

RAPPORT - PROJET MBOX

Alexi Miaille Septembre 2025 - Janvier 2026

SOMMAIRE

1. Contexte et Présentation du Projet
2. Architecture Technique et Infrastructure
3. Implémentation des Services Réseau
4. Interface et Expérience Utilisateur
5. Bilan et Conclusion

1. CONTEXTE ET PRÉSENTATION DU PROJET

1.1 Contexte et Justification

Les box Internet fournies par les opérateurs ont considérablement évolué ces dernières années. Au-delà du simple accès à Internet, elles assurent désormais des fonctions essentielles : routage, NAT, DHCP, DNS, sécurisation du réseau local et gestion du Wi-Fi.

Cependant, les interfaces d'administration de ces équipements présentent souvent deux défauts opposés : soit une simplification excessive masquant les réglages avancés, soit une complexité décourageante pour les utilisateurs non experts. Le projet MBox tente de répondre à ce déséquilibre en proposant une interface accessible aux débutants tout en offrant des options avancées aux utilisateurs expérimentés.

1.2 Portée du Projet

Réalisé de septembre 2025 à janvier 2026 dans un cadre universitaire, ce projet reproduit les fonctionnalités principales d'un routeur domestique via une infrastructure entièrement virtualisée.

Le travail se décompose en deux axes :

- **Backend** : Administration d'un système Ubuntu avec sept services réseau configurés (Apache/HTTPS, BIND9, isc-dhcp-server, Postfix, vsftpd, MariaDB, SSH). L'architecture repose sur trois machines virtuelles simulant un opérateur, une box et un client.
- **Frontend** : Développement d'une interface web PHP permettant d'administrer le système sans recourir à la ligne de commande.

Fonctionnalités implémentées :

- Configuration réseau : IP statique, DHCP (modes débutant/expert), DNS avec délégation de zone, HTTPS
- Messagerie : Serveur Postfix avec Webmail et authentification
- Diagnostic : Speedtest avec historique graphique
- Support : Forum communautaire

1.3 Chronologie du Développement

Période	Phase	Réalisations
24 sept. - 6 oct.	Infrastructure	Création des VMs, configuration réseau, Apache/PHP, pages IP et DHCP
8 - 23 oct.	Services réseau	DNS (BIND9), premiers tests speedtest, navigation, refactorisation
4 - 17 nov.	Évolution architecture	Ajout VM Opérateur, refonte adressage WAN/LAN, double NAT
19 - 26 nov.	Base de données	Installation MariaDB, HTTPS, maquette Figma
3 - 22 déc.	Services applicatifs	Forum, Postfix, Webmail, synchronisation DNS/Mail
25 déc - 05 janv.	Finalisation	Design system, historique speedtest, uniformisation

1.4 Principes de Conception

Trois principes ont guidé le développement :

1. **Lisibilité** : Navigation fixe avec code couleur par fonctionnalité (bleu pour le réseau, violet pour les performances, vert pour la configuration, jaune pour le support).
2. **Centralisation** : Regroupement des outils d'administration dans une interface unique.
3. **Progressivité** : Page d'accueil épurée, paramètres avancés accessibles via le mode expert.

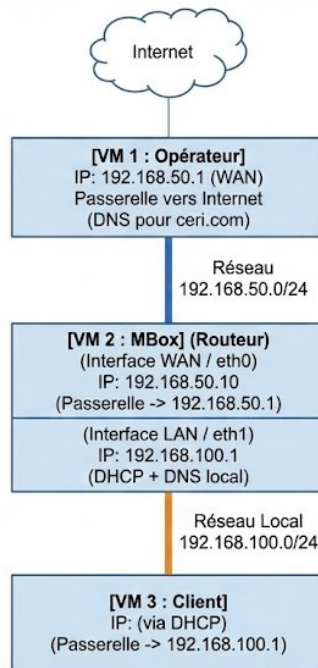
2. ARCHITECTURE TECHNIQUE ET INFRASTRUCTURE

2.1 Infrastructure Virtualisée

L'architecture initiale (septembre-octobre) ne comprenait que deux machines virtuelles. L'ajout de l'Opérateur en novembre a permis de simuler plus fidèlement la séparation WAN/LAN, au prix d'une refonte complète du plan d'adressage.

VM	Rôle	Configuration Réseau	Services
Opérateur	FAI simulé	192.168.50.1 (WAN)	BIND9 (ceri.com), NAT
MBox	Routeur	192.168.50.10 (WAN) / 192.168.100.1 (LAN)	Apache, BIND9, DHCP, Postfix, FTP, MariaDB
Client	Poste utilisateur	Attribuée par DHCP	Validation de l'interface

2.2 Topologie Réseau



Le schéma illustre l'architecture à trois niveaux :

- L'Opérateur (192.168.50.1) simule le FAI et héberge la zone DNS parente.
- La MBox assure le routage entre le WAN (192.168.50.0/24) et le LAN (192.168.100.0/24).
- Le Client obtient son adresse via DHCP et utilise la MBox comme passerelle.

Le mécanisme de routage repose sur un double NAT : la MBox masque le trafic du LAN vers le WAN, puis l'Opérateur masque le trafic WAN vers Internet. L'IP forwarding est activé via sysctl et les règles iptables MASQUERADE assurent la translation d'adresses.

2.3 Stack Technologique

L'infrastructure repose sur un système Ubuntu, choisi pour sa stabilité en environnement serveur et sa gestion des paquets via APT. Le serveur web Apache2 héberge l'interface d'administration avec chiffrement HTTPS.

L'application combine PHP pour le rendu dynamique et des scripts Bash exécutés via `shell_exec()` pour le contrôle système. La persistance des données (historique speedtest, forum, comptes mail) est assurée par MariaDB.

Côté services réseau, BIND9 gère la résolution DNS et la délégation de zone, isc-dhcp-server assure l'attribution dynamique d'adresses IP, Postfix avec stockage Maildir fait office de serveur de messagerie, et vsftpd supporte les tests de débit.

3. IMPLÉMENTATION DES SERVICES RÉSEAU

3.1 Gestion de l'Adresse IP et du Serveur DHCP

La page `change_ip.php` permet de modifier l'adresse IP de la MBox. Le script associé met à jour séquentiellement le fichier `/etc/network/interfaces`, les zones DNS, puis redémarre les services concernés.

Le serveur DHCP propose deux modes :

- **Mode Débutant** : L'utilisateur indique le nombre de machines souhaitées, le système calcule automatiquement la plage d'adresses.
- **Mode Expert** : Définition manuelle des adresses de début et fin avec validation des paramètres.

3.2 Configuration du Serveur DNS

L'architecture DNS repose sur une délégation hiérarchique :

- Zone parente (`ceri.com`) hébergée sur l'Opérateur, déléguant `alexi.ceri.com` vers la MBox.
- Zone déléguée sur la MBox avec enregistrements pour les services locaux.
- Forwarder configuré pour relayer les requêtes externes vers l'Opérateur.

Le script `config_dns.sh` synchronise automatiquement la configuration avec Postfix, ajoutant le domaine dans `mydestination` pour éviter les boucles de messagerie.

3.3 Serveur de Messagerie et Webmail

Postfix a été configuré avec stockage Maildir dans `/home/[utilisateur]/Maildir/`. Les comptes utilisateurs sont gérés via une table MariaDB avec mots de passe hachés (`password_hash`). Trois comptes de test ont été créés : `alexi`, `rachid` et `stud`.

L'interface Webmail permet la consultation de la boîte de réception et l'envoi de courriels. L'accès aux répertoires Maildir a nécessité l'ajout de `www-data` aux groupes utilisateurs concernés.

3.4 Speedtest

Le test de débit repose sur un serveur FTP dédié mesurant les performances côté serveur :

- Ping : Connexion TCP chronométrée sur le port 21.
- Download : Téléchargement d'un fichier de 100 Mo via curl.
- Upload : Envoi d'un fichier de 10 Mo généré dynamiquement.

Les résultats sont stockés dans MariaDB et affichés sous forme de graphiques comparatifs via Chart.js.

3.5 Forum de Support

Le forum repose sur deux tables MariaDB liées par clé étrangère (ON DELETE CASCADE) :

- `forum_topics` : sujets de discussion.
- `forum_posts` : messages associés.

L'interface permet de lister les sujets avec leur nombre de réponses, consulter et répondre aux discussions, et créer de nouveaux topics.

4. INTERFACE ET EXPÉRIENCE UTILISATEUR

4.1 Étude Comparative et Choix de Design

Une analyse des interfaces concurrentes a été menée en novembre pour orienter les choix ergonomiques :

Opérateur	Points forts	Points faibles
Orange (Livebox)	Structure claire par onglets	Interface datée, options avancées cachées
Free (Freebox OS)	Concept bureau innovant, très complet	Courbe d'apprentissage importante
Bouygues (Bbox)	Vision globale, équilibre débutant/expert	Configuration parfois dispersée

L'interface MBox s'inspire principalement de la Bbox avec un système de dashboard à cartes modulaires. Un toggle JavaScript permet de basculer entre mode débutant et mode expert, le choix étant persisté dans le localStorage du navigateur.

4.2 Design System

L'interface adopte une charte graphique sobre et fonctionnelle. La barre de navigation horizontale utilise un bleu nuit créant un contraste marqué avec le contenu. Le fond gris pâle accueille des cartes blanches rehaussées d'une ombre portée, détachant visuellement chaque module.

Un code couleur permet d'identifier rapidement la nature des fonctionnalités : bleu pour le réseau (IP, DNS), violet pour les performances (speedtest, logs), vert pour la configuration (DHCP), et jaune pour le support (forum).

Le développement s'appuie sur des technologies légères : CSS natif avec Grid et Flexbox, variables CSS pour la maintenance, Chart.js pour les graphiques, FontAwesome pour les icônes.

4.3 Architecture du Code

La structure du code respecte le principe **DRY (Don't Repeat Yourself)** :

- header.php et navbar.php : éléments d'interface réutilisables.
- db_connect.php : connexion MariaDB centralisée.
- server_info.php : fonction get_server_info() regroupant les requêtes système.

Mesures de sécurité implémentées :

- Protection XSS via htmlspecialchars() sur les sorties.
- Requêtes SQL préparées avec PDO.
- Échappement des paramètres shell via escapeshellarg().

- Blocage des fichiers sensibles via .htaccess.

5. BILAN ET CONCLUSION

5.1 Compétences Acquises

Ce projet a permis de développer des compétences transversales couvrant plusieurs domaines.

En administration Linux, j'ai appris à configurer des services réseau critiques (BIND9, DHCP, Postfix), à gérer rigoureusement les permissions système et à mettre en œuvre le routage NAT avec iptables.

En développement web, j'ai conçu une architecture hybride PHP/Bash, manipulé MariaDB via PDO avec requêtes préparées, et développé une interface en CSS natif utilisant Grid et Flexbox.

Sur le plan réseau, j'ai approfondi ma compréhension des mécanismes de délégation DNS, du fonctionnement du protocole DHCP, du serveur SMTP Postfix et du transfert de fichiers via FTP.

Enfin, sur le plan méthodologique, ce projet m'a confronté à des problématiques concrètes de refactorisation de code, de débogage système et de résolution de problèmes en environnement réel.

5.2 Difficultés Rencontrées et Solutions

Le développement a été ponctué de plusieurs obstacles techniques qui ont nécessité des remises en question.

En octobre, l'implémentation du speedtest en JavaScript produisait des résultats aberrants dépassant 15 000 Mbps. Après investigation, j'ai identifié que le navigateur lisait les fichiers depuis le cache mémoire au lieu de les télécharger réellement. Cette découverte m'a conduit à abandonner l'approche client-side pour implémenter un serveur FTP dédié, garantissant des mesures fiables côté serveur.

L'ajout de la machine virtuelle Opérateur en novembre a imposé une refonte majeure. L'architecture initiale à deux VMs ne permettait pas de simuler correctement la séparation WAN/LAN. J'ai dû redéfinir entièrement le plan d'adressage et configurer un double NAT, ce qui a retardé le projet mais l'a rendu plus réaliste.

La configuration de la messagerie a également posé problème en décembre. Les courriels envoyés au domaine local généraient l'erreur *mail loops back to myself*. L'analyse des logs Postfix a révélé que le domaine DNS n'était pas déclaré dans mydestination. J'ai alors implémenté une synchronisation automatique entre le script DNS et la configuration Postfix.

Enfin, l'interface Webmail refusait d'afficher les courriels : `www-data` n'avait pas les permissions de lecture sur les répertoires Maildir. La solution a consisté à ajouter l'utilisateur aux groupes concernés et à ajuster les permissions avec `chmod g+rX`.

5.3 Limites et Analyse Critique

L'architecture actuelle présente des vulnérabilités incompatibles avec un déploiement en production : l'exécution de commandes via `shell_exec()` avec droits `sudo` étendus, l'absence de protection CSRF, et un certificat HTTPS auto-signé générant des avertissements navigateur.

Plusieurs fonctionnalités sont simulées : la carte « Appareils connectés » est codée en dur, le speedtest utilise l'adresse de bouclage (127.0.0.1), la section « Téléphonie » est illustrative, et la page `logs.php` reste un placeholder.

Sur le plan technique, l'absence de requêtes AJAX impose des rechargements complets, et le forum ne dispose pas d'authentification. L'environnement virtualisé ne reproduit pas non plus les contraintes matérielles réelles.

Comparée aux box commerciales, la MBox ne propose pas de gestion Wi-Fi, pare-feu configurable, contrôle parental, QoS ni stockage réseau — fonctionnalités hors périmètre mais axes d'amélioration évidents.

5.4 Perspectives d'Évolution

- **Réseau** : Pare-feu configurable via interface web, QoS, VPN, réseau invité isolé.
- **Interface** : Liste dynamique des appareils connectés, actualisation AJAX, mode sombre.
- **Sécurité** : Tokens CSRF, limitation des tentatives de connexion, audit des logs.
- **Déploiement** : Migration sur Raspberry Pi pour confronter le projet aux contraintes matérielles.

5.5 Conclusion

Le projet MBox démontre la faisabilité d'une interface d'administration masquant la complexité des services Linux tout en conservant un accès aux paramètres avancés. L'automatisation de la chaîne de services, où un changement d'adresse IP met à jour automatiquement les zones DNS et recharge les services concernés, valide l'approche « zéro ligne de commande » pour l'utilisateur final.

Malgré les limites identifiées, notamment en matière de sécurité et de fonctionnalités simulées, ce projet a permis d'acquérir une compréhension approfondie des mécanismes réseau et de l'administration système Linux. Le mode débutant/expert offre un compromis pertinent entre accessibilité et contrôle, positionnant la MBox à mi-chemin entre les solutions techniques type OpenWRT et les interfaces grand public. Ce projet constitue une base solide pour de futures évolutions.