

Mandant : École polytechnique fédérale de Lausanne

Chef de projet : M. Luc Venries

Responsable : M. Alain Bron

Mandataire : M. Nathan Ernst

Du 21 août 2023 au 26 septembre 2023

[Introduction 2](#_Toc314509814)

[Le projet 3](#_Toc1488243865)

[Matrix qu’est-ce que c’est ? 3](#_Toc960841806)

[Matrix – Le fondamental 3](#_Toc1632727155)

[Les rooms 4](#_Toc39084010)

[Les bridges 5](#_Toc1166808743)

[Les spaces 6](#_Toc1101437377)

[Les APIs 8](#_Toc2063689432)

[Principe de communication 9](#_Toc1873074516)

[Matrix – Architecture 10](#_Toc1988310904)

[Les Acteurs 11](#_Toc1202176593)

# Introduction

Les collaborateurs de l’École polytechnique fédérale de Lausanne − ci-après EPFL − utilisent différents outils pour communiquer. Les produits Cisco (Cisco Jabber) sont utilisés pour la téléphonie (softphones) et fournissent également une messagerie (protocole XMPP). Bien que certaines salles de visioconférences utilisent Cisco WebEx, l’outil de vidéoconférence officiel est Zoom, qui lui aussi propose une messagerie instantanée. Dans les différents services et unités de l’EPFL, on trouve aussi Slack, Mattermost, Telegram, Signal, Discord ou WhatsApp et bien que certains d’entre-eux proposent aussi la téléphonie ou la vidéo-conférence, ils sont principalement utilisés pour les fonctionnalités de messageries instantanée (chats).

Le mandant, l’EPFL, aimerai agréger / féderer ses différents flux de communication tels que produits par les applications Cisco *Jabber* ou *WebEx*, ou encore *Zoom*, via le protocole ouvert Matrix

# Le projet

Ce projet consiste à étudier et mettre en œuvre le protocole Matrix et son environnement technique, afin d’agréger différents flux de communication temps réel du mandant.

Le protocole Matrix

# Matrix qu’est-ce que c’est ?

Matrix est un protocole de communication ouvert et décentralisé pour la communication sécurisée en temps réel sur internet. Il fournit une base pour les services de messageries et de voix sur IP et pour l’internet des objets. Il est Open Source, ce qui lui permet d’éviter les licences propriétaires et est conçu pour être interopérable. Il permet également d’intégrer d’autres applications de messagerie tels que Telegram, Discord, WhatsApp, etc.

Pour accéder à Matrix, il est possible de se connecter sur des serveurs publics ou de créer son propre serveur. On y accède ensuite via un client qui utilise ce protocole comme FluffyChat, Element, Cinny, etc.

# Matrix – Le fondamental

Le protocole Matrix est conçu pour être une infrastructure de communication décentralisée, sécurisée et interopérable. Voici les principaux concepts qui guident sa conception :

**Fédération :** Le protocole Matrix repose sur le principe de la fédération, ce qui signifie que différents serveurs Matrix peuvent interagir et communiquer les uns avec les autres. Cela permet donc aux utilisateurs de créer des comptes sur n’importe quel serveur compatible Matrix et de communiquer avec des utilisateurs sur d’autres serveur. Cela représente le fonctionnement des mails à travers différents fournisseurs.

**Décentralisation :** L'un des principaux principes de conception du protocole Matrix est la décentralisation. Contrairement aux services de messagerie centralisés, où un seul fournisseur contrôle l'ensemble du système, Matrix repose sur de nombreux serveurs indépendants (homeservers). Chaque serveur héberge des comptes d'utilisateurs et des données, garantissant que le contrôle des données est distribué.

**Identifiants Uniques :** Le protocole utilise des identifiants uniques pour chaque utilisateur et chaque pièce. Les IDs Matrix sont utilisés pour localiser et identifier les ressources sur le réseau.

**Pièces (Rooms) :** Le concept de pièces est au cœur du protocole Matrix. Les utilisateurs communiquent dans des pièces, qui peuvent être publiques ou privées. Les pièces peuvent être organisées autour de thèmes spécifiques, de groupes d'utilisateurs ou de projets.

**Messages et Événements :** Le protocole traite les messages sous forme d'événements. Chaque action (envoi de message, ajout d'utilisateur, changement d'autorisation, etc.) est représentée comme un événement dans l'historique de la pièce.

**Synchronisation Continue :** Les clients et les serveurs travaillent ensemble pour garantir que les clients sont constamment synchronisés avec les dernières mises à jour dans les pièces. Cela assure une expérience de communication fluide et en temps réel.

**Chiffrement de Bout en Bout :** La sécurité et la confidentialité sont prioritaires dans la conception du protocole. Le chiffrement de bout en bout permet de sécuriser les messages de telle sorte que seuls les destinataires légitimes puissent les lire.

**Extensions et Intégrations :** Le protocole Matrix est conçu pour être extensible. Cela signifie que de nouvelles fonctionnalités peuvent être ajoutées et que des intégrations avec d'autres systèmes sont possibles.

**Règles de Communication :** Le protocole définit les règles et les normes de communication entre les clients et les serveurs. Il spécifie comment les messages sont envoyés, reçus, stockés et synchronisés.

**Transparence :** L'architecture et le code du protocole sont ouverts, ce qui signifie que la conception et le fonctionnement sont transparents et accessibles à tous. Cela encourage la confiance et la participation de la communauté.

## Les rooms

Il est possible de créer des pièces (rooms) qui est un espace de discussion où les utilisateurs peuvent échanger des messages, des fichiers et d'autres contenus en temps réel. Les utilisateurs peuvent participer à plusieurs pièces à la fois.

**Création de Rooms :** Les utilisateurs ont la possibilité de créer leurs propre rooms. Lors de la création d'une room, l'utilisateur peut définir un nom, une description et d'autres paramètres pour définir le contexte de la conversation.

**Types de Rooms :** Les rooms peuvent être de différents types en fonction de leur accessibilité et de leurs règles de participation. Il existe principalement deux types de rooms

**Rooms Publiques :** Les rooms publiques sont accessibles à tous les utilisateurs de Matrix. Les utilisateurs peuvent rejoindre ces rooms sans invitation et participer aux discussions.

**Rooms Privées :** Les rooms privées sont restreintes et nécessitent une invitation pour y accéder. Les créateurs de la room peuvent inviter des utilisateurs spécifiques et contrôler qui peut rejoindre la conversation.

**Participation et Interaction :** Une fois dans une room, les participants peuvent échanger des messages en temps réel. Les messages peuvent contenir du texte, des images, des liens, des fichiers et d'autres contenus multimédias.

**Interopérabilité :** Une caractéristique importante des rooms Matrix est leur capacité à être interopérables entre différents serveurs Matrix. Cela signifie que les utilisateurs sur des serveurs différents peuvent participer à la même room et communiquer ensemble, tout comme des adresses e-mail peuvent communiquer entre différents fournisseurs.

**Historique des Messages :** Les rooms conservent un historique des messages échangés. Cela signifie que lorsque les utilisateurs rejoignent une room, ils peuvent consulter les messages précédents pour se mettre à jour sur la conversation.

**Gestion et Modération :** Les créateurs et les administrateurs d'une room peuvent généralement gérer la configuration de la room, inviter de nouveaux participants, supprimer des messages indésirables et gérer d'autres aspects de la conversation.

## Les bridges

Matrix prend en charge l'acheminement des messages de différentes applications de discussion vers Matrix. Ces ponts (bridge) sont des programmes qui permettent de relier Matrix à d'autres protocoles de messagerie. Cela permet aux utilisateurs de ces services externes de communiquer avec des utilisateurs Matrix sans quitter leur application préférée. Voici comment ça fonctionne pour un pont Matrix-Telegram :

**Configuration du Pont :** Sur un serveur Matrix, un administrateur configure le bridge spécifique. Ce bridge est un logiciel qui sera responsable de la communication entre les deux plateformes. Le bridge Matrix-Telegram est configuré pour se connecter aux serveurs de Telegram grâce à l’API de Telegram.

**Communication avec Telegram :** Lorsqu'un utilisateur de Telegram envoie un message dans une conversation Matrix, le bridge reçoit ce message depuis l'API Telegram et il interprète le message et le traduit dans un format compréhensible par le protocole Matrix.

**Synchronisation des Messages :** Le bridge envoie ensuite ce message traduit vers le serveur Matrix auquel il est connecté. Le serveur Matrix distribue alors ce message à la room auquel les utilisateurs Matrix et le participant de Telegram communiquent.

**Envoi vers Matrix :** Lorsqu’un utilisateur de Matrix envoie un message dans une communication Telegram, le processus est le même.

L’avantage principal des bridges est la transparence pour les utilisateurs. Cela permet de faire croire aux utilisateurs Matrix et aux utilisateurs Telegram qu’ils sont dans la même room.

## Les spaces

Les espaces (spaces) sont une extension du protocole Matrix qui permet d’organiser et de structurer les rooms de manière hiérarchique. Cela permet une meilleur organisation des conversations.

**Création de Spaces :** Les utilisateurs peuvent créer des spaces pour regrouper plusieurs rooms en fonction de leurs besoins. Un space peut être créé autour d'un thème spécifique, d'un projet en cours, d'un groupe d'utilisateurs ou de toute autre organisation logique.

**Hiérarchie et Structuration :** À l'intérieur d'un space, nous pouvons créer des catégories et organiser les rooms de manière hiérarchique. Les catégories aident à structurer les discussions en groupant des rooms similaires ou liées sous un même en-tête.

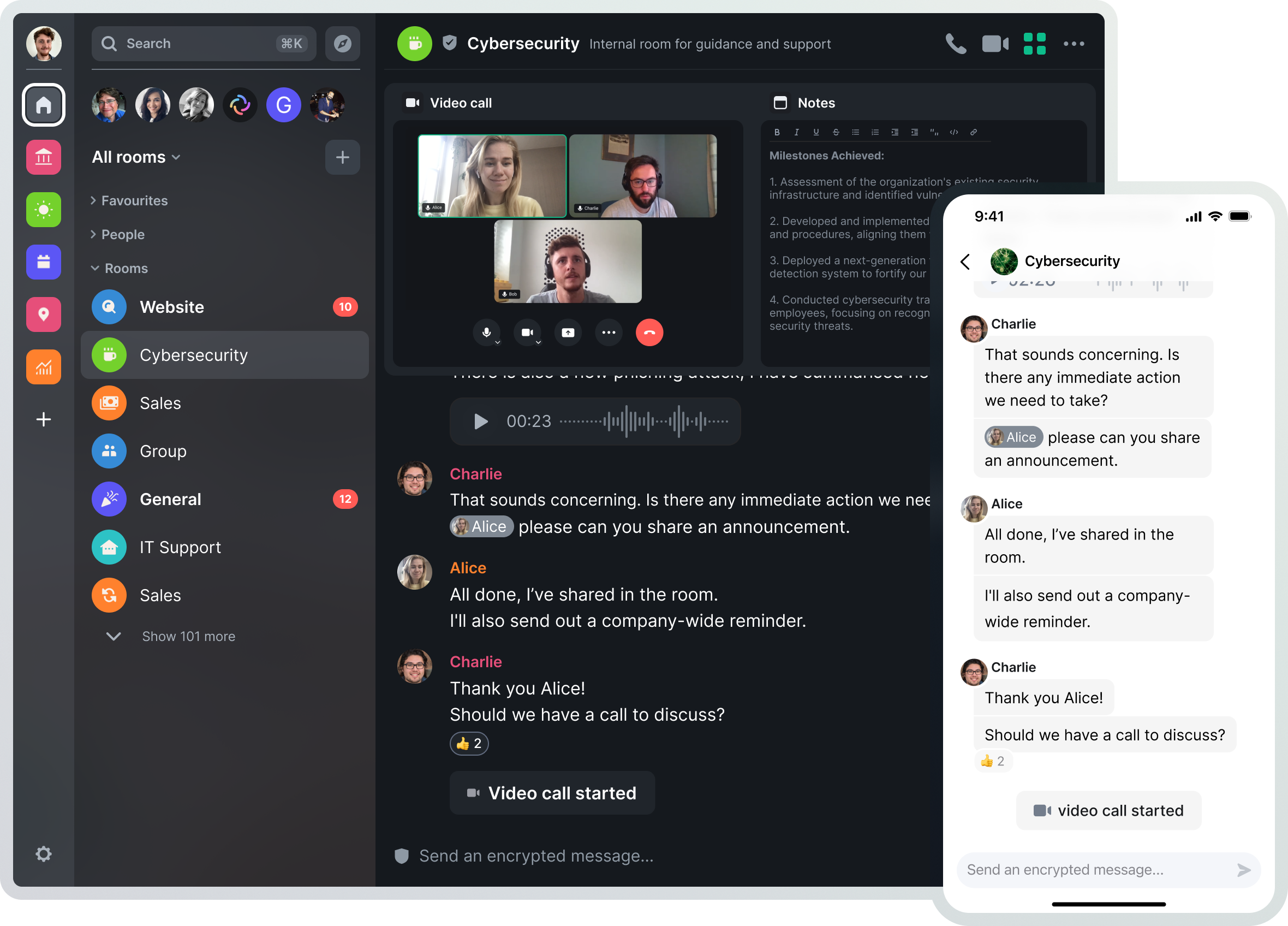
**Navigation Facilitée :** Les spaces facilitent la navigation entre les différentes rooms. Plutôt que d'avoir une liste de toutes les rooms auxquelles nous participons, les spaces nous permettent de visualiser les rooms dans un contexte organisé et compréhensible.

**Thématiques Cohérentes :** Les spaces offrent un moyen de rassembler des conversations ayant des liens thématiques sous une même bannière. Par exemple, un space pourrait regrouper toutes les rooms liées à un projet de développement, un espace personnel ou une communauté spécifique.

**Contrôle d'Accès :** Les propriétaires d'un space peuvent définir des règles d'accès pour les rooms qui en font partie. Cela permet de contrôler qui peut rejoindre les rooms et participer aux discussions.

**Gestion Centralisée :** En tant qu'administrateur d'un space, il est possible de gérer les paramètres du space et les paramètres individuels des rooms qu'il contient. Cela permet une gestion centralisée de l'ensemble des rooms liées à un contexte spécifique.

**Gestion des Conversations :** Les spaces aident à garder les discussions organisées en évitant que les utilisateurs se perdent parmi un grand nombre de rooms indépendantes.



Ci-dessus, voici une capture d’écran du client Element. Les icones à gauche sont des spaces. Ensuite, nous pouvons voir les différentes rooms/bridges de l’utilisateur. Au milieu se trouve une visioconférence en cours et dessous la conversation de la room et à droite la version téléphone.

## Les APIs

Les API du protocole Matrix fournissent des points d'entrée pour que les développeurs puissent créer des applications, des clients personnalisés, des intégrations et des fonctionnalités qui interagissent avec le protocole et les serveurs Matrix. Voici quelques-unes des utilisations principales des API du protocole Matrix :

**Clients Matrix personnalisés :** Les API permettent aux développeurs de créer des clients Matrix personnalisés. Ces clients peuvent avoir des interfaces utilisateur spécifiques, des fonctionnalités uniques et des expériences utilisateur adaptées aux besoins.

**Intégrations :** Les API offrent la possibilité d'intégrer Matrix dans d'autres applications ou services. Cela peut inclure l'ajout de fonctionnalités de discussion en temps réel à des plateformes existantes ou la création de passerelles pour relier Matrix à d'autres protocoles de messagerie.

**Bridges :** Les API permettent la création de bridges entre Matrix et d'autres protocoles de messagerie.

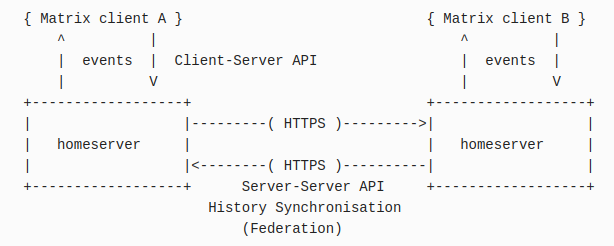
**Automatisation :** Les API permettent de créer des bots et des scripts automatisés qui effectuent des actions spécifiques, comme l'envoi de messages programmés, la gestion des invitations dans les pièces, etc.

**Développement d'applications personnalisées :** Les développeurs peuvent utiliser les API pour créer des applications personnalisées qui utilisent les fonctionnalités du protocole Matrix. Cela peut inclure des applications de gestion, des outils de collaboration, etc.

**Analyse de Données :** Les API permettent d'extraire des données et des statistiques à partir du réseau Matrix. Cela peut aider à comprendre les modèles de communication, à surveiller l'activité des utilisateurs, etc.

**Chiffrement de Bout en Bout :** Les API permettent d'implémenter le chiffrement de bout en bout dans les clients et les applications. Cela assure la sécurité des messages en les chiffrant localement avant l'envoi et en les déchiffrant localement à la réception.

## Principe de communication



Ci-dessus, nous pouvons observer un échange d’événements entre deux clients Matrix qui ne sont pas sur le même serveur.

Par exemple, pour que le client A envoie un message au client B, le client A effectue une requête d’ajout d’un événement (HTTP PUT), dans ce cas un message, au HomeServer (serveur où l’utilisateur est enregistré) à l'aide de l'API client-serveur. Le HomeServer de A ajoute cet événement à sa copie du graphe d'événements de la salle, en signant le message dans le contexte du graphe pour en garantir l'intégrité.

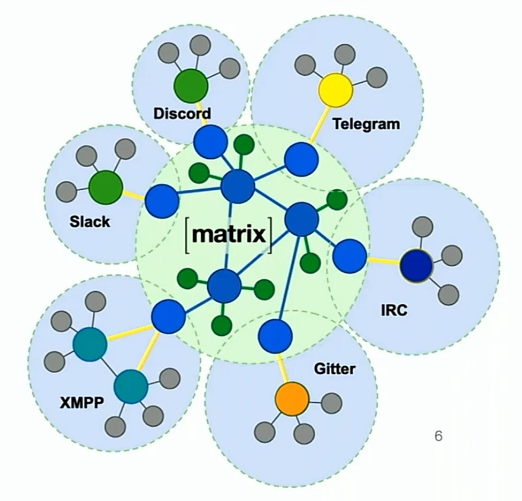
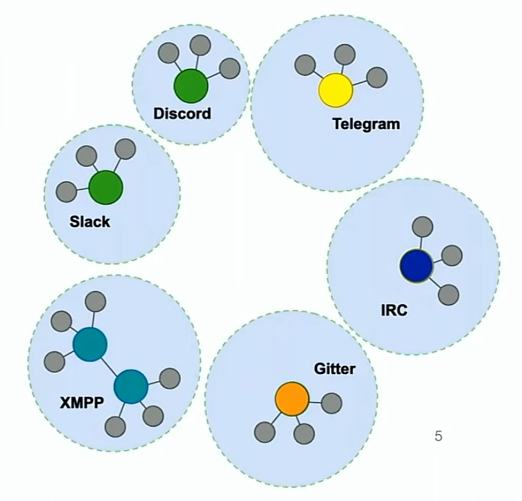
L'Event Graph est une représentation sous forme de graphe qui enregistre l'historique des événements au sein d'une room dans le protocole Matrix. Il assure la cohérence, la synchronisation et l'ordre des interactions et des communications entre les utilisateurs et les serveurs dans le réseau Matrix.

Puis, le HomeServer de A réplique ensuite le message au HomerServer de B en effectuant la même requête (HTTP PUT) à l'aide de l'API serveur-serveur. Le HomeServer de B authentifie la demande, valide la signature de l'événement, autorise le contenu de l'événement et l'ajoute à sa copie du graphe d'événements de la salle. Le client B reçoit ensuite le message de son HomeServer en effectuant une requête de demande au serveur (HTTP GET) pour savoir si des nouveaux événements sont arrivés.

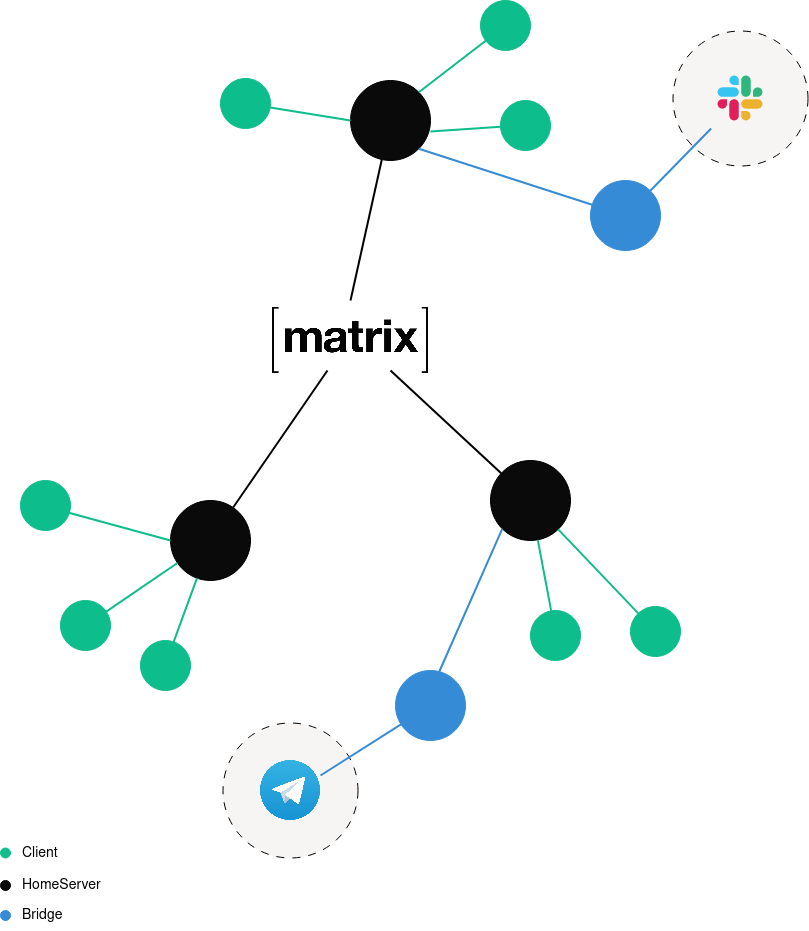
# Matrix – Architecture

Comme cité ci-dessus, l'architecture de Matrix est conçue pour être décentralisée, interopérable et sécurisée. Elle repose sur une combinaison de serveurs, de clients et de protocoles pour permettre des communications en temps réel.

Dans un contexte plus complexe nous pouvons observer que Matrix permet de regrouper différents systèmes entre eux



Si nous faisons un zoom sur l’architecture de Matrix :



Les HomeServer sont les serveurs Matrix qui contiennent les utilisateurs, les rooms, etc. Nous observons la fédération des serveurs grâce au protocole Matrix. Les clients accèdent au serveur afin d’afficher la vue du serveur pour les utilisateurs. Le HomeServer est interopérable avec Telegram et Slack (pour ce schéma) grâce aux bridges qui assurent la communication entre les deux systèmes.

## Les Acteurs

Les principaux acteurs sont :

**Clients :** Les clients Matrix sont des applications utilisées par les utilisateurs pour accéder au réseau. Ils peuvent être des applications de bureau, des applications mobiles ou même des clients basés sur le Web. Les clients permettent aux utilisateurs de se connecter aux serveurs, de rejoindre des pièces, d'envoyer et de recevoir des messages, et d'interagir en temps réel.

**Serveurs Matrix (Homeservers) :** Les serveurs Matrix sont des composants centraux de l'architecture. Chaque utilisateur est lié à un serveur Matrix, qui gère les comptes d'utilisateurs, les messages, les pièces et d'autres données associées à cet utilisateur. Les serveurs Matrix communiquent entre eux via la fédération pour permettre l'interopérabilité.

**Fédération :** La fédération permet aux serveurs Matrix de communiquer entre eux pour partager des messages et des événements. Cela signifie que les utilisateurs sur différents serveurs peuvent interagir comme s'ils étaient sur le même serveur.

**Pièces (Rooms) :** Les pièces sont des espaces où les utilisateurs peuvent communiquer. Les pièces peuvent être publiques (accessibles à tous) ou privées (accessibles uniquement aux membres invités). Les utilisateurs peuvent rejoindre des pièces pour participer à des conversations et échanger des messages.

**Événements :** Le protocole Matrix traite les actions comme des événements. Chaque message, ajout d'utilisateur, modification d'autorisation, etc., est représenté sous forme d'événement dans l'historique de la pièce.

**API :** Les API permettent aux développeurs de créer des applications, des clients personnalisés, des intégrations et des ponts (bridges) pour interagir avec le réseau Matrix. Les API offrent des points d'entrée pour gérer les comptes, les messages, les autorisations, etc.

**Chiffrement de Bout en Bout :** L'architecture prend en charge le chiffrement de bout en bout, ce qui signifie que les messages sont chiffrés localement par les clients avant l'envoi et déchiffrés localement à la réception. Cela assure la confidentialité des messages, même lorsque les données transitent entre les serveurs.

**Extensions et Intégrations :** L'architecture de Matrix est extensible, ce qui permet aux développeurs d'ajouter de nouvelles fonctionnalités et de créer des intégrations avec d'autres services et protocoles.