IFAPME - Module Programmation : Evaluation “Fil Rouge”

Cahier des charges – Cloud Broker

Vincent Corvers

Date de mise à jour : 12 mai 2025

---

1. Présentation générale du projet

1.1 Contexte

Le projet **Cloud Broker** est né du besoin croissant, exprimé par les ingénieurs en intelligence artificielle, de disposer d’un outil simple, rapide et fiable pour comparer les prix d’instances GPU dans le cloud. Les ressources nécessaires à l’entraînement de modèles IA sont souvent coûteuses, et les prix varient fortement d’un fournisseur à l’autre, en fonction de la zone géographique, du type d’instance (notamment les **instances spot** ou **preemptibles**) et de l’instant.

Ce projet vise à créer un **comparateur web** qui permet aux utilisateurs de trouver les offres les plus économiques parmi plusieurs fournisseurs, facilitant ainsi les décisions budgétaires liées à l’infrastructure IA.

1.2 Cadre pédagogique

Le projet est réalisé dans le cadre du **Fil Rouge IFAPME**, en première année de formation en développement web. Il mobilise les compétences acquises en Symfony (backend), Next.js (frontend), bases de données MySQL, gestion d’API REST, accessibilité web, et méthodologie projet.

1.3 Objectif

Créer une **interface web fluide et accessible** qui permet, en quelques clics, de visualiser et comparer les offres GPU cloud disponibles, tout en mettant en pratique une démarche de conception web centrée utilisateur.

---

2. Objectifs SMART

Spécifique	Mesurable	Atteignable	Réaliste	Temporel
Comparer les prix spot GPU sur le web	3 clics pour obtenir un prix	Stack vue en TP	Projet cadré et simple	Démo prévue **15 juin 2025**

---

3. Stratégie projet et utilisateur cible

3.1 Analyse SWOT

Forces (F)	Faiblesses (W)	Opportunités (O)	Menaces (T)
Expertise IA & GPU	Ressources humaines limitées	Croissance de la demande IA	Concurrence (ex. Vantage)

3.2 Persona cible : Éloïse

- Profession : Ingénieure Machine Learning
- Besoin : accéder rapidement à une offre GPU économique (ex : A100)
- Contraintes : interface sans publicité, précise, ergonomique, avec filtre + export CSV/JSON

---

4. Fonctionnalités (Méthode MoSCoW)

Priorité	Fonctionnalité
Must	Tableau TanStack filtrable et triable avec export JSON/CSV
Must	Notification back-office si nouveau type de GPU détecté via une commande PHP Symfony
Should	Authentification sécurisée (JWT) pour l'administration
Could	Historique de prix sur 7 jours
Won't	Abonnement utilisateur, alertes email

> User story : En tant qu'Éloïse, je veux filtrer les instances par GPU=A100 et région=europe-west1 afin de comparer les prix en temps réel entre différents fournisseurs et sélectionner la meilleure offre pour mon prochain entraînement IA.

---

5. Stack technique

Composant	Technologie	Version	Justification
Frontend	Next.js	14	Framework moderne étudié en formation
UI/Design	Tailwind CSS + shadcn/ui	latest	Respect des standards A11y / design sobre
Table	TanStack Table	8	Puissant pour filtrage, tri, réactivité
Backend	Symfony	6.4	API simple basée sur contrôleur Symfony
Commandes	Console PHP	n/a	Appels aux APIs externes (fournisseurs cloud)
BDD	MySQL	8	Relationnelle, vue en cours
Authent.	JWT	1	Sécurisation accès admin
Hébergement	Vercel (front) + EC2 AWS (back)	2025	Économie + flexibilité

---

6. Architecture technique (schéma simplifié)

```
[ Interface utilisateur Next.js ] ⇅ HTTPS [ Symfony Controller REST API ] ⇅ Doctr
ine ORM [ Base de données MySQL ]
```

---

7. Accessibilité et UX

- Contrastes de couleurs conformes WCAG 2.1 AA
- Navigation au clavier complète avec focus visibles
- Composants interactifs annotés (rôles ARIA)
- Design réactif (responsive) multi-écrans
- Structure HTML hiérarchique (titres, sections)

---

8. Sécurité

- HTTPS via Let's Encrypt

- Authentification admin via JWT sécurisé
- Hachage mots de passe avec Bcrypt (`password_hash`)
- CSRF token activé (Symfony)
- Protection contre injections SQL via Doctrine
- Prévention XSS (échappement Twig)
- Limitation des tentatives via RateLimiter
- Headers de sécurité : CSP, X-Frame, etc.
- Si JWT en cookie : `HttpOnly` , `Secure` , `SameSite=Strict`

---

9. Données et modèle relationnel

Entités principales :

```
Provider 1-N InstanceDetail 1-1 InstanceSpot
```

---

10. Planning prévisionnel

Semaine	Tâches réalisées / attendues
S1	Étude du besoin, rédaction MCD
S2	Modélisation BDD + entités Symfony
S3	Implémentation API Symfony + commande de récupération
S4	Frontend : affichage tableau avec tri et filtre
S5	Intégration back-office sécurisé
S6	Tests finaux, vérif A11y, déploiement final

---

11. RACI – Répartition des responsabilités

Tâche	Responsable	Autorité	Consulté	Informé
Dev Symfony API	Moi-même	Formateur	Pair	Jury
Interface utilisateur	Moi-même	Formateur	Pair	Jury
Déploiement	Moi-même	Formateur	Pair	Jury

---

12. Tests et maintenance

- Tests unitaires (PHPUnit)
- Tests de navigation (Playwright)
- Checklist accessibilité IFAPME vérifiée
- Suivi dépendances avec Dependabot (GitHub)
- CI/CD via Vercel et EC2 auto-déployé

---

13. RGPD et mentions légales

- Aucune collecte de données personnelles nominatives
- Adresse IP anonymisée (/24)

- CGU et mentions légales dans le footer

---

14. Annexes utiles

- Wireframes : page d'accueil + interface admin
- Commande Symfony PHP : récupération prix GPU spot + détection nouveaux types

---

Auteur : *Vincent Corvers*

Date de mise à jour : 12 mai 2025
Module IFAPME : Programmation – Projet Fil Rouge – Cahier des charges