Chaton sur Élement/Matrix

https://wiki.chatons.org/doku.php/services/messagerie_instantanee/matrix

Matrix/Element

Element est une application libre de messagerie instantanée et voix sur basée sur le protocole Matrix.

Matrix fonctionne sur le principe de l'email, c'est à dire qu'il vous faut un compte chez un fournisseur de compte Matrix. Vous pouvez ensuite utiliser Element pour vous y connecter par l'outil de votre choix (smartphone, tablette ou ordinateur).

La messagerie instantanée de confiance de l'Etat français , Tchap, repose sur Matrix/Element.

Fonctionnalités

- messagerie instantanée entre deux personnes ou pour un groupe
- possibilité d’échanger photos, vidéos ou n’importe quel type de fichiers dans le fil de conversation
- chiffrement des communications de bout en bout pour la messagerie via les protocoles OLM et MEGOLM;
- plusieurs clients peuvent être utilisés simultanément avec la même identité ;
- en installant son propre homeserver, on peut fonctionner complètement en autonome ou bien en rejoignant la fédération. Serveur et client libre et décentralisé (auto-hébergeable)
- Multiplateforme ;
- Connexions possibles avec d'autres messageries instantanées via des passerelles Matrix : envoi de SMS, de courriels, WhatsApp, XMPP,…
- appels audio-vidéo entre deux personnes ou bien audio et visio conférences entre plusieurs personnes;
- Bonne qualité sonore, sans temps de latence (en fonction de la qualité de la connexion internet) ;
- Partage d'écran ;

Compliqué du à la fédération :

- Base d'utilisateur. inscription/sauvegarde/restauration facile.
- Appels audio/visio pas vraiment exploitable en réseau mobile

Découvrir à quoi ça ressemble

Si vous le souhaitez, vous pouvez essayer Element sur le navigateur internet **depuis votre ordinateur** en allant sur le groupe de discussion ouvert d'Alsace Entraide Numérique . Nota : pour essayer depuis un smartphone il vous faudra l'application Element et un compte.

Important: n'hésitez pas à indiquer que c'est un test pour essayer Element pour que les bénévoles d'ARN ou du Hackstub ne soient pas trop confus :) .

Procédure pour se connecter

Etape 1 : choisir votre fournisseur pour le compte Matrix

Ci-dessous une liste d'hébergeur du collectif qui proposent d'ouvrir un compte Matrix en accès libre (ou sur simple inscription) :

https://www.chatons.org/entraide/list?title=service_type_target_id=All&field_software_target_id=274&field_is_shared_value=All

L'outil de recherche des services CHATONS peut vous permettre de trouver d'autres hébergeurs du collectif qui peuvent accueillir votre compte Matrix sous conditions.

Si vous cherchez à vous interconnecter avec des contacts Signal ou même WhatsApp comme sur le chat Alsace Entraide Numérique, vous pouvez adhérer chez Alsace Réseau Neutre pour bénéficier de leurs passerelles.

Etape 2 : installer Element sur votre smartphone

Une fois que vous avez votre compte, vous avez probablement envie de pouvoir l'utiliser en dehors d'un navigateur web.

1. Installer l'application Element sur votre smartphone depuis Google Play, le store d'iOS ou encore F-droid (l'application pèse 140Mo)
2. Ouvrez l'application
3. Cliquez sur “J'ai déjà un compte”
4. Cliquez sur le lien: “Me connecter avec mon identifiant Matrix”
5. Indiquez votre compte Matrix, sous la forme: “@utilisateur:domaine.com”
6. Indiquez votre mot de passe
7. Validez
8. A cette étape: il vous sera demandé une phrase de chiffrement à retenir. Cette phrase est nécessaire en plus du mot de passe du fait des techniques utilisées par Element pour sécuriser votre communication.

Etape 3 : discuter pour la première fois

Arrivée à cette étape, vous avez accès à l'interface pour discuter, mais il n'y a aucun contact disponible.

SI vous connaissez quelqu'un sur Matrix vous pouvez ajouter son compte en utilisant l'icône + verte.

Si vous ne connaissez personnes, vous pouvez chercher et ajouter un salon, par exemple celui des chatons ou celui d'Alsace Entraide Numérique cité plus haut.

Enfin, il est possible en utilisant des passerelles de discuter avec d'autres protocoles notamment IRC. Si vous souhaitez communiquer avec quelqu'un utilisant Signal, Whatsapp ou Facebook messenger, vous devriez vous rapprocher des chatons qui proposent des passerelles pré-installées:

- Alsace Réseau Neutre - WhatsApp / Signal - pas de compte personnel sur WA ou SIgnal nécessaire

Documentation à destination des techniciens des hébergeurs des CHATONS

Relier Matrix et d'autres réseaux de messagerie via des passerelles

Traduction de https://matrix.org/docs/guides/types-of-bridging et adaptation au contexte CHATONS.

Une idée fondamentale qui a motivé le développement de Matrix est la possibilité de communiquer entre différents réseaux de messagerie, libres comme ouverts. L'idée est que deux utilisateur.ice.s de deux réseaux différents discutent dans un salon qui est rendu accessible sur les deux réseaux grace à une passerelle. Plusieurs solutions techniques existent pour implémenter une passerelle entre deux réseaux (en Anglais “bridge”).

Types de salons "rooms"

Migrer des salons entre portail et salon raccord n'est pas pratique. En effet il n'y a pas de possibilité pour les utilisateurs de retirer les portails une fois créés. Le risque est donc de se retrouver avec un mélange d'utilisateurs de portails et raccords dans un salon.

Salons Portails "portal rooms"

La passerelle peut s'identifier elle-même sur le serveur du réseau tiers et contrôle des morceaux du domaine d'adressage des alias/identifiants de salons. Ainsi les utilisateur.ice.s Matrix rejoignent un salon du réseau tiers de manière transparente en rejoignant freenode_#wherever:matrix.org Le salon Matrix résultant est ponté automatiquement au seul salon cible sur le réseau tiers. Le controle d'accès des users Matrix est généralement géré par le réseau tiers. C'est la passerelle la plus puissante, qui permet l'intégration des deux réseaux la plus étroite et transparente. Il est possible pour un user Matrix de rejoindre un salon du réseau tiers, simplement en renseignant l'identifiant, sans manipulation complexe.

Salons Raccords "plumbed rooms"

Un salon Matrix existant peut etre raccordé à un ou plusieurs salons tiers en configurant une passerelle (qui peut etre auto-hébergé). Par exemple, #matrix:matrix.org est raccordé sur #matrix sur Freenode. Le controle d'accès des utilisateurs Matrix est nécessairement géré par le salon du coté Matrix. C'est utile pour relier différents réseaux via Matrix, meme dans le cas de réseaux privateurs pour lesquels le protocole n'est pas libre et/ou il n'y a pas d'accès admin aux serveurs pour mettre en place une passerelle de type portail.

Types de passerelles

Du plus simple au plus avancé

Passerelle reposant sur un robot

La manière la plus simple d'échanger des messages avec un réseau tiers est que la passerelle se connecte au réseau tiers en utilisant un (ou plusieurs) compte fictif appelé robot-passerelle. Par exemple on créé un compte WhatsAppBot ou WhatsAppBot[2] dont la passerelle connaît les identifiants. Ce robot relaie les messages entre les deux réseaux pour le compte des autres utilisateur.ice.s. L'expérience utilisateur est très mauvaise car toutes les méta-données concernant les messages sont perdues, en particulier leur expéditeur. C'est une méthode utilisée pour les réseaux privateurs centralisés des GAFAM tel que Telegram

https://github.com/SijmenSchoon/telematrix

Passerelle reposant sur un robot-API pour la création d'utilisateurs virtuels

Certains réseaux tiers supportent l'injection de messages provenant de comptes fictifs / utilisateurs virtuels. Ce peut être utilisé pour représenter les utilisateur.ice.s Matrix avec un compte fictif sur le réseau tiers. Par exemple, Slack permet la création de robots distants sur demande. Ces robots peuvent alors créer des comptes fictifs aux utilisateurs Matrix qui sont alors affichés proprement dans les salons Slack comme des utilisateurs virtuels. Cependant, les comptes fictifs ne sont pas implémentés de la même manière que les comptes réels du réseau tiers, et n'ont pas les mêmes droits/accès aux méta-données et notifications temps-réel du réseau (sémantique). Ainsi, ils n'ont pas de présence, de profil, ne peuvent pas être contactés en message privé (MP), et leur identifiant ne peut pas être auto-complété, par exemple pour être pingé. Ils ne voient pas non plus quand quelqu'un est entrain d'écrire. C'est la méthode utilisée pour la passerelle Slack matrix-appservice-slack https://github.com/matrix-org/matrix-appservice-slack

Passerelle marionnette simple

C'est une passerelle permettant une intégration plus étroite. La passerelle s'identifie sur le réseau tiers comme étant un logiciel client tiers. L'utilisateur Matrix doit donc posséder un compte sur le réseau tiers. La passerelle créé des “marionnettes” Matrix des utilisateurs du réseau tiers. Autrement dit chaque utilisateur du réseau tiers se voit attribuer un compte fictif Matrix avec lequel l'utilisateur Matrix discute. A chaque salon du réseau tiers est associé un salon miroir Matrix. Les utilisateurs côté réseau tiers ont l'impression de discuter avec le compte de l'utilisateur Matrix du réseau tiers et ne peuvent pas savoir que tout est mis en miroir sur Matrix. Toute la sémantique (méta-données et notifications) du réseau tiers est disponible pour la passerelle via le protocole/API du logiciel client et peut donc être mise en miroir sur Matrix. Cependant, la passerelle doit gérer le processus d'authentification du réseau tiers pour que l'utilisateur Matrix se connecte à son compte sur le réseau tiers. C'est la méthode utilisée par les passerelles IRC (si vous la configurez pour se connecter au serveur IRC avec votre compte IRC réel). https://github.com/matrix-org/matrix-appservice-irc , télégram https://github.com/matrix-org/matrix-appservice-tg et git https://github.com/matrix-org/matrix-appservice-gitter

Passerelle marionnette double

Dans le cas d'une marionnette simple, la passerelle “tire les ficelles” sur le réseau tiers pour le compte de l'utilisateur Matrix. Donc ce qui se passe sur Matrix est répercuté sur le réseau tiers. Dans l'idéal on aimerait aussi que ce que l'utilisateur fait sur le réseau tiers avec son compte tiers soit également répercuté sur Matrix. Par exemple si l'utilisateur lit un message depuis l'appli WhatsApp, on aimerait que ce message soit marqué comme lu dans la conversation Matrix miroir. Pour cela, la passerelle doit pouvoir tirer les ficelles de la partie Matrix du miroir. Et donc manipuler le compte Matrix de l'utilisateur en fonction de ce que le compte tiers de l'utilisateur fait. C'est la méthode ultime pour une passerelle personnelle, car toute la sémantique (méta-données et notifications) est précisément mise-en-miroir sur chaque réseau. A l'inverse, dans le cas d'un robot-relai, le compte côté réseau tiers et celui côté Matrix ne représentent pas un et un seul même utilisateur “ne font pas un”. La difficulté côté Matrix est qu'il faut trouver une manière élégante pour que la passerelle se connecte au compte Matrix de l'utilisateur (délégation de jetons d'accès). Les difficultés sont spécifiques aux limitations de chaque réseau tiers. Par exemple, la plupart des réseaux tiers ne permettent pas de représenter un autre utilisateur Matrix que celui fournissant son compte tiers (voir robot-relai hybride). Matrix-puppet-bridge était le premier projet pour de tels passerelles https://github.com/matrix-hacks/matrix-puppet-bridge#examples Les passerelles mautrix sont les plus performantes à ce jour pour les réseaux privateurs des GAFAM tels que Telegram, WhatsApp, Facebook (Messenger). https://docs.mau.fi/bridges/index.html Voir aussi celles de https://github.com/Sorunome/mx-puppet-bridge

Passerelle marionnette à robot-relai hybride

Il s'agit d'une combinaison entre les passerelles marionnette simple et double. Une telle passerelle essaie de permettre à plusieurs utilisateurs Matrix d'être représenté sur le réseau tiers via un seul compte tiers. Elle exploite la méthode de robot-passerelle.

Passerelle serveur-à-serveur

Certains réseaux tiers (IRC,XMPP,SIP,SMTP,NNTP,GnuSociql, etc.), tout comme Matrix, supportent la fédération, libre ou controlée. C'est-à-dire que ces réseaux sont composés de plusieurs sous-réseaux hébergés sur différents serveurs et noms de domaines. Les utilisateurs ayant un compte sur un sous-réseau peuvent communiquer avec les utilisateurs d'un autre sous-réseau à condition que les deux serveurs soient fédérés, c-à-d qu'ils acceptent de communiquer entre eux. La manière la plus élégante de ponter ces réseaux entre eux est que la passerelle se comporte comme un serveur fédéré du réseau tiers. Ainsi tous les utilisateurs et salons de Matrix sont directement accessibles depuis le réseau tiers et vice-versa. Il suffit pour un utilisateur Matrix de préfixer le pseudo d'un utilisateur ou l'identifiant d'un salon tiers par le nom de son serveur pour lancer la conversation. Voir salons-portail.

Passerelle à sens-unique

Des passerelles d'un réseau tiers vers Matrix sont utilisés pour suivre des actualités, comme par exemple https://github.com/turt2live/matrix-appservice-instagram ou github.

Passerelle sidecar

La passerelle est un client tiers pour le réseau tiers. Seul un utilisateur Matrix peut voir les conversations du réseau tiers. Les permissions sont gérées par le réseau tiers, pas par Matrix. Voir bitlbee. https://matrix.org/docs/guides/types-of-bridging/#sidecar-bridge

Présentations / tutoriels d'utilisation

Tutoriels détaillés

- Concept et première utilisation
- Première utilisation
- Passerelle WhatsApp ARN-Messager

Tutoriels succincts

Wiki de Tetaneutral

Présentations

- Doc de Tedonum
- Présentation de Matrix par Grésille
- https://linuxfr.org/news/matrix-pour-decentraliser-skype-whatsapp-signal-slack-et-discord

Configuration / architecture / guides d'installation du service

- Déployer la passerelle WhatsApp ARN-Messager. Repose sur:

- la passerelle Mautrix-WhatsApp installée via Yunohost
- le robot de mise en relation ARN-Messager.
- Contacter ARN (#arn:sans-nuage.fr) pour plus d'infos et les droits sur le tuto de déploiement.

Bug de connection CAS

Si un SSO/CAS est configuré comme avec le paquet synapse_ynh, seule la connexion par CAS est possible, la connexion par identifiant+mdp est désactivée. Certains clients comme pidgin ne supportent pas la connexion CAS, ou Element iOS est buggué. Voici comment résoudre ça :

https://github.com/YunoHost-Apps/synapse_ynh/issues/379

Configuration du Client web Element

Pour permettre aux utilisateur.ice.s de trouver facilement les salons publics des autres CHATONS ajouter à `element/config.json` :

```
    "embedded_pages": {
        "login_for_welcome": true
    },
    "roomDirectory": {
        "servers": [
            "deuxfleurs.fr",
            "exarius.org",
            "garbaye.fr",
            "hadoly.fr",
            "libreon.fr",
            "matrix.domainepublic.net"
            "matrix.fdn.fr",
            "matrix.interhop.org",
            "matrix.lavallee.ynh.fr",
            "matrix.nomagic.uk",
            "matrix.org",
            "matrix.parinux.org",
            "matrix.underworld.fr",
            "sans-nuage.fr",
            "tedomum.net"
        ]
    },
```

Notez que seuls exarius, garbaye, nomagic, sans-nuage, underworld et tedomum ont autorisé à lire la liste des salons public (dans la config synapse `allow_public_rooms_over_federation: true`). Ou alors pour les autres c'est qu'ils n'ont pas de salons publics ou que l'url renseignée ici est inexacte.

Gestion quotidienne de synapse

L'implémentation serveur Matrix synapse propose une API d'administration plutôt complète, accessible aux URL débutant par `/_synapse/`. Cette API est documentée sur la documentation officielle de synapse. Les requêtes et les réponses sont formatées en JSON pour la plupart et interprétables automatiquement grâce à l'outil jq. Pour employer aisément la commande, nous proposons la fonction bash suivante.

```
function synadm { curl -X "$1" -H "Content-Type: application/json" -H "Authorization: Bearer $SY
# Avant usage, exporter les variables suivantes (le token doit être associé à un compte administ
export SYNAPSE_HS=matrix.monchaton.org
export SYNAPSE_TOKEN=XXX
```

Il existe des outils d'administration alternatifs tels que synadm en Python, présentant les fonctionnalités de l'API Synapse avec plus d'abstraction encore.

A titre d'exemple, la fonction permet de manipuler les utilisateurs et les salons :

```
# Afficher les profils dont le nom contient un mot-clé
synadm GET 'v2/users?name=bot' | jq
# Afficher uniquement les mxid (pour export ou traitement automatique)
synadm GET 'v2/users?name=bot' | jq -r '.users[].name'
# Profil basique d'un compte donné
synadm GET v2/users/@user:monchaton.org | jq
# Informations de connexion du compte
synadm GET v1/whois/@user:monchaton.org | jq
# Toutes les IP connues du compte
synadm GET v1/whois/@user:monchaton.org |  jq '.devices[].sessions[].connections | map(.ip) | ur
# Salons rejoints par le compte
synadm GET v1/users/@user:monchaton.org/joined_rooms | jq
# Réinitialiser un mot de passe
synadm POST v1/reset_password/@user:monchaton.org -d '{"new_password": "<mdp aleatoire>", "logou
# Désactiver un compte
synadm POST v1/deactivate/@user:monchaton.org -d '{"erase": true}'
# Bloquer un compte silencieusement (shadow ban)
synadm POST v1/users/@user:monchaton.org/shadow_ban

# Lister les salons sur le serveur
synadm GET v1/rooms | jq
# Lister les salons sans limite
synadm GET 'v1/rooms?limit=9999999' | jq
# Informations générales sur un salon
synadm GET 'v1/rooms/!OhSiGLbNAQMVgDlMaF:monchaton.org' | jq
# Lister les membres d'un salon
synadm GET 'v1/rooms/!OhSiGLbNAQMVgDlMaF:monchaton.org/members' | jq
# Supprimer un salon localement (adapter le block et le purge)
synadm DELETE 'v1/rooms/!OhSiGLbNAQMVgDlMaF:monchaton.org' -d '{"block": false, "purge": true}'
```

```
# Vider l'historique de messages d'un salon (cache local)
synadm POST v1/purge_history/!OhSiGLbNAQMVgDlMaF:monchaton.org' -d "{\"purge_up_to_ts\":\"$(date
```

Nettoyage d'un serveur synapse

Préférer les solutions basées sur l'API. Modifier directement la BDD est fortement déconseillé, car cela risque de corrompre des choses au niveau de la fédération.

- [recommandé sur le chat synapse](https://jacksonchen666.com/posts/2022-12-03/14-33-00/)
- https://foss-notes.blog.nomagic.uk/2021/03/matrix-database-house-cleaning/
- https://levans.fr/shrink-synapse-database.html
- https://gitea.prk.st/romain/synapse-tools/raw/branch/master/synapse-purge.sh
- https://www.tedomum.net/dev/service/matrix/administration/

Un serveur synapse est gourmand en ressources, principalement en consommation disque et mémoire à travers sa base de données. Nettoyer régulièrement le serveur permet de conserver l'usage mémoire et la taille des tables à un niveau raisonnable (en 2021, un serveur synapse de taille moyenne doit consommer moins de 6Go de RAM et 100Go de disque, beaucoup moins s'il ne fédère pas avec de gros salons). Le guide de Levans est très complet pour le nettoyage d'un serveur Matrix, et quelques extraits adaptés sont proposés à suivre.

Supprimer les salons sans membre local :

```
# Récupérer tous les salons dans un fichier
synadm GET 'v1/rooms?limit=999999' > rooms.json
# Supprimer les salons vides (attention, l'exécution peut prendre plusieurs jours et consommer c
jq -r '.rooms | map(select(.joined_local_members == 0)) | .[].room_id' < rooms.json | xargs -ti
```

Nettoyer l'historique des salons remplis :

```
# D'abord extraire la liste des salons les plus remplis, à lancer sur le serveur PostgreSQL
> \copy (select room_id, count(*) as cnt from events group by room_id order by cnt desc) to '/tm
# Puis purger l'historique du top 10 jusqu'à -1 mois, attention les traitements sont lancés en p
head -n 10 /tmp/rooms_to_clean | cut -d, -f1 | xargs -ti $SHELL -ic 'synadm POST \'v1/purge_hist
```

Optimiser la table d'état (nécessite l'installation de synapse-compress-state) :

```
# D'abord extraire la liste des salons les plus lourds en table d'état, à lancer sur le serveur
> \copy (select room_id, count(*) AS cnt from state_groups_state group by room_id order by cnt d
# Puis compresser l'historique du top 100 (exporter les variables de connexion au serveur postgr
# Cette commande ne modifie pas l'état en base et peut tourner pendant plusieurs jours
head -n 100 /tmp/rooms_to_compress | cut -d, -f1 | xargs -ti synapse-compress-state -p "host=$PG
# Finalement exécuter tous les fichiers SQL
```

```
ls *.sql | xargs -ti psql -U$PGUSER -p 5432 -h $PGHOST $PGUSER -f '{}'
```

Nettoyer les évènements orphelins

Problèmes : https://github.com/element-hq/synapse/issues/9406 https://github.com/element-hq/synapse/issues/15575

Solution trouvée par club1 puis fouine@underworld : Avec le temps, ma base a continué de grossir et ce malgré l'application régulière des préceptes/règles d'hygiènes ci-dessus, qui permettaient quand même de limiter les dégâts.

En cherchant… encore… et encore, je suis tombé sur ce blog : https://blog.gelesneeuw.nl/technology/2024/04/20/matrix-maintenance/

Il est fait référence à un "bug" de jeunesse de synapse, qui perdure, dont voici l'issue : https://github.com/matrix-org/synapse/issues/3364

Et dont voici le remède : https://github.com/erikjohnston/synapse-find-unreferenced-state-groups

C'est EXACTEMENT ce qui causait l'embonpoint de ma base de donnée. Après application du traitement, celle-ci est passée de 250Go à 18Go… oui, DIX-HUIT Giga.

Pour faire simple, ARRÊTEZ SYNAPSE, ARRÊTEZ SYNAPSE et ARRÊTEZ SYNAPSE, puis générez le fichier csv qui va contenir la liste des "événements orphelins":

```
rust-synapse-find-unreferenced-state-groups -p "postgresql://$DBUSER:$PGPASSWORD@localhost/$DBNA
```

Enfin, dans le cli psql, une fois connecté à votre base de donnée:

```
CREATE TEMPORARY TABLE unreffed(id BIGINT PRIMARY KEY);
COPY unreffed FROM '/home/user/unreferenced.csv' WITH (FORMAT 'csv');
DELETE FROM state_groups_state WHERE state_group IN (SELECT id FROM unreffed);
DELETE FROM state_group_edges WHERE state_group IN (SELECT id FROM unreffed);
DELETE FROM state_groups WHERE id IN (SELECT id FROM unreffed);
```

Ce bug m'aura pourri la vie durant 3 ans, m'imposant des plans de remédiations lourds, des arbitrages de ressources, avec indispo. bref, je suis content :)

Si vous souhaitez automatiser les opérations de nettoyage, vous pouvez reprendre / vous inspirer du script proposé ici et l'appelé via une tâche cron de façon quotidienne pendant les heures creuses (en anglais).

Instances publiques déployées

Il ne s'agit pas de proposer une liste exhaustive car il est difficile d'avoir connaissance de toutes les instances déployées. Si vous êtes administrateur d'une instance et qu'elle n'est pas encore recensée, n'hésitez pas à l'ajouter. L'usage de cette liste est à la discrétion de toutes les personnes y ayant accès. Nous ne garantissons pas la qualité du service fourni par les liens ci-dessous.

Chez les CHATONS :

https://www.chatons.org/entraide/list?title=service_type_target_id=All&field_software_target_id=274&field_is_shared_value=All

Ailleurs :

Création de compte libre

* matrix.org (serveur par défaut hébergé par la fondation New Vector https://vector.im)
https://matrix.org/docs/projects/try-matrix-now/

Création de compte sur demande/inscription

* matrix.fdn.fr (Pack FDN niveau I) https://www.fdn.fr/fdn-ouvre-son-instance-synapse-matrix-et-la-bridge-a-irc-geeknode/

Revision #1
Created 2025-09-27 22:57:22 UTC by Nicolas Farrie
Updated 2025-09-27 22:58:25 UTC by Nicolas Farrie