

**Projet SI-SOR – Mars 2024**  
**Location de films**

**Évaluation :**  
**Démonstration à effectuer**  
**Rapport d'une page à faire**  
**(liste des technologies utilisées et liste des fonctions mises en oeuvre)**  
**Sources à déposer sur Moodle au plus tard le vendredi 22 mars**

**Le travail peut être fait en binôme**

Le sujet : on veut créer un site Web pour louer des films. Un film est décrit par une notice formée d'au moins un nom, une date de sortie, une description textuelle et des photos. Les personnes qui consultent le site peuvent ajouter des commentaires sur chaque film. On garde également une trace des consultations.

Le but du projet est de construire une application n-tiers qui utilise des éléments étudiés dans le cadre des UE SOR et SI ; application React ou Vue, serveur Apache, serveur NodeJS/express/JWT, serveur Tomcat/Servlet/Spring/JPA.

**Fonctionnalités à mettre en oeuvre :**

Partie publique

- Consultation des films proposés (accès libre)
- Location d'un film (gestion d'un panier)
- Création de compte, identification des clients
- Ajout de commentaires sur un film, textes et/ou photos. L'ajout est possible seulement après identification.

Partie administration

- Ajout et suppression de films
- Modération des commentaires sur les films
- Gestion des comptes
- Historique des locations de films

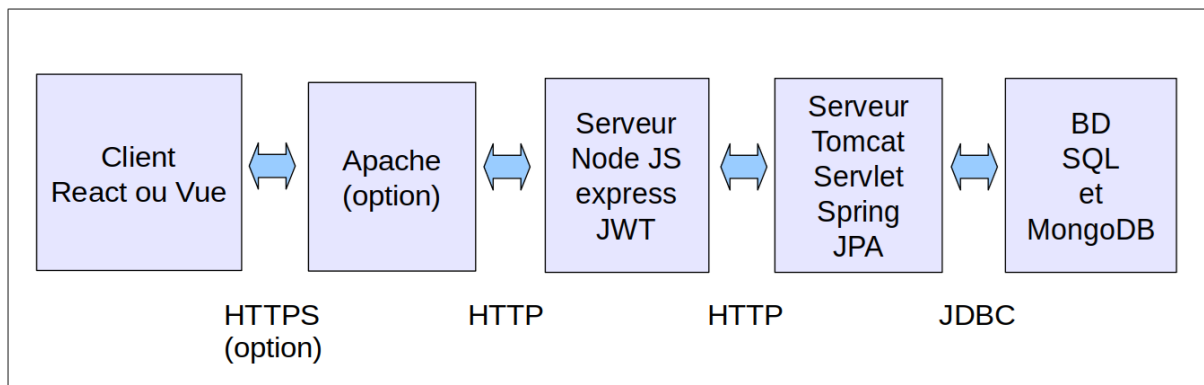
Option

- Chat pour échanger sur le site (utiliser les Websockets)

**Technologies à utiliser :**

- Client (front) : **React ou Vue**
- Serveur n°1 (back, en option) : Apache, gestion SSL/TLS (HTTPS)
- Serveur n°2 (back) : **NodeJS** (express / JWT), vérification des droits d'accès
- Serveur n°3 (back) : **Tomcat** (Servlet, Spring, JPA), accès aux bases de données.
- Serveur de base de données : **MongoDB et SQL**

## Architecture à mettre œuvre



Le client React ou Vue communique en HTTPS avec un serveur Apache. Le serveur Apache retransmet certaines requêtes à un serveur Nodejs.

Remarque : intégrer Apache seulement à la fin du projet, en option. Dans un premier temps, le client React ou Vue pourra communiquer en HTTP avec le serveur Nodejs.

Le serveur Nodejs est chargé de sécuriser l'application WEB. Il vérifie que les requêtes reçues sont autorisées. Il peut accéder directement à la base de données pour effectuer cette vérification. Une fois la vérification faite, il retransmet la requête au serveur Tomcat ou indique au client que l'opération est impossible.

Le serveur Tomcat gère tous les autres accès à la base de données. On peut utiliser les Servlets et/ou Spring.

## Organisation et évaluation

- Le travail peut être effectué en binôme
- Déposer sur le cours Moodle « Mise en pratique accompagnée (M1-TIIL) » une archive contenant :
  - Un résumé d'environ une page donnant la liste des fonctionnalités mises en œuvre et les technologies utilisées
  - Le schéma de la base de données (MySQL et MongoDB)
  - Les sources de l'application
- L'**évaluation** se fera sur ce dépôt et surtout sur une **démonstration à la fin du projet**.

## Critères d'évaluation

- La fiabilité de l'application. La **gestion des erreurs** doit être traitée avec soin (cas des formulaires mal remplis, ...)
- Le niveau fonctionnel atteint
- La qualité du jeu de tests
- La qualité du code (propreté, fiabilité)