dans l'admin

- https://docs.djangoproject.com/fr/1.8/ref/contrib/admin/#modeladmin-objects

4 Création des vues Liste et Détail

- La liste affichera le listing de toutes les taches
- Le détail affichera le détail d'une tache
- (slide 22 → 44)
- Utiliser les Class Based Views / Generic Views
- https://docs.djangoproject.com/fr/1.8/topics/class-based-views/generic-display/

5 Création des vues Ajout et Modification

- https://docs.djangoproject.com/fr/1.8/topics/class-based-views/generic-editing/

6 Liste des taches partagées entre utilisateurs

- quelle modification(s) apporter au modèle ?
- Slide 54
- https://docs.djangoproject.com/fr/1.8/topics/migrations/

7 Formulaire de filtrage sur la liste

- Créer un modèle Categorie lié à Tache en M2M
- Créer un formulaire de filtrage qui sera passé à la vue de liste
- Transmettre et traiter le contenu du formulaire pour effectuer le filtre

2 Exercice 2 : Base de données et Tests

1 Changer le backend de base de données

- Changer le backend de BDD pour mySQL
- https://docs.djangoproject.com/fr/1.8/ref/settings/#databases
- attention aux spécificités MySQL !
- https://docs.djangoproject.com/fr/1.8/ref/databases/#mysql-notes

2 Ecriture des tests

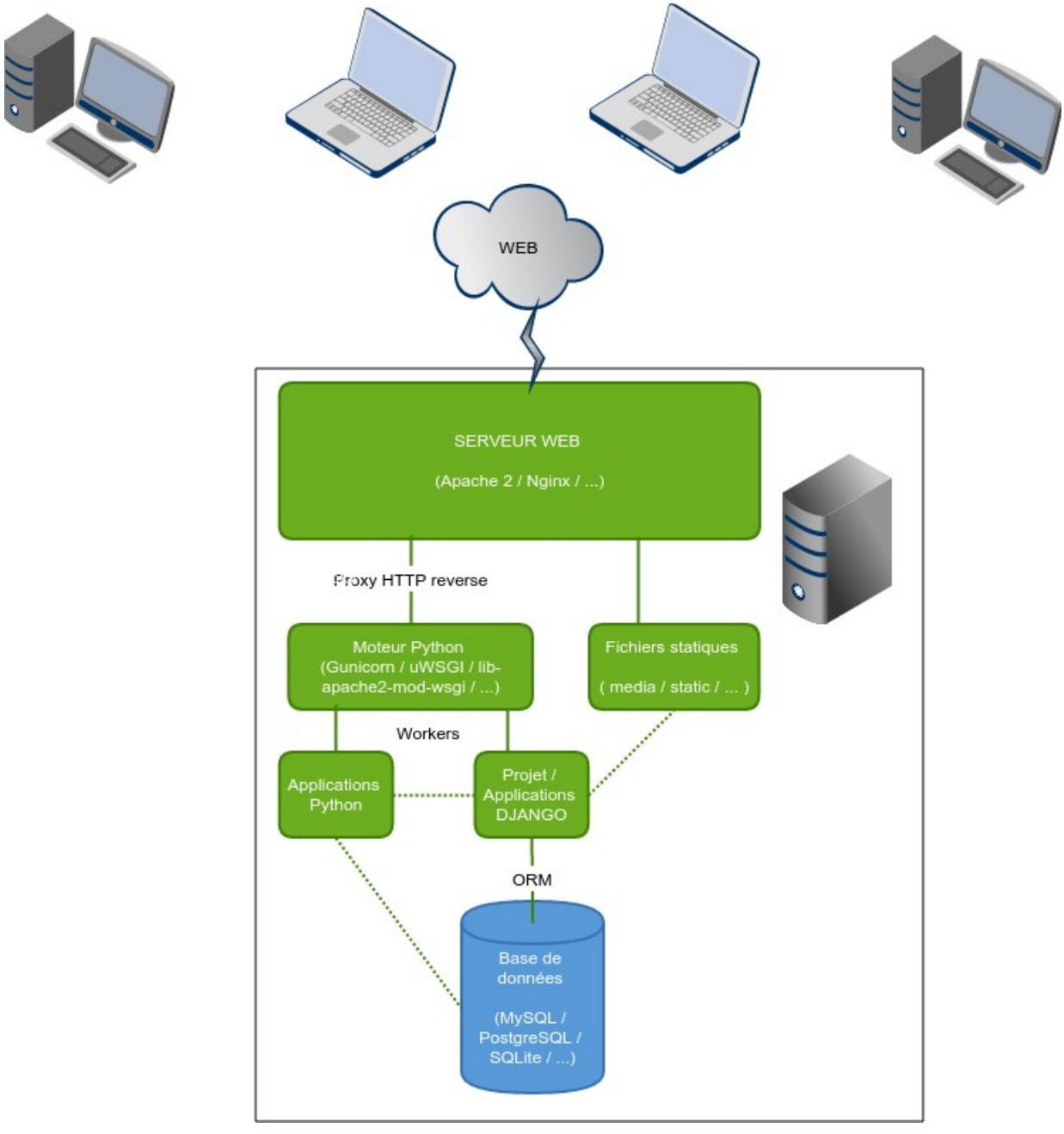
- https://docs.djangoproject.com/fr/1.8/topics/testing/overview/
- Ecrire et lancer les tests de création / modification / suppression des modèles Tache et Categorie
- Reflechir à comment tester un vue, un formulaire...

3 S'intégrer à une base de données existante

- https://docs.djangoproject.com/fr/1.8/howto/legacy-databases/
- créer une application 'client'
- configurer la base de données mysql existante
  - host : localhost
  - dbname : inspectdb
  - login : root
  - password : root

- utiliser inspectdb pour générer le modèle
- corriger le modèle, activer l'application client, ajouter les classes dans l'admin, tester

3 Exercice 3 : Déploiement et mise en production



1 Moteur python

utiliser gunicorn en lieu et place de ./manage.py runserver pour lancer l'application
en dev : --reload pour detecter tout changement de code python (couteux)

- https://docs.djangoproject.com/fr/1.8/howto/deployment/wsgi/gunicorn/
- http://docs.gunicorn.org/en/latest/install.html

2 Gestion du processus

utiliser supervisor pour controler automatiquement le lancement de gunicorn

- fichier LIBRARY/private/conf/LIBRARY-supervisor.conf à adapter et placer dans /etc/supervisor/conf.d/)
- sudo supervisorctl reread && sudo supervisorctl reload

3 Frontend WEB

- configurer apache en reverse proxy frontend de gunicorn
- fichier LIBRARY/private/conf/LIBRARY-apache.conf à adapter et placer dans /etc/apache2/sites-available/
- activer la conf : sudo a2ensite monfichier.conf
- recharger apache : sudo apache2 reload
- virtualhost → ne pas oublier de modifier le fichier /etc/hosts

4 Aller plus loin

1 Multi base de données

- Utile pour ne pas 'salir' une base de données existante avec des tables et données django
- https://docs.djangoproject.com/fr/1.8/topics/db/multi-db/#database-routers

2 Intégrer le projet django dans un autre contexte python

- Un projet django est un module python (presque) comme les autres
- Il nécessite une variable d'environnement DJANGO_SETTINGS_MODULE
- Intégré à un autre projet pour son ORM, il n'aura pas besoin d'être lancé par gunicorn etc.

3 Intéger et utiliser d'autres modules python dans le projet django

- pip install xxxx

4 Autres formations django

- http://makina-corpus.com/formations/formation-django-intermediaire
- http://makina-corpus.com/formations/formation-django-avance
- http://makina-corpus.com/formations/formation-django-integration
- http://makina-corpus.com/formations/formation-django-mise-en-production