

SAE
Cahier des charges

2025/2026

I. Introduction.....	2
1. Objet du document.....	2
2. Objectif du document.....	2
3. Structure du document.....	2
4. Documents de référence.....	2
II. Enoncé.....	3
1. Contexte du projet.....	3
2. Problématique.....	3
3. Présentation de l'existant.....	3
4. Objectifs du projet.....	3
Objectif principal :	3
Objectifs détaillés :	3
5. Ce qui est dans / hors du périmètre.....	4
III. Pré-requis.....	4
1. Connaissances requises.....	4
2. Ressources matérielles.....	5
3. Ressources logicielles.....	5
4. Contraintes techniques.....	5
5. Exigences juridiques.....	6
IV. Priorités.....	6
1. Priorité 1 : Fonctionnalités essentielles.....	6
2. Priorité 2 : Fonctionnalités importantes mais non critiques.....	6
3. Priorité 3 : Fonctionnalités futures.....	6

I. Introduction

1. Objet du document

Le présent cahier des charges définit les besoins fonctionnels, techniques et organisationnels nécessaires à la conception d'une plateforme permettant l'exécution de programmes de calcul distribués ou parallèles sur un **cluster de Raspberry Pi**. Il synthétise les attentes du client, les contraintes techniques, les fonctionnalités attendues ainsi que les priorités du projet.

2. Objectif du document

Ce document a pour objectif :

- de préciser clairement le problème à résoudre ;
- d'établir les exigences et contraintes à respecter ;
- de définir les fonctionnalités à implémenter ;
- de fournir une base de référence contractuelle et technique entre les parties prenantes.

3. Structure du document

Le cahier des charges est organisé en quatre parties principales :

1. Introduction
2. Énoncé du besoin
3. Pré-requis
4. Priorités de développement

4. Documents de référence

- Recueil des besoins
- Documentation technique du cluster Raspberry Pi
- Documentation MPI et Python

II. Enoncé

1. Contexte du projet

Le client souhaite disposer d'une plateforme permettant d'exécuter des tâches de calcul distribuées sur un **cluster de Raspberry Pi**.

Les utilisateurs doivent pouvoir accéder à une interface web sécurisée, soumettre des scripts de calcul, suivre leur exécution et consulter les résultats.

2. Problématique

Fournir une solution simple, fiable et sécurisée pour :

- gérer un cluster de Raspberry Pi ;
- distribuer et exécuter automatiquement des tâches de calcul ;
- permettre aux utilisateurs de lancer des calculs et d'en consulter les résultats via une interface web.

3. Présentation de l'existant

Aucun système préexistant ne couvre l'ensemble des fonctionnalités demandées. Le cluster matériel est fourni, mais aucune interface web, ni aucun système de gestion des tâches n'existe actuellement.

4. Objectifs du projet

Objectif principal :

Créer une plateforme complète pour l'exécution de calcul distribué/parallèle sur un cluster Raspberry Pi. Les scripts à implémenter sont prime en python usant de la librairie NPI et Monte Carlo en Java utilisant l'API concurrent et Java Sockets.

Objectifs détaillés :

- **Mettre en place le cluster** : installation, configuration et communication entre les nœuds.
- **Développer une interface web sécurisée** avec authentification.
- **Gérer l'envoi et l'exécution de tâches** sur les nœuds du cluster.

- **Afficher les résultats** dans un espace utilisateur.
- **Proposer un historique des calculs** (consultation, téléchargement, suppression).
- **Assurer l'ergonomie, la rapidité et la fiabilité du système.**

5. Ce qui est dans / hors du périmètre

Dans la portée	Hors portée
Installation du cluster et configuration des RPis	Récupération d'un mot de passe perdu
Interface web + authentification	Confirmation d'inscription par mail/SMS
Lancement de calculs et affichage des résultats	Gestion avancée des rôles admin
Historique complet des calculs	Journaux d'activité admin
TBA...	

III. Pré-requis

1. Connaissances requises

- Connaissances en :
 - HTML, CSS, PHP
 - Python
 - MPI (niveau Bac+5 recommandé)
 - Java
 - Java Sockets et API Concurrent
 - Bases de données MySQL/MariaDB
 - Services réseau

- Calcul distribué et parallèle
- Compétences en ergonomie et développement d'interface web

2. Ressources matérielles

- 1 Raspberry Pi 4 - 1.8 GHz
- 4 Raspberry Pi Zero - 1.0 GHz
- Réseau local pour interconnexion
- Serveur (Apache) pour héberger le site web

3. Ressources logicielles

- OS du cluster
- Python (PyCharm, VSCode)
- MPI pour la distribution des calculs
- Java (IntelliJ, VSCode)
- API Concurrent (parallélisation en Java)
- Java Sockets (calcul distribué en Java)
- Apache
- HTML / CSS / PHP (PhpStorm/WebStorm)
- MySQL ou MariaDB

4. Contraintes techniques

- Sécurisation des mots de passe (stockage chiffré)
- Interface réactive et accessible
- Protection contre les injections (SQL)

- Logging centralisé sur le nœud maître
- Modularité du code pour faciliter la maintenance

5. Exigences juridiques

- Respect du RGPD
- Limiter la conservation des données au strict nécessaire

IV. Priorités

1. Priorité 1 : Fonctionnalités essentielles

Dovent impérativement être livrées pour que la plateforme soit utilisable :

1. Installation du cluster
2. Gestion des utilisateurs (inscription, connexion, profil)
3. Fonctionnement des algorithme distribué Prime et Monte Carlo
4. Exécution distribuée via MPI / Java Sockets
5. Affichage des résultats
6. Historique (lecture, suppression, téléchargement)
7. Interface web réactive et sécurisée

2. Priorité 2 : Fonctionnalités importantes mais non critiques

- Export des résultats (CSV/PDF)
- Vidéo explicative en page d'accueil
- Améliorations ergonomiques

3. Priorité 3 : Fonctionnalités futures

- Stopper un calcul en cours