

CAHIER DES CHARGES



UNIVERSITÉ SAINT JEAN
SAINT JEAN INGÉNIEUR

THEME: CONCEPTION ET MISE EN ŒUVRE DE BOOT INSPECTOR, UN OUTIL D'ANALYSE DU DEMARRAGE D'UNE DISTRIBUTION LINUX: CAS DE MANJARO.

Rédigé par:

- TCHINDA METEWOUE Florelle Ecladore
- OMGBA ESSENGUE Patrice

Encadrant: M. NGUIMBUS EMMANUEL

Master 1 Sécurité des Systèmes d'Information 2025/2026

PLAN

- I. CONTEXTE
- II. JUSTIFICATION
- III. OBJECTIF GENERAL
- IV. OBJECTIFS SPECIFIQUES
- V. EXPRESSION DES BESOINS
- VI. PLANIFICATION DU PROJET
- VII. ESTIMATIONS DU PROJET
- VIII. LIVRABLES

CONTEXTE

Le diagnostic du démarrage sous Manjaro est complexe à cause de logs bruts et dispersés. Boot Inspector centralise et visualise les données de boot, identifie les services lents, détecte les erreurs et facilite l'optimisation des performances du système.

JUSTIFICATION

L'outil **Boot Inspector** réponds aux principaux défis liés à l'analyse, l'optimisation et la gestion des performances du démarrage des systèmes Linux modernes sous *systemd*. Il centralise dans une seule interface toutes les données habituellement dispersées : journaux *systemd*, temps de démarrage, dépendances, erreurs et propose une analyse intelligente capable d'interpréter automatiquement ce que les administrateurs doivent aujourd'hui examiner manuellement, tels que les services lents, les dépendances critiques, les erreurs du noyau ou les problèmes liés à GRUB. Grâce à une interface visuelle intégrant graphes, timelines et tableaux interactifs, il offre une compréhension immédiate et intuitive du fonctionnement interne du système. En améliorant la fiabilité du diagnostic et en facilitant la maintenance préventive, notamment en anticipant la dégradation du temps de démarrage ou l'apparition de services défaillants, Boot Inspector transforme des données brutes en informations exploitables, renforçant ainsi la performance, la stabilité et le support des systèmes Linux.

OBJECTIF GÉNÉRAL

Concevoir un outil capable de **surveiller, analyser, diagnostiquer** et **visualiser** l'ensemble des étapes du processus de démarrage d'un système **Manjaro Linux**, afin d'optimiser ses performances et d'identifier efficacement les anomalies.

.

OBJECTIFS SPÉCIFIQUES

Compréhension et visibilité du démarrage:

- Décomposer visuellement chaque phase : Firmware → Bootloader → Kernel → Initramfs → Userspace.
- Fournir une timeline interactive des événements du boot.
- Faciliter la compréhension du rôle et du délai de chaque service systemd.

OBJECTIFS SPÉCIFIQUES

Diagnostic rapide et automatisé:

- Identifier automatiquement les **services lents** (blame).
- Déterminer la **chaîne critique systemd** responsable du temps minimal de démarrage.
- Extraire et filtrer automatiquement les **erreurs, warnings, kernel panics**, blocages, timeouts, etc.

OBJECTIFS SPÉCIFIQUES

Optimisation des performances:

- Fournir des recommandations basées sur l'analyse (ex : service trop lent, dépendances mal configurées).
- Détecter les tendances de dégradation du temps de démarrage.
- Aider à la prise de décisions (désactivation, parallélisation, modification de l'initramfs).

OBJECTIFS SPÉCIFIQUES

Amélioration de l'expérience utilisateur:

- Offrir une interface claire, moderne, ergonomique.
- Rendre les analyses accessibles à tous les niveaux d'expertise.
- Permettre un debug rapide grâce aux liens directs vers les logs filtrés.

EXPRESSION DES BESOINS

Besoins fonctionnels:

- Visualisation du temps de boot;
- Affichage des services et de leur durée;
- Filtrage des services lents;
- Recommandations pour optimisation;
- Exportation de rapports (HTML/JSON).

EXPRESSION DES BESOINS

Besoins non fonctionnels:

- Interface intuitive et ergonomique;
- Sécurité des traitements ;
- Rapidité d'analyse (en 3-5 secondes);
- Stabilité et résilience face aux mises à jour;
- Architecture flexible, portable et facile à installer.

PLANIFICATION DU PROJET

Le projet est planifié selon le tableau ci-dessous:

PHASE	DESCRIPTION
Analyse & spécifications	Compréhension du boot Linux , Étude des outils existants, Élaboration des besoins, Définition de l'architecture préliminaire
Conception technique	Architecture backend / frontend, Conception des maquettes IUG, Diagrammes UML (séquence, classes, composants)
Développement Backend	Collecte automatisée des données systemd, Parsing JSON/texte, Détection des anomalies API interne
Développement Frontend	Interface graphique complète, Timeline du démarrage, Graphiques et tableaux filtrables, Intégration totale avec backend
Tests & validation	Tests unitaires et d'intégration, Tests sur différentes configurations Manjaro, Corrections et optimisation
Livraison & documentation	Paquet installable, Guide utilisateur + documentation technique, Soutenance / démonstration

ESTIMATIONS DU PROJET

- Estimation en temps (durée du projet)

PHASE	DURÉE ESTIMÉE
Analyse & spécifications	1 semaine
Conception technique	2 semaines
Développement Backend	3 semaines
Développement Frontend	3 semaines
Tests & validation	1 semaine
Livraison & documentation	1 semaine

ESTIMATIONS DU PROJET

FR

- Estimation indicative des coûts (variables selon structure) : Le projet repose essentiellement sur des outils open source gratuits. Le coût est donc très faible, hors main d'œuvre.

PHASE	DURÉE	RESSOURCES MOBILISÉES	DÉTAILS DES ACTIVITÉS	COÛT ESTIMÉ
Analyse & spécifications	1 semaine	2 personnes	Étude approfondie du boot Linux, analyse des besoins professionnels, benchmark d'outils existants, définition de l'architecture cible	400 000
Conception technique	2 semaines	2 personnes	Architecture backend/frontend, conception des maquettes UI orientées utilisateurs techniques, modélisation UML complète	800 000
Développement Backend	3 semaines	1 personne (Back-end)	Collecte automatisée des données systemd, parsing avancé, détection d'anomalies, API interne stable	600 000
Développement Frontend	3 semaines	1 personne (Front-end)	Interface professionnelle, timeline du démarrage, graphiques exploitables, tableaux filtrables, intégration complète	600 000
Tests & validation	1 semaine	2 personnes	Tests unitaires et d'intégration, tests sur environnements réels (Manjaro), corrections et durcissement	400 000
Livraison & documentation	1 semaine	2 personnes	Paquet installable, guide utilisateur professionnel, documentation technique, démonstration	400 000
TOTAL GÉNÉRAL	11 semaines	—	—	3 200 000

LIVRABLES

- Application Boot Inspector fonctionnelle
- Code source et version packagée
- Documentation technique et utilisateur;
- Rapport de projet complet
- Rapports (Tests, journaux)
- Scripts internes