



Origine	Rédacteur	Date de réunion	Id	Version	
iD3i	Amaury Betaouaf	06/06/2025	CER-CDC-IDE -06062025	1	
Sujet	Cahier des charges - IDE interne Front				



Cahier des charges - IDE interne Front

TABLE DES MATIERES

1. Contexte et Objectifs.....	2
2. Objectifs Fonctionnels	2
3. Objectifs Non Fonctionnels	3
4. Livrables et Planning.....	4
5. Technologies envisagées	4
6. Équipe projet	5
7. Critères techniques détaillés	5

Origine	Rédacteur	Date de réunion	Id	Version	
iD3i	Amaury Betaouaf	06/06/2025	CER-CDC-IDE -06062025	1	
Sujet	Cahier des charges - IDE interne Front				

1. CONTEXTE ET OBJECTIFS

Notre objectif est de développer un **IDE interne** permettant de produire efficacement les écrans du projet CER. Cet outil doit être performant, facile à utiliser et adaptable à notre contexte métier. Nous visons un outil final utilisable en production d'ici 3 mois.

Cet IDE sera alimenté par des composants générique développé « à la main ».

Référence de performance :

Nous souhaitons nous inspirer de sites comme <https://www.mcmaster.com/> pour garantir une interface fluide et rapide à utiliser.

2. OBJECTIFS FONCTIONNELS

2.1. Éditeur de code intégré

- Utiliser **CodeMirror.net** ou une alternative pour un éditeur moderne, performant et extensible.
- Support syntaxique pour nos langages front (HTML, CSS, JS/TS) et éventuellement pour le code métier (via transpiler).

2.2. Gestion des écrans et composants front

- Pouvoir :
 - Créer des pages/composants front rapidement.
 - Visualiser les pages/composants en live preview.
 - Organiser les écrans en modules ou fenêtres de code.
 - Relier les composants au **code métier** de manière fluide.

2.3. Transpiler métier


- Développer ou intégrer un transpiler permettant de transformer le code métier (ex. DSL interne) en code front exploitable.
- Objectif : faciliter l'intégration métier sans perturber le cycle de développement.

2.4. Collaboration temps réel

- Intégrer une **communication WebSocket** pour permettre l'édition collaborative (plusieurs utilisateurs sur le même projet).

2.5. Interface web et/ou plugin

- Première version **full web** (accessible depuis un navigateur).
- Possibilité d'envisager un **plugin VSCode**.

Origine	Rédacteur	Date de réunion	Id	Version	
iD3i	Amaury Betaouaf	06/06/2025	CER-CDC-IDE -06062025	1	
Sujet	Cahier des charges - IDE interne Front				

2.6. Navigation et UX

- Inspirée de McMaster.com : chargement rapide, transitions fluides, menus ergonomiques, bonne gestion des volumes de données.
- Préloading / prerendering pour optimiser les temps de chargement.

3. OBJECTIFS NON FONCTIONNELS

3.1. Performance


- Temps de chargement rapide.
- Navigation fluide même avec des pages complexes.
- Utiliser des outils de mesure (exemples : Lighthouse, Web Vitals pour les outils Google).

3.2. Sécurité

- Authentification basique pour accéder à l'IDE.
- Sécurisation des échanges WebSocket.
- Limiter les accès aux seuls utilisateurs autorisés.

3.3. Évolutivité

- Architecture modulaire pour pouvoir ajouter de nouvelles fonctionnalités.
- Prévoir un plugin ou API pour interagir avec d'autres outils internes.


Origine	Rédacteur	Date de réunion	Id	Version	
iD3i	Amaury Betaouaf	06/06/2025	CER-CDC-IDE -06062025	1	
Sujet	Cahier des charges - IDE interne Front				

4. LIVRABLES ET PLANNING

Étape	Livrable	Date cible
1. Analyse technique	Document de choix technologiques (CodeMirror vs alternatives, frameworks front, architecture)	13 juin 2025
2. POC léger	Prototype fonctionnel minimal : éditeur de code, preview, transpiler basique	Fin juin 2025
3. Version alpha	IDE web utilisable en interne : création de composants, transpilation, collaboration	Mi-juillet 2025
4. Version bêta	Version finalisée, testée, optimisée (UX inspirée McMaster)	Fin août 2025
5. Mise en production	Déploiement interne et formation	Septembre 2025

5. TECHNOLOGIES ENVISAGEES

Besoin	Technos proposées
Éditeur de code	CodeMirror.net ou Monaco Editor
Framework front	A définir => Etape1
WebSocket	Socket.io ou WebSocket natif
Transpiler métier	Développement interne ou adaptation d'un existant
Backend	A définir => Etape1
Performance monitoring	Lighthouse, Web Vitals, outils internes (A définir => Etape1)

Origine	Rédacteur	Date de réunion	Id	Version	
iD3i	Amaury Betaouaf	06/06/2025	CER-CDC-IDE -06062025	1	
Sujet	Cahier des charges - IDE interne Front				

6. ÉQUIPE PROJET

- **2-3 personnes** motivées, mentalité « bidouille » et orientés résultats.
- Rôle clé : chef de projet pour coordonner les développements et suivre le planning.

7. CRITERES TECHNIQUES DETAILLES

7.1. Transpiler métier

- Langage d'entrée : DSL métier existant (à définir avec l'équipe).
- Langage de sortie : JavaScript/TypeScript pour intégration front.
- Options recommandées :
 - Compilation à la volée pour un retour rapide dans l'IDE.
 - Gestion des erreurs claire (logs et visualisation dans l'IDE).
 - Tests unitaires et validation des conversions.

7.2. Code editor

- Support multi-langages.
- Syntax highlighting.
- Auto-complétion et snippets personnalisables.
- Plugins pour le linting et le formatage.

7.3. WebSocket

- Synchronisation des modifications en temps réel.
- Indication visuelle des curseurs des autres utilisateurs.
- Mécanisme de déconnexion/reconnexion sans perte.

7.4. UI et ergonomie

- Temps de chargement < 2s.
- Navigation fluide, sans rechargement complet.
- Système de fenêtres ou de panneaux pour organiser les écrans.