

Projet de Web

Site de notation d'oeuvres

BEDIN Dylan
BORDE Antoine
SABOURIN Jérémy

Enseignant: Laurent RÉVEILLÈRE

Résumé du cahier des charges

- Principal :

- Inscription / Connexion
- Différents types d'oeuvres
- Rechercher une oeuvre
- Parcourir les oeuvres
- Fiche descriptive + Liste des critiques
- Créer une oeuvre
- Ajouter à sa collection + noter
- Ajouter à sa liste d'envies

- Secondaire :

- OAuth
- Profil
- Rechercher Utilisateur
- Demande d'ami
- Consulter profil ami
- Agenda des sorties

Etat du backlog en fin de projet

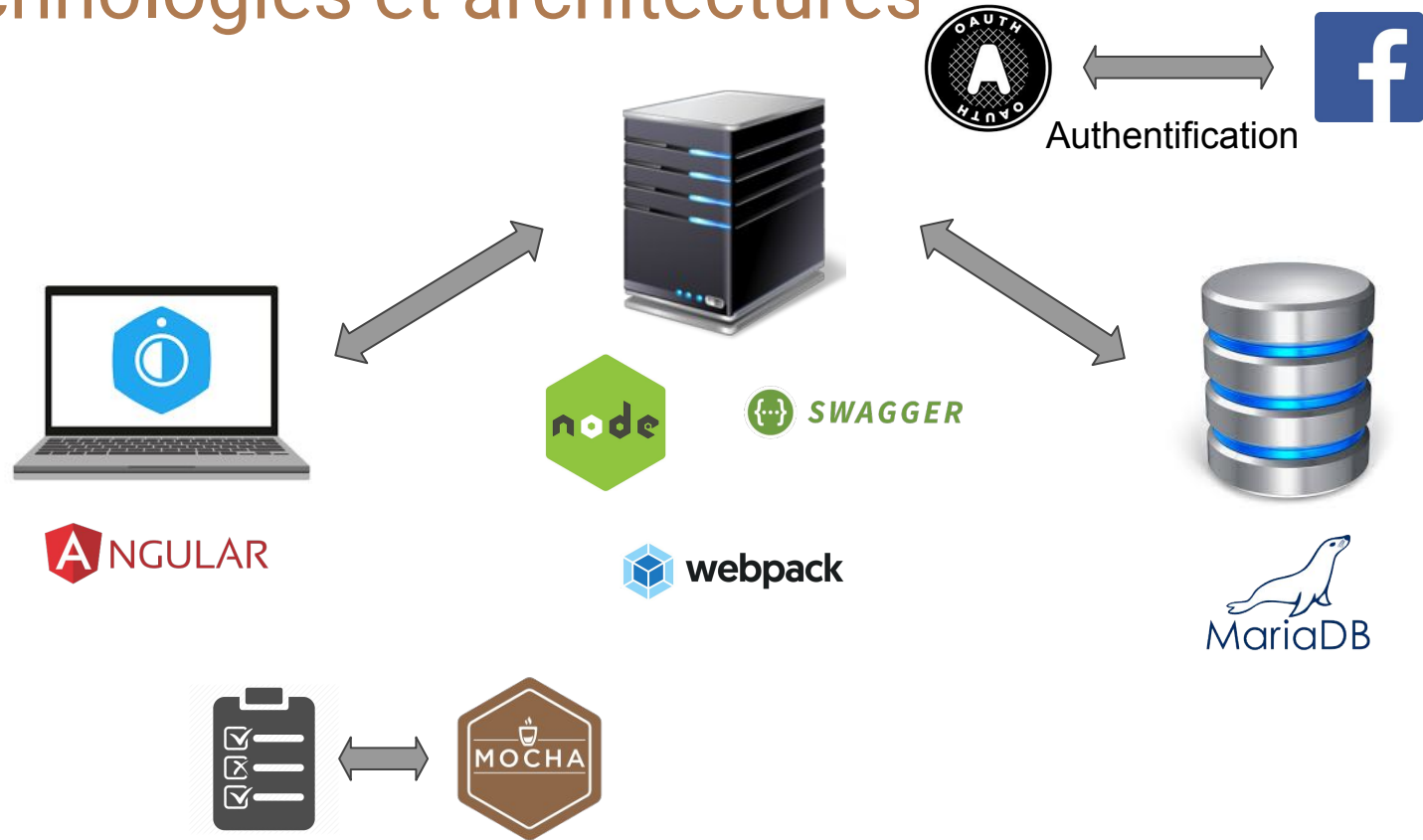
- Principal :

- ✓ Inscription / Connexion
- ✓ Différents types d'oeuvres
- ✗ Rechercher une oeuvre
- ✓ Parcourir les oeuvres
- ✓ Fiche descriptive + Liste des critiques
- ✓ Créer une oeuvre
- ✓ Ajouter à sa collection + noter
- ✓ Ajouter à sa liste d'envies

- Secondaire :

- ✗ OAuth
- ✓ Profil
- ✗ Rechercher Utilisateur
- ✗ Demande d'ami
- ✗ Consulter profil ami
- ✗ Agenda des sorties

Technologies et architectures



Technologie mise en place



Résumé de l'organisation

Semi méthode agile :

- Planification d'US et de tâches.
- Planification de sprints.
- Mise en place d'une chaîne d'intégration.
- Déploiement sur serveur distant.

Le Résultat

Projet Web

Connecté en tant que user

Mon Profil

Déconnexion

Parcourir

Ma collection

Ma liste d'envies

Musique

Livres

Films

Séries

Jeux Vidéo

Créer la fiche d'un album manquant

Titre	Auteur	Année de publication	
Flockaveli	Waka Flocka Flame	2010	Détails
Chicken Talk	Gucci Mane	2006	Détails
Le Monde Chico	PNL	2015	Détails
1994	Hamza	2017	Détails
000\$	Josman	2017	Détails
Batterie Faible	Damso	2016	Détails

Présentation de l'architecture via l'authentification



app.module.ts



path: 'login', component: LoginComponent

login.component.html



button -> login()

login.component.ts



authenticationService(username,password)

authentication-service.ts



http.post('/users/authenticate, {username, password})

User = require('./models/User.js')
User.findOne({username : username}, ...)

JS

models/User.js

userService.authenticate(username, encrypt(password))

JS

user.service.js



JS

routes/user.js



Système d'authentification

Authentification

- sessionStorage pour avoir accès à l'ID de l'utilisateur connecté
- Mots de passe cryptés à l'aide du module Node "Crypto"
- Système de garde des routes

La base de données

mongoose

elegant **mongodb** object modeling for **node.js**



Serveur nodeJS



Hébergement
pour la base de
données mongodb



Base de données

API REST et Swagger

API REST :

- Rédigée avec Swagger editor pour plus de facilité
- Difficultés rencontrées lors de la tentative de déploiement
- Déployé grâce à Swagger_ui sur un autre serveur node

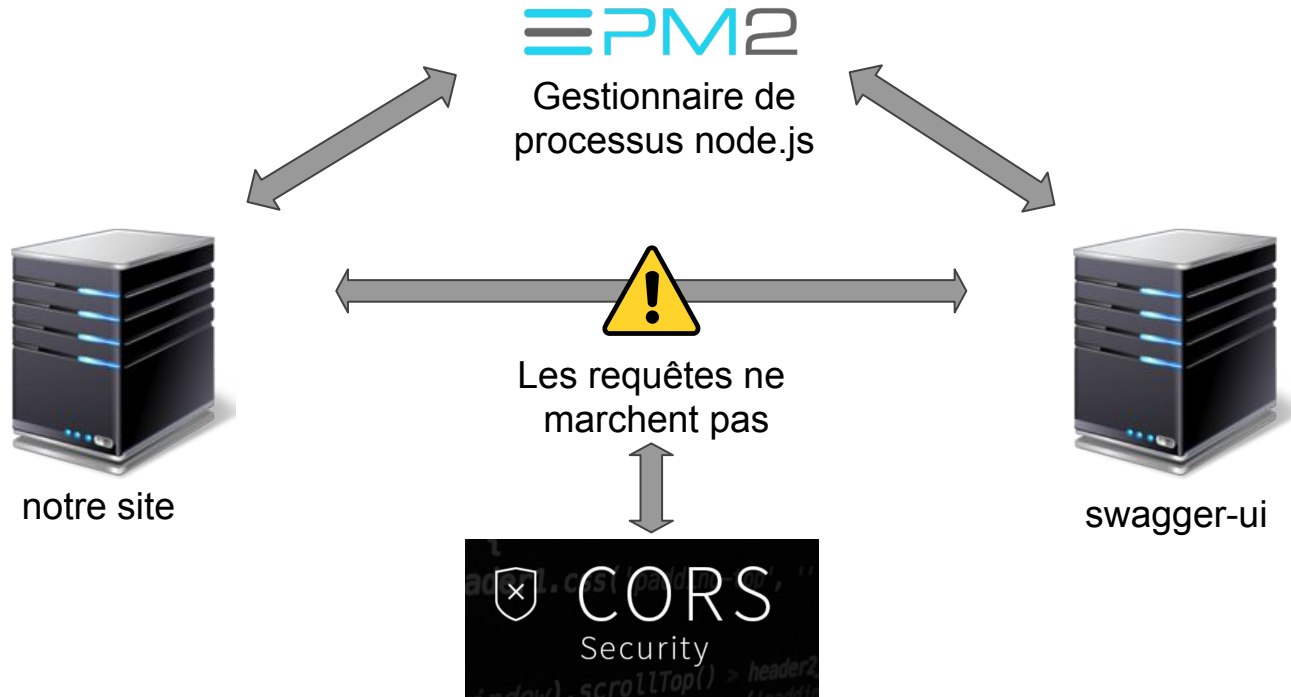
The screenshot displays a Swagger API interface with the following components:

- PUT /series/{SerieId}**: Update an existing Serie
- DELETE /series/{SerieId}**: Deletes a Serie
- Books** (expanded):
 - GET /books**: return list of Books
 - POST /books**: Add a new Book
 - GET /books/{BookId}**: Find Book by ID
 - PUT /books/{BookId}**: Update an existing Book
 - DELETE /books/{BookId}**: Deletes a Book
- Models** (expanded):
 - User**:

```
{
  id: string
  username: string
  firstName: string
  lastName: string
  email: string
  password: string($password)
}
```
 - Album**:

```
{
  _id: string
  title: string
  author: string
  description: string
  published_year: integer
}
```

Déploiement de l'application et de l'API



Déploiement distant, chaîne d'intégration et tests

- Déploiement sur serveur distant avec une chaîne d'intégration Jenkins.
- Mise en place de tests unitaires automatisés à chaque déploiement avec newman.
- Mise en place de tests e2e avec Protractor, Jasmine et Selenium.



+



Jenkins



Conclusion

Les apports :

- Découverte de nombreuses technologies web récente et actuel dans le monde professionnel...
- Mise en place d'une architecture relativement propre et complète

Les inconvénients :

- ... Mais survolé en trop peu de temps ce qui implique une maîtrise approximative
- Un projet trop ambitieux par rapport pour le temps disponible
- Une approche du début de projet maladroite

Possibilités d'évolution :

- Utilisation d'API externes (Facebook, Agenda, BD d'oeuvres...)
- Fonctionnalités sociales (système d'amis, chat...)