CacheLab- CCF

1-Besoins du client

haute performance : Temps de réponse < 1ms pour la lecture en mémoire
Résilience : Systéme stable même sous 100k requêtes/seconds
Flexibilité : Stockage de tout type de valeur (JSON, string, ...)
Sécurité : API sécurisée , validation d'entrées , conformités RGPD si données perso
Simplicité d'utilisation : interface REST claire , documentation fournie

2-Fonctionnalités prioritaires

Ajouter une paire clé/valeur : POST /Keys
Récupérer une paire valeur via sa clé : GET /Keys/:Key
Modifier une valeur existante : PUT /Keys/:Key
Supprimer une clé / valeur : DELETE /Keys/:Key
(Optionnel) Lister toutes les clés : GET /Keys

3- Contraintes techniques

Performance : o(1) en moyenne pour les opérations CRUD grâce a une Hashmap
Langage : Choix imposé : Node.js (typescript) ou GO
API : REST JSON , compatibleavex clients HTTP
Stockage: 100% en mémoire (RAM), persistance non prioritaire dans ce MVP
Sécurité : Validation des inputs, gestion des erreurs, pas d'info sensible loggée

4- Spécialisation des choisie

Le systéme de cache sera optimisé pour stocker les données fréquemment consultées sure des sites e-commerce (example : prix des produits, état des stocks, panier d’achat)

Implications fonctionnelles :

Le cache doi accepter des valeurs en format JSON

Le temps de réponse pour une opération GET doit être inférieur a 1 ms

Une fonctionnalité TTL pourra être ajoutée pour gérer l’expiration automatique des données qui évoluent (stock,prix)

En cas de données personnelles cachées , le systéme doit gérer leur suppression selon l’article du droit a l’oubli (RGPD)