

# Document d'architecture – Étude de cas INNOTECHFUSION

---

## 1. Vue d'ensemble du projet

L'application InnotechFusion permet de gérer un processus de vote simple pour les membres d'une association, avec une interface Angular, un backend Java 21 Spring Boot packagé en WAR, et une base MySQL.

La V1 permet :

- De visualiser tous les membres de l'association.
- De voter une seule fois de manière irréversible.
- De tracer le fait qu'un utilisateur a voté.

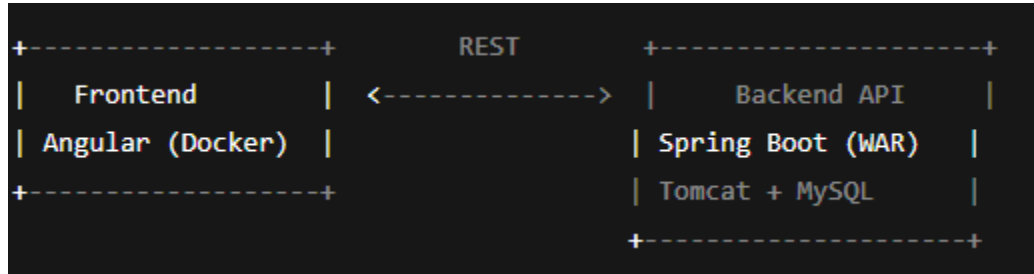
La V2 envisagée ajoutera :

- La gestion de plusieurs scrutins.
  - Le vote anonyme.
- 

## 2. Architecture technique

- **Frontend** : Angular 17, Angular Material, tests Jasmine/Karma
- **Backend** : Java 21, Spring Boot 3 (WAR), Tomcat 10, tests JUnit
- **Base de données** : MySQL 8
- **Conteneurisation** : Docker
- **CI/CD** : GitLab CI (build + test + dockerisation)

Vue logique simplifiée :



---

### 3. Schéma base de données – Version 1

Table : user

- id (PK)
- firstname
- lastname
- birthdate

Table : vote

- id (PK)
- user\_id (FK -> user.id)
- candidat\_id (FK -> candidat.id)
- vote\_date (timestamp)

---

### 4. Schéma base de données – Prévision V2

```
Table : user
- id (PK)
- firstname
- lastname
- birthdate

Table : election
- id (PK)
- name
- start_date
- end_date

Table : vote
- id (PK)
- election_id (FK)
- vote_hash (string, encodé anonymement)
- vote_date (timestamp)
```

🔒 Dans la V2, l'identité du votant ne sera pas conservée : `vote_hash` sera une empreinte anonyme générée côté frontend.

## 5. Évolutivité prévue – Version 2

### À ajouter :

- Table `election` pour gérer plusieurs scrutins
- Séparation des votes par scrutin
- Encodage anonyme des votes (`hash`)
- Interface de visualisation des résultats par scrutin
- Évolution du modèle frontend pour gérer plusieurs élections
- Éventuellement, un rôle "admin" pour gérer les scrutins