

Systeme d'échange sécurisé : discussion sur les options possibles

<https://claude.ai/chat/64d8e228-2365-4c99-8e91-d108a2091d3d>

Q: Dans le cadre d'échanges politique et confidentiel, je souhaite mettre en place un service de chat amélioré, type mattermost par exemple. Le principe est d'installer une application sur un serveur linux dédié, accessible par le net. Je voudrais avoir un avis sur le niveau de confidentialité réel, les conseils de sécurisation.

Pour un service de chat confidentiel à usage politique, voici les aspects clés à considérer :

Niveau de confidentialité réel

Mattermost offre un bon niveau de base mais nécessite une configuration rigoureuse :

- Chiffrement en transit (HTTPS/TLS) et au repos possible
- Contrôle total des données (hébergement propre)
- Code open-source auditable

Alternatives à considérer :

- **Element/Matrix** : protocole décentralisé, chiffrement de bout en bout natif
- **Rocket.Chat** : similaire à Mattermost, bonnes options de sécurité
- **Signal** en version serveur (plus complexe à déployer)

Conseils de sécurisation essentiels

Infrastructure :

- Serveur dédié isolé, pas de services tiers
- VPS chez un hébergeur respectant la confidentialité (Hetzner, OVH, ou mieux : hébergeur local)
- Système à jour, hardening Linux standard
- Firewall strictement configuré
- Monitoring des accès et tentatives d'intrusion

Application :

- Chiffrement de bout en bout activé
- Authentification forte (2FA obligatoire)
- Rétention des messages limitée dans le temps
- Logs d'audit configurés mais anonymisés
- Pas de plugins tiers non audités

Opérationnel :

- Accès administrateur minimal et tracé
- Sauvegardes chiffrées, stockage séparé
- Plan de continuité en cas de compromission
- Formation des utilisateurs aux bonnes pratiques

Pour un usage réellement sensible, considérez également l'aspect juridique : localisation du serveur, législation applicable, possibles réquisitions.

Quel est votre niveau technique actuel pour ce type de déploiement ? Q: C'est quasiment tout ce à quoi je pensais. Précision : dans la proposition "Chiffrement en transit (HTTPS/TLS) et au repos possible", que veut dire "au repos possible" ?

"Au repos possible" signifie que le chiffrement des données stockées sur le disque dur du serveur n'est pas activé par défaut dans Mattermost - c'est une option à configurer. Chiffrement au repos = chiffrement des données quand elles sont stockées physiquement sur le serveur (base de données, fichiers uploadés, logs, etc.) Dans Mattermost spécifiquement :

Par défaut : les messages sont stockés en clair dans la base de données PostgreSQL/MySQL Il faut configurer manuellement le chiffrement au niveau :

- De la base de données (encryption-at-rest PostgreSQL/MySQL)
- Du système de fichiers (LUKS, dm-crypt)
- Ou des deux

Pourquoi c'est crucial pour votre usage : Sans chiffrement au repos, si quelqu'un accède physiquement au serveur ou aux sauvegardes, il peut lire tous les messages en clair, même avec des mots de passe forts.

Solutions recommandées :

- Chiffrement complet du disque (LUKS) lors de l'installation Linux
- Chiffrement au niveau base de données
- Éventuellement chiffrement applicatif des messages sensibles

C'est pourquoi j'évoquais Element/Matrix qui propose du chiffrement de bout en bout natif - dans ce cas, même l'administrateur serveur ne peut pas lire les messages, contrairement à Mattermost

standard.

Cette précision change-t-elle votre orientation ?

Q/R : Mon niveau technique administrateur système linux basé sur 15 ans d'expérience, sans formation de base préalable. Donc, humilité oblige. Mais je sais déployer mattermost et autre application, sous docker en général, et assurer la partie sécurisation avec nginx/certbot par exemple. Je code en python depuis longtemps et j'ai une expertise moyenne avec postgres plus qu'avec mysql ou mariadb ; je dirais celle d'un programmeur utilisant les SGBDR

Parfait, avec votre profil vous avez largement les compétences pour un déploiement sécurisé !

Stack recommandée pour votre cas

Déploiement :

- Docker Compose avec Mattermost + PostgreSQL + nginx
- Volumes chiffrés (LUKS au niveau host)
- PostgreSQL avec TDE (Transparent Data Encryption) si version récente, sinon chiffrement au niveau filesystem

Sécurisation PostgreSQL spécifique :

```
# Dans postgresql.conf
ssl = on
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
# Connexions chiffrées uniquement
hostssl all all 0.0.0.0/0 md5
```

Points d'attention avec votre expérience :

- Mattermost stocke aussi des fichiers uploadés → volume séparé chiffré
- Logs applicatifs peuvent contenir des données sensibles → rotation + chiffrement
- PostgreSQL WAL et dumps de sauvegarde → chiffrement obligatoire

Monitoring discret :

- Fail2ban sur les tentatives de connexion
- Monitoring PostgreSQL des requêtes anormales
- Alertes sur les connexions admin

Question pratique : Envisagez-vous une authentification LDAP/AD ou comptes locaux ? Pour du politique, l'intégration avec des annuaires existants peut créer des surfaces d'attaque supplémentaires.

Avez-vous déjà une préférence d'hébergeur en tête, ou des contraintes géographiques particulières ?

Q: *pas d'authentification par annuaire. seul l'admin peut ajouter un/des comptes et gérer les accès : c'est mon idée si elle est conforme à une politique de sécurité restrictive.*

Hébergement possible : un des héberger cités ou alors sur un serveur dédié sur un LAN privé (à domicile) ; sur ce sujet, quel serait l'optimum ?

Authentification : excellente approche

Comptes locaux avec admin unique = optimal pour la sécurité. Réduire la surface d'attaque au maximum.

Configuration Mattermost recommandée :

```
# Dans config.json
"EnableUserCreation": false,
"EnableTeamCreation": false,
"EnableSignInWithEmail": true,
"EnableSignUpWithEmail": false
```

L'admin crée tous les comptes manuellement, pas d'auto-inscription.

Hébergement : LAN privé vs VPS

LAN privé (domicile) - AVANTAGES :

- Contrôle physique total
- Pas de tiers impliqué
- Latence minimale pour utilisateurs locaux
- Coût récurrent faible

LAN privé - INCONVÉNIENTS :

- Votre connexion Internet = point de défaillance unique
- IP résidentielle souvent bloquée par filtres corporate
- Pas de redondance électrique/réseau
- Vous êtes responsable de la maintenance 24/7
- Juridiquement : les données sont chez vous

VPS dédié - AVANTAGES :

- Infrastructure professionnelle (redondance, bande passante)
- IP propre, réputée
- Maintenance infrastructure déléguée
- Possibilité de changer rapidement d'hébergeur

RECOMMANDATION pour usage politique : VPS dédié chez **Hetzner** (Allemagne) ou hébergeur français type **Scaleway** - législation européenne, pas de coopération automatique avec services US.

L'avantage juridique du VPS : en cas de problème, vous pouvez migrer rapidement ailleurs.

Q : **Quelle est la criticité/sensibilité des échanges prévus ?**

Q/R: *échanges de stratégie politique en vue de campagne électorale dans un contexte tendu.*

Dans ce contexte : VPS dédié fortement recommandé

Raisons spécifiques à votre usage :

- **Disponibilité critique** : campagne = timing serré, pas de marge pour pannes domicile
- **Mobilité des utilisateurs** : équipes sur le terrain, déplacements fréquents
- **Pression potentielle** : domicile peut devenir cible (légale ou autre)
- **Déni plausible** : hébergement tiers = séparation claire données/personne

Hébergeur recommandé : **Hetzner (Allemagne)**

- RGPD strict, pas de Patriot Act
- Historique respectueux de la vie privée
- Infrastructure robuste, prix correct
- Serveur dédié (pas VPS partagé) pour isolation maximum

Sécurisation renforcée pour ce contexte

Accès :

- VPN obligatoire pour tous les utilisateurs (WireGuard)
- IP serveur jamais exposée directement
- Changement de ports standards
- Géoblocking des pays non pertinents

Juridique/Opérationnel :

- Nom de domaine via registrar respectueux (Gandi, Namecheap)

- Pas de logs applicatifs détaillés
- Rétention messages courte (7-15 jours max)
- Plan de migration rapide vers autre hébergeur
- Dead man's switch pour destruction automatique

Question cruciale : *Avez-vous envisagé Element/Matrix plutôt que Mattermost ? Le chiffrement bout-en-bout natif serait un plus énorme pour ce niveau de sensibilité.*

Q/R: *suite à notre conversation, je vais explorer la solution Element/Matrix que je ne connais que de nom.*

Une précision supplémentaire est la sécurisation des application mobile. Je trouve la sécurité Mattermost Mobile assez faible. Qu'en est-il pour Element/Matrix ?

Element Mobile : bien plus sécurisé

Chiffrement :

- **Bout-en-bout par défaut** (contrairement à Mattermost)
- Même l'admin serveur ne peut pas lire les messages
- Verification croisée des appareils obligatoire
- Perfect Forward Secrecy (clés changent régulièrement)

Sécurité mobile spécifique :

- Verrouillage par PIN/biométrie natif
- Pas de stockage en clair sur l'appareil
- Screenshots désactivables par politique
- Session timeout configurable
- Destruction à distance possible

Comparaison Mattermost vs Element Mobile

Mattermost Mobile - faiblesses :

- Messages stockés déchiffrés localement
- Push notifications contiennent du contenu
- Sécurisation dépend largement du MDM
- Pas de chiffrement bout-en-bout

Element Mobile - avantages :

- Stockage local chiffré même si appareil compromis
- Notifications chiffrées (pas de contenu lisible)
- Vérification cryptographique des identités

- Mode "low bandwidth" pour connexions surveillées

Recommandations pour votre contexte

Configuration Element serveur :

- Serveur Matrix auto-hébergé (Synapse ou Dendrite)
- Pas de fédération (serveur isolé)
- Registration fermée (admin seul)
- Retention politiques strictes

Politique mobile :

- Installation uniquement via F-Droid ou APK direct
- Pas de Google Play (tracking réduit)
- Téléphones dédiés si budget le permet

Q : *Element vous semble-t-il maintenant plus adapté ? L'installation est un peu plus technique que Mattermost mais largement dans vos cordes. Q/R: Je vois tout l'intérêt de cette solution. Je vais étudier ces 2 applications. Bien sûr, je compterais sur ton aide pour l'installation le cas échéant.*

Parfait ! C'est une excellente approche d'étudier les deux solutions avant de trancher.

Pour votre évaluation, je suggère :

Tests locaux :

- Docker Compose pour tester rapidement les deux
- Element : essayez Synapse (plus mature) vs Dendrite (plus léger)
- Interface utilisateur : testez avec vos futurs utilisateurs non-techniques

Points de comparaison :

- Facilité d'adoption par l'équipe (crucial en campagne)
- Performance avec votre charge prévue
- Complexité de maintenance au quotidien
- Fonctionnalités spécifiques (partage fichiers, calls, etc.)

Ressources utiles :

- Documentation Element : très complète
- Matrix.org : spécifications du protocole
- Communauté active sur leurs channels respectifs

N'hésitez absolument pas à revenir quand vous aurez fait votre choix ! Je serai ravi de vous accompagner sur :

- Architecture détaillée
- Docker Compose sécurisé
- Configuration des politiques de sécurité
- Scripts de déploiement/maintenance
- Procédures de sauvegarde/restauration

Bonne exploration des deux solutions ! La décision finale dépendra beaucoup de l'adoption par vos utilisateurs - c'est souvent le facteur déterminant en pratique.

Q: Une question supplémentaire au fait : utiliser une adresse IP directement plutôt qu'une redirection DNS est-ce une solution plus ou moins sécurisée dans ce contexte ?

IP directe vs DNS : plus sécurisé mais avec des trade-offs

IP directe - AVANTAGES sécuritaires :

- **Pas de trace DNS** dans les logs des résolveurs publics
- **Anonymat renforcé** : pas d'enregistrement domaine traçable
- **Pas de certificat SSL** associé à un nom (moins d'OSINT possible)
- **Résistance à la censure** : pas de blocage DNS possible

IP directe - INCONVÉNIENTS :

- **Certificats SSL complexes** : soit auto-signés (alertes navigateur), soit certificats IP (rares)
- **Flexibilité limitée** : changement serveur = nouveau partage IP à tous
- **Mémorisation difficile** pour les utilisateurs
- **Applications mobiles** : certaines n'acceptent que HTTPS valide

Recommandation hybride pour votre contexte

Solution optimale :

1. **Domaine neutre** via registrar anonyme (Njalla, 1984.is)
2. **Sous-domaine générique** : `chat.entreprise-consulting.com`
3. **Pas de WHOIS traçable**
4. **TTL DNS très court** (300s) pour migration rapide

Alternative très sécurisée :

- IP directe + **certificats auto-signés** pré-distribués
- Formation équipe : accepter le certificat manuellement
- Backup sur Tor (.onion) si situation se dégrade

Pour Element spécifiquement : Les clients mobiles Element gèrent mieux les certificats non-standard que Mattermost.

Dans votre contexte tendu, je pencherais pour le **domaine neutre** : bon compromis sécurité/usabilité. L'IP directe si la situation devient vraiment critique.

Q: *Votre instinct vous pousse vers quelle approche ?*

R: *Je n'ai pas de compétences sur cette question. Je m'en remet à cette analyse.*

Parfait, alors **domaine neutre** ce sera !

Plan d'action concret :

1. Registrar anonyme :

- **Njalla.is** (recommandé) : paiement crypto accepté, pas de WHOIS
- Domaine générique type : `consulting-web.com`, `tech-services.eu`

2. Configuration DNS :

```
chat.votre-domaine.com A xxx.xxx.xxx.xxx (TTL 300)
```

3. Certificat SSL :

- Let's Encrypt via certbot (gratuit, automatique)
- Renouvellement auto dans Docker

4. Migration d'urgence :

- Script préparé pour changer l'IP en 5 minutes
- Nouveau serveur déployable rapidement

Cette approche vous donne :

- Sécurité élevée (pas de trace personnelle)
- Facilité d'usage (HTTPS valide, mémorisable)
- Flexibilité (migration rapide possible)

Quand vous serez prêt pour le déploiement, on pourra détailler :

- Le choix exact du registrar
- La config Docker complète

- Les scripts de migration d'urgence

C'est parti pour une solution robuste et discrète ! ☐☐

Revision #5

Created 2025-09-27 21:45:53 UTC by Nicolas Farrie

Updated 2025-09-28 14:25:54 UTC by Nicolas Farrie