

Comparatif des Solutions Techniques

Contexte et objectifs

ORION souhaite développer un réseau social innovant dédié aux développeurs, appelé **Monde de Dév (MDD)**. L'objectif principal est de faciliter la mise en relation entre développeurs en recherche d'emploi, par le biais d'un espace d'échange et de collaboration autour de sujets techniques.

Avant un déploiement à large échelle, un MVP sera réalisé pour tester en interne les fonctionnalités clés : abonnement à des sujets, publication d'articles, commentaires et fil d'actualité personnalisé.

Description fonctionnelle du MVP

Gestion des utilisateurs : inscription, authentification, profil minimal

Sujets (tags) : possibilité pour l'utilisateur de s'abonner à des sujets de programmation (JavaScript, Java, Python, Web3, etc.)

Fil d'actualité : affichage chronologique des articles liés aux sujets abonnés

Articles : création, modification, suppression d'articles par les utilisateurs

Commentaires : possibilité de poster des commentaires sur les articles

Choix techniques proposés

Composant	Technologie choisie	Justification
Backend API	Java + Spring Boot	<ul style="list-style-type: none"> - Écosystème mature et robuste - Support natif pour REST, sécurité (Spring Security) - Facilité d'intégration avec MySQL et gestion des transactions - Large communauté et documentation fiable
Frontend	Angular + Angular Material	<ul style="list-style-type: none"> - Framework SPA performant et structuré - Composants UI préconçus et adaptés (Angular Material) - Support Typescript pour une meilleure qualité de code - Expérience utilisateur fluide et responsive
Base de données	MySQL	<ul style="list-style-type: none"> - SGBD relationnel éprouvé, performant pour données structurées - Large adoption en entreprise - Facilité de gestion et administration - Support natif avec Spring Data JPA
Hébergement / Déploiement	Serveurs internes pour MVP (option Cloud à l'avenir)	<ul style="list-style-type: none"> - MVP déployé en interne pour limiter les coûts - Contrôle total sur les données et l'environnement - Possibilité d'évolution vers cloud public pour montée en charge

Alternatives technologiques envisagées

Dans le cadre du choix de la solution technique pour MDD, plusieurs alternatives ont été étudiées pour le backend, le frontend et la base de données. Ces options pourraient être envisagées selon les compétences des équipes, les contraintes de délais ou les évolutions futures du projet.

Alternatives pour le back-end		
Technologie	Avantages	Inconvénients
Python (Django / Flask)	<ul style="list-style-type: none"> - Rapidité de prototypage grâce à la simplicité du langage - Écosystème riche pour le développement web - Bonne intégration avec les bases de données relationnelles et NoSQL 	<ul style="list-style-type: none"> - Performances généralement inférieures à Java pour les applications à haute charge - Moins adapté aux applications fortement transactionnelles
.NET (C# avec ASP.NET Core)	<ul style="list-style-type: none"> - Performance élevée, proche de Java - Outils et IDE puissants (Visual Studio) - Bon support multiplateforme avec .NET Core 	<ul style="list-style-type: none"> - Courbe d'apprentissage plus élevée si équipe non familière - Dépendance possible à l'écosystème Microsoft

Alternatives pour le front-end		
Technologie	Avantages	Inconvénients
React.js	<ul style="list-style-type: none"> - Très populaire avec une grande communauté - Flexibilité et modularité - Large écosystème de bibliothèques tierces - Facilité d'intégration avec divers styles et outils 	<ul style="list-style-type: none"> - Nécessite souvent plus de configuration que Angular - Moins "clé en main" que Angular Material pour les composants UI
. Vue.js	<ul style="list-style-type: none"> - Syntaxe simple et facile à apprendre - Performant et léger - Intégration progressive possible dans des projets existants 	<ul style="list-style-type: none"> - Communauté plus petite que React ou Angular - Moins de ressources et de bibliothèques comparé à React

<i>Alternatives pour la base de données</i>		
Technologie	Avantages	Inconvénients
PostgreSQL	- SGBD relationnel très puissant et open source - Fonctionnalités avancées (JSONB, recherche textuelle)	- Peut nécessiter une courbe d'apprentissage plus élevée que MySQL
MongoDB (NoSQL)	- Flexible pour données semi-structurées ou non structurées - Adapté pour les évolutions rapides du schéma	- Moins adapté aux relations complexes et aux contraintes d'intégrité

Conclusion

Les alternatives présentées offrent des avantages spécifiques qui pourraient être exploités en fonction des besoins futurs, des profils des développeurs ou de l'évolution technologique. Cependant, pour la phase MVP de Monde de Dév, le stack Java/Spring, Angular et MySQL reste le choix le plus équilibré en termes de rapidité de mise en œuvre, robustesse et évolutivité.

Choix du système de gestion de versions (Version Control)

Afin d'assurer un développement structuré, collaboratif et sécurisé pour le projet Monde de Dév (MDD), l'utilisation d'un système de gestion de versions est indispensable.

Solution retenue : Git avec hébergement sur GitHub

Justification :

- Référence mondiale : GitHub est la plateforme de gestion de code source la plus largement adoptée par les développeurs et entreprises.
- Fonctionnalités collaboratives : Pull requests, code reviews, discussions, gestion fine des branches et protections de branche.
- Intégration continue : Utilisation de GitHub Actions pour l'automatisation des tests, des déploiements et autres workflows CI/CD.
- Visibilité et gestion des projets : Suivi des tickets (issues), gestion de projet Kanban, wiki, discussions.
- Facilité d'intégration : Compatible avec les outils externes comme Slack, Trello, et avec les IDE comme Visual Studio Code.

Choix d'une organisation privée :

Les dépôts relatifs au projet MDD seront hébergés dans une organisation privée GitHub afin de garantir la confidentialité du code tout en bénéficiant de l'infrastructure cloud de GitHub.

Architecture proposée

Backend : API RESTful exposée par Spring Boot, sécurisée via JWT et Spring Security.

Frontend : Application Angular consommant l'API, utilisant Angular Material pour l'interface utilisateur.

Base de données : Modèle relationnel avec tables utilisateurs, sujets, articles, commentaires, et relations (abonnements).

Communication : JSON sur HTTP/HTTPS entre frontend et backend.

Avantages et risques

Aspect	Avantages	Risques / Limites
Java + Spring Boot	<ul style="list-style-type: none">- Maturité et stabilité- Performances élevées- Écosystème riche	<ul style="list-style-type: none">- Temps d'apprentissage initial pour développeurs non familiers avec Spring
Angular + Angular Material	<ul style="list-style-type: none">- UI riche et responsive- Productivité accrue grâce aux composants	<ul style="list-style-type: none">- Complexité parfois élevée pour les débutants Angular
MySQL	<ul style="list-style-type: none">- Fiabilité et scalabilité- Facilité de maintenance	<ul style="list-style-type: none">- Pas adapté aux données très non structurées ou volumineuses (big data)
Déploiement interne	<ul style="list-style-type: none">- Sécurité renforcée- Contrôle complet	<ul style="list-style-type: none">- Scalabilité limitée au début- Maintenance et infrastructure à gérer

Perspectives d'évolution

Fonctionnalités à ajouter : système de messagerie privée, recommandations intelligentes, profils enrichis, gamification.

Architecture : passage à une architecture microservices si montée en charge.

Déploiement : migration vers un cloud public (AWS, GCP, Azure) pour scalabilité et haute disponibilité.

Base de données : ajout d'une base NoSQL pour certains cas d'usage (ex. recherche rapide, tags).

Conclusion

Le choix de technologies **Java/Spring** pour le backend, **Angular/Angular Material** pour le frontend, et **MySQL** pour la base de données répond efficacement aux besoins du MVP de Monde de Dév. Ce stack offre un bon équilibre entre rapidité de développement, robustesse, maintenabilité et expérience utilisateur, tout en permettant une évolution facile vers des versions plus complètes.