Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: "Створення та налаштування сервера федеративного месенджера на основі протоколу Matrix"

Виконав: студент

Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)

Маслянка Т. В.
підпис (прізвище та ініціали)

Керівник Козак Р. О.
підпис (прізвище та ініціали)

Нормоконтроль Тимощук Д. І.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н. В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Маслянці Тарасу Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Створення та налаштування сервера федеративного месенджера на основі протоколу Matrix

Керівник роботи Козак Руслан Орестович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «__» ___ 2024 року № _____

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи

4. Зміст роботи (перелік питань, які потрібно розробити)
Проаналізувати вимоги та існуючі рішення
Проаналізувати та дослідити теоретичні основи та принципи роботи протоколу Matrix
Розробити, налаштувати та протестувати сервер федеративного месенджера на основі протоколу Matrix
Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Мариненко С.Ю.		

7. Дата видачі завдання 29.01.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.01 – 30.01	Виконано
2.	Підбір джерел для аналізу існуючих рішень	30.01 – 31.01	Виконано
3.	Проведення аналізу вимог та існуючих рішень	31.01 – 02.02	Виконано
4.	Оформлення першого розділу	05.02 – 06.02	Виконано
5.	Підбір джерел для аналізу теоретичних основ та принципів роботи протоколу Matrix	06.02 – 07.02	Виконано
6.	Аналіз теоретичних основ та принципів роботи протоколу Matrix	06.02 – 09.02	Виконано
7.	Оформлення другого розділу	08.02 – 09.02	Виконано
8.	Створення, конфігурація та тестування сервера федеративного месенджера	03.06 – 12.06	Виконано
9.	Оформлення третього розділу	12.04 – 13.06	Виконано
10.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	12.06 – 13.06	Виконано
11.	Оформлення кваліфікаційної роботи	12.06 – 14.06	Виконано
12.	Нормоконтроль	14.06 – 17.06	Виконано
13.	Перевірка на плагіат	14.06 – 17.06	Виконано
14.	Попередній захист кваліфікаційної роботи	18.06 – 21.06	Виконано
15.	Захист кваліфікаційної роботи	27.06.2024	

Студент _____ Маслянка Т.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Козак Р.О.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Створення та налаштування сервера федеративного месенджера на основі протоколу Matrix // Кваліфікаційна робота ОР «Бакалавр» // Маслянка Тарас Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-42 // Тернопіль, 2024 // С. 57 , рис. – 37, табл. – 0 , кресл. – 0, додат. – 0.

КЛЮЧОВІ СЛОВА: Matrix, Open Source, децентралізація, конфіденційність.

Ця кваліфікаційна робота присвячена дослідженню та впровадженню сервера федеративного месенджера на основі протоколу Matrix. Протокол Matrix є одним із найсучасніших та найперспективніших рішень для створення децентралізованих комунікаційних мереж, які відповідають високим вимогам безпеки та надійності.

У роботі здійснено детальний аналіз технічних вимог до месенджера, включаючи вимоги до функціональності, безпеки та конфіденційності. Розглянуто основні існуючі протоколи комунікаційних систем, їхні переваги та недоліки.

Практична частина роботи включає створення, налаштування та тестування роботи сервера федеративного месенджера на базі Matrix.

Результати дослідження демонструють високу надійність та безпеку комунікаційної системи, побудованої на основі протоколу Matrix. Запропоновані методи та підходи можуть бути використані для створення безпечних комунікаційних мереж у різних сферах діяльності, що потребують високого рівня конфіденційності та захисту даних.

ABSTRACT

Creation and Configuration of a Federated Messenger Server Based on the Matrix Protocol // Thesis of education level «Bachelor» // Maslianka Taras Volodymyrovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group SBs-42 // Ternopil, 2024 // P. 57, fig. – 37, tab. – 0, chair. – 0, added. – 0.

KEYWORDS: Matrix, Open Source, decentralization, privacy.

This qualification paper is devoted to the research and implementation of a federated messenger server based on the Matrix protocol. The Matrix protocol is one of the most modern and promising solutions for creating decentralized communication networks that meet high security and reliability requirements.

The thesis provides a detailed analysis of the technical requirements for the messenger, including requirements for functionality, security, and privacy. The main existing protocols of communication systems, their advantages and disadvantages are considered.

The practical part of the paper includes the creation, configuration, and testing of a federated messenger server based on Matrix protocol.

The results of the paper demonstrate the high reliability and security of the communication system built on the basis of the Matrix protocol. The proposed methods and approaches can be used to create secure communication networks in various fields of activity requiring a high level of confidentiality and data protection.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

XMPP – Extensible Messaging and Presence Protocol.

IRC – Internet Relay Chat.

E2EE – end-to-end encryption.

MITM – man-in-the-middle.

ВСТУП

Сучасне цифрове середовище охоплює широкий спектр технологій, інфраструктури і платформ, які створюють інтерактивну, з'єднану та цифрову інфраструктуру. Інформаційні технології стали невід'ємною частиною нашого повсякденного життя та охоплюють сфери бізнесу та економіки, освіти, медицини, транспорту та логістики, розваг, культури та багатьох інших. Цифрове середовище стає все більш інтегрованим і впливовим у сучасному суспільстві, тому питання безпеки комунікацій набувають все більшого значення. Зі збільшенням обсягів даних, які передаються через інтернет, виникає необхідність у надійних засобах комунікації. Зокрема, це стосується месенджерів, які стали основним інструментом для обміну повідомленнями як в особистому, так і в професійному середовищі. Ось лише деякі з популярних месенджерів на сьогодні:

– WhatsApp: один з найпоширеніших месенджерів у світі, що підтримує текстові повідомлення, голосові і відеодзвінки, обмін файлами тощо;

– Facebook Messenger: месенджер, що інтегрований у соціальну мережу Facebook, з можливістю обміну повідомленнями, відеодзвінками і іншими функціями;

– Telegram: месенджер з акцентом на конфіденційність і безпеку, що підтримує шифрування, канали для масових повідомлень, ботів;

– Signal: месенджер, який відомий своєю високою безпекою та наскрізним шифруванням для всіх типів комунікацій;

– Viber: месенджер з можливістю текстових повідомлень, голосових і відеодзвінків, обміну файлами і стікерами.

Основні проблеми сучасних месенджерів включають недостатній рівень конфіденційності, ризик витоку даних, централізованість управління та можливість цензури. Ці проблеми створюють потребу у використанні нових підходів для забезпечення безпеки та конфіденційності користувачів.

Одним із найбільш перспективних рішень у цій галузі є протокол Matrix, який забезпечує федеративну мережу для безпечного та конфіденційного обміну

повідомленнями. Matrix дозволяє створювати децентралізовані системи комунікації, де кожен сервер може взаємодіяти з іншими, забезпечуючи високу ступінь відмовостійкості та масштабованості.

Метою даного дипломного проекту є створення персонального сервера федеративного месенджера на основі протоколу Matrix. Для досягнення поставленої мети необхідно виконати такі завдання, як:

– огляд сучасних підходів до побудови месенджерів;

– вимоги безпеки в сучасних месенджерах;

– огляд існуючих протоколів;

– теоретичний опис роботи Matrix;

– практичне впровадження, налаштування та тестування сервера.

Очікується, що результати даного проекту будуть корисними для фахівців у галузі кібербезпеки, а також для організацій, які прагнуть підвищити рівень захисту своїх комунікаційних мереж.

1 АНАЛІЗ ВИМОГ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Огляд сучасних підходів до побудови месенджерів

Сучасні месенджери є ключовими інструментами для забезпечення комунікації як у повсякденному житті, так і в професійній діяльності. Вони дозволяють користувачам миттєво обмінюватися текстовими повідомленнями, файлами, здійснювати голосові та відео дзвінки. Розробка месенджерів включає врахування різноманітних вимог та викликів, що виникають у зв'язку з забезпеченням безпеки, конфіденційності та зручності використання.

Месенджери можуть бути класифіковані на декілька груп:

– централізовані месенджери: такі як WhatsApp, Telegram, де всі повідомлення проходять через один центральний сервер. Це забезпечує простоту управління, але створює ризики з точки зору безпеки та конфіденційності, оскільки один сервер є точкою відмови;

– децентралізовані (розподілені) месенджери: наприклад, Matrix або XMPP, де повідомлення розподіляються між багатьма незалежними серверами. Це підвищує стійкість до атак та цензури, але ускладнює управління та забезпечення сумісності між різними серверами.

Хоча децентралізовані та розподілені мережі (див. рисунок 1.1) мають різну архітектуру [1], у протоколі Matrix комунікація між серверами відбувається у децентралізованій мережі, хоча користувачів і залежать від одного сервера.

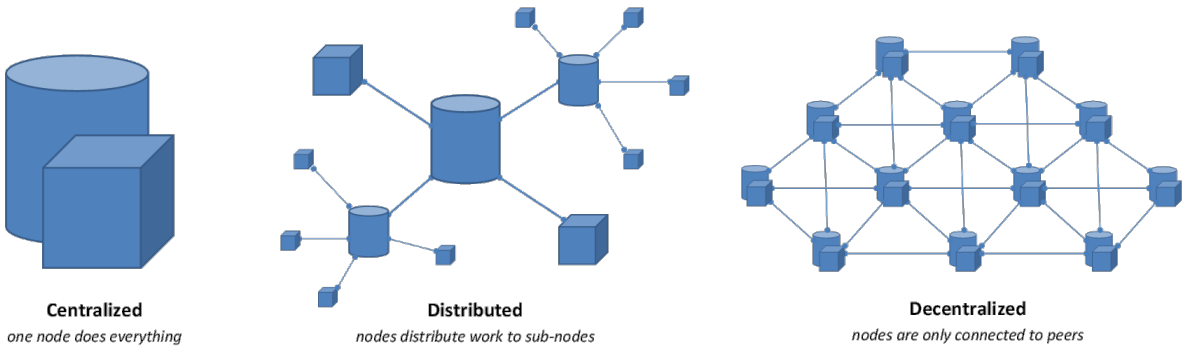

Рисунок 1.1 – Централізована, розподілена та децентралізована мережі

Месенджери можуть використовувати наскрізне шифрування як між користувачами, так і не використовувати його, а вірніше відправляти повідомлення спочатку на сервер, де воно зберігається, а тільки потім іншому користувачу. Наскрізне шифрування між користувачами є критично важливим для забезпечення конфіденційності та безпеки комунікацій у месенджерах.

Месенджери можуть бути розроблені на основі пропрієтарних або відкритих принципів побудови. Пропрієтарні месенджери, такі як WhatsApp, Viber і Facebook Messenger, використовують закриті протоколи та алгоритми, що контролюються виключно їхніми розробниками. Важливо, що користувачі таких месенджерів повинні довіряти розробнику у питанні захисту їхніх даних, оскільки архітектура та механізми безпеки цих платформ є закритими і недоступними для незалежної перевірки. Це може створювати ризики щодо конфіденційності та безпеки даних користувачів, оскільки вони не можуть бути впевнені у відсутності прихованих вразливостей або навмисних функцій для збору даних.

На противагу цьому, відкриті месенджери, такі як Signal, Telegram і Matrix, базуються на відкритих стандартах і протоколах, які доступні для незалежного аудиту та модифікації спільнотою розробників. Це забезпечує прозорість у роботі платформи, дозволяючи будь-кому перевірити надійність алгоритмів шифрування і відсутність вразливостей. Відкриті принципи побудови сприяють більшій довірі з боку користувачів та спільноти безпеки, оскільки будь-які потенційні проблеми можуть бути виявлені та виправлені швидше.

1.2 Вимоги безпеки в сучасних месенджерах

Сучасні вимоги до месенджерів постійно зростають, оскільки користувачі прагнуть до більшої безпеки, конфіденційності та функціональності. Протоколи, що використовуються в месенджерах, повинні відповідати цим вимогам, щоб забезпечити надійність і зручність користування. З розвитком технологій та зростанням вимог до конфіденційності та безпеки, з’являються нові виклики, які

потрібно враховувати при розробці месенджерів. В цьому розділі ми розглянемо основні вимоги до сучасних месенджерів, включаючи безпеку, конфіденційність та децентралізацію, а також проблеми, пов'язані з використанням пропрієтарних месенджерів.

Месенджери відіграють критичну роль у сучасній комунікаційній інфраструктурі. Вони забезпечують миттєвий обмін повідомленнями, підтримку групових чатів, відео- та аудіозв'язок та багато іншого. Завдяки месенджерам, комунікація стала набагато ефективнішою та зручнішою. Проте, з ростом популярності месенджерів, виникають і нові виклики, які потребують уваги [2].

Однією з основних вимог до сучасних месенджерів є забезпечення високого рівня безпеки. Месенджери повинні захищати дані користувачів від несанкціонованого доступу та зловмисних атак. Це включає використання сучасних методів шифрування, автентифікації та управління доступом. Наскрізне шифрування (див. рисунок 1.2), яке гарантує, що тільки відправник і отримувач можуть читати повідомлення [3], є ключовою вимогою до безпечних месенджерів.

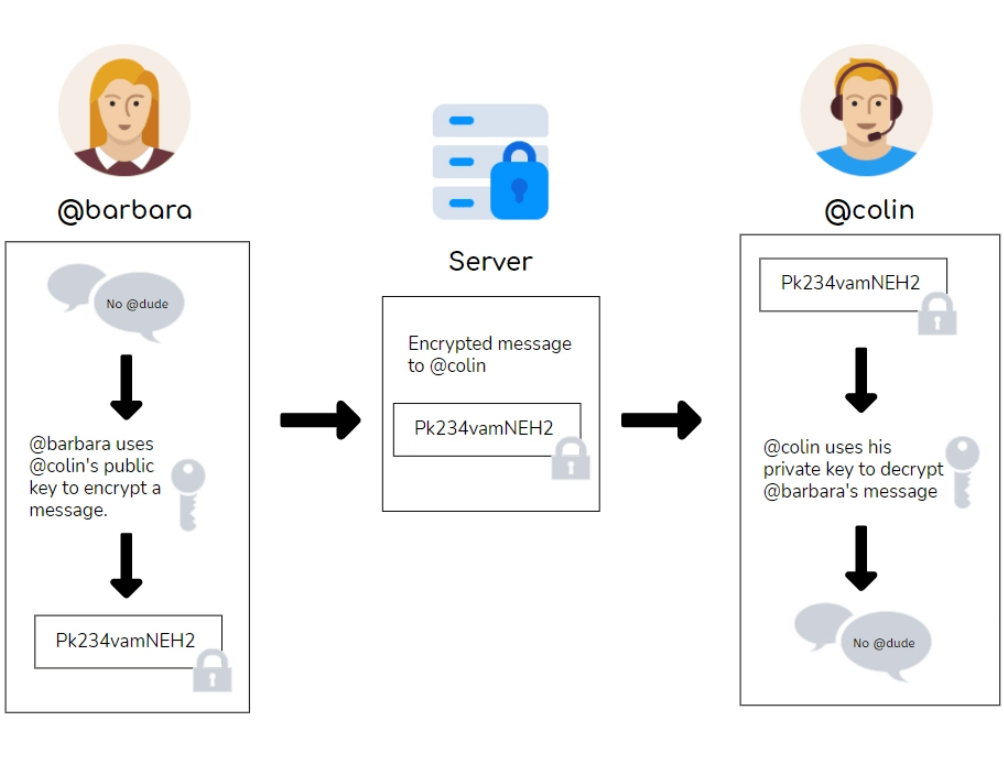

Рисунок 1.2 – Наскрізне шифрування як основний метод захисту даних

Конфіденційність є ще однією критичною вимогою. Користувачі повинні мати впевненість, що їхні особисті дані не будуть використовуватись без їхнього дозволу. Це включає захист метаданих, таких як інформація про відправника, отримувача та час відправлення повідомлень. Месенджери повинні мінімізувати збирання та зберігання персональних даних і забезпечувати можливість видалення даних за запитом користувачів.

Децентралізація є важливою вимогою для забезпечення незалежності та стійкості мережі. Децентралізовані месенджери не залежать від єдиного центрального серверу, що робить їх менш уразливими до атак та збоїв. Вони також сприяють підвищенню конфіденційності, оскільки дані розподілені між багатьма серверами.

Проблеми, пов’язані з пропрієтарними месенджерами, є суттєвими і потребують детального аналізу. Пропрієтарні месенджери часто мають повний контроль над даними користувачів. Це означає, що компанія, яка володіє месенджером, може мати доступ до всіх повідомлень та особистих даних користувачів. Це створює ризик витоку інформації та зловживання даними.

Цензура є ще однією проблемою пропрієтарних месенджерів. Компанії можуть обмежувати або блокувати контент на свій розсуд, що може порушувати права користувачів на свободу слова. Це особливо актуально у країнах з авторитарними режимами, де уряди можуть примушувати компанії до цензурування контенту.

Реклама у пропрієтарних месенджерах також викликає занепокоєння. Для отримання прибутку, компанії можуть використовувати дані користувачів для таргетованої реклами. Це не тільки порушує конфіденційність, але й може впливати на якість користувацького досвіду.

У зв’язку з вищезазначеними вимогами та проблемами, стає очевидним, що сучасні месенджери повинні відповідати високим стандартам безпеки, конфіденційності та децентралізації. Тільки тоді користувачі можуть бути впевнені у захисті своїх даних та свобод. У наступних розділах ми розглянемо існуючі протоколи месенджерів, їхні переваги та недоліки, а також вимоги до

нових протоколів, які здатні забезпечити належний рівень захисту та конфіденційності.

Також сучасні месенджери повинні підтримувати різні типи комунікацій, включаючи текстові повідомлення, аудіо- та відеодзвінки. Це забезпечує зручність та гнучкість у використанні, дозволяючи користувачам вибирати найзручніший спосіб комунікації залежно від ситуації. Підтримка мультимедійних комунікацій також включає можливість обміну файлами, зображеннями, відео та іншими типами даних.

Таким чином, аналіз вимог до месенджерів дозволяє сформулювати ключові критерії, яким повинні відповідати сучасні комунікаційні інструменти. Це забезпечить не тільки безпеку та конфіденційність, але й сприятиме розвитку більш відкритих та децентралізованих мереж.

1.3 Огляд існуючих протоколів месенджерів

У цьому розділі ми розглянемо основні протоколи, які використовуються у сучасних месенджерах, включаючи IRC, XMPP та Signal. Проаналізуємо їхні переваги та недоліки, а також порівняємо їх за ключовими параметрами, такими як безпека, підтримка децентралізації та функціональність. На основі цього аналізу зробимо висновки про потребу в нових підходах до протоколів для месенджерів.

IRC (Internet Relay Chat) – один з найстаріших протоколів обміну повідомленнями, був розроблений у 1988 році. Він забезпечує текстовий обмін повідомленнями в реальному часі між користувачами в різних чат-кімнатах (каналах) [4]. Схема мережі IRC зображена на рисунок 1.3.

Переваги IRC:

– простота та стабільність: завдяки простоті своєї архітектури IRC залишається стабільним та надійним протоколом;

– широка підтримка: IRC підтримується багатьма клієнтами та платформами.

Недоліки IRC:

– відсутність шифрування: IRC не має вбудованого механізму шифрування, що робить його вразливим до прослуховування;

– центральна структура: IRC-сервери є централізованими, що може створювати точку відмови та робить їх уразливими до атак.

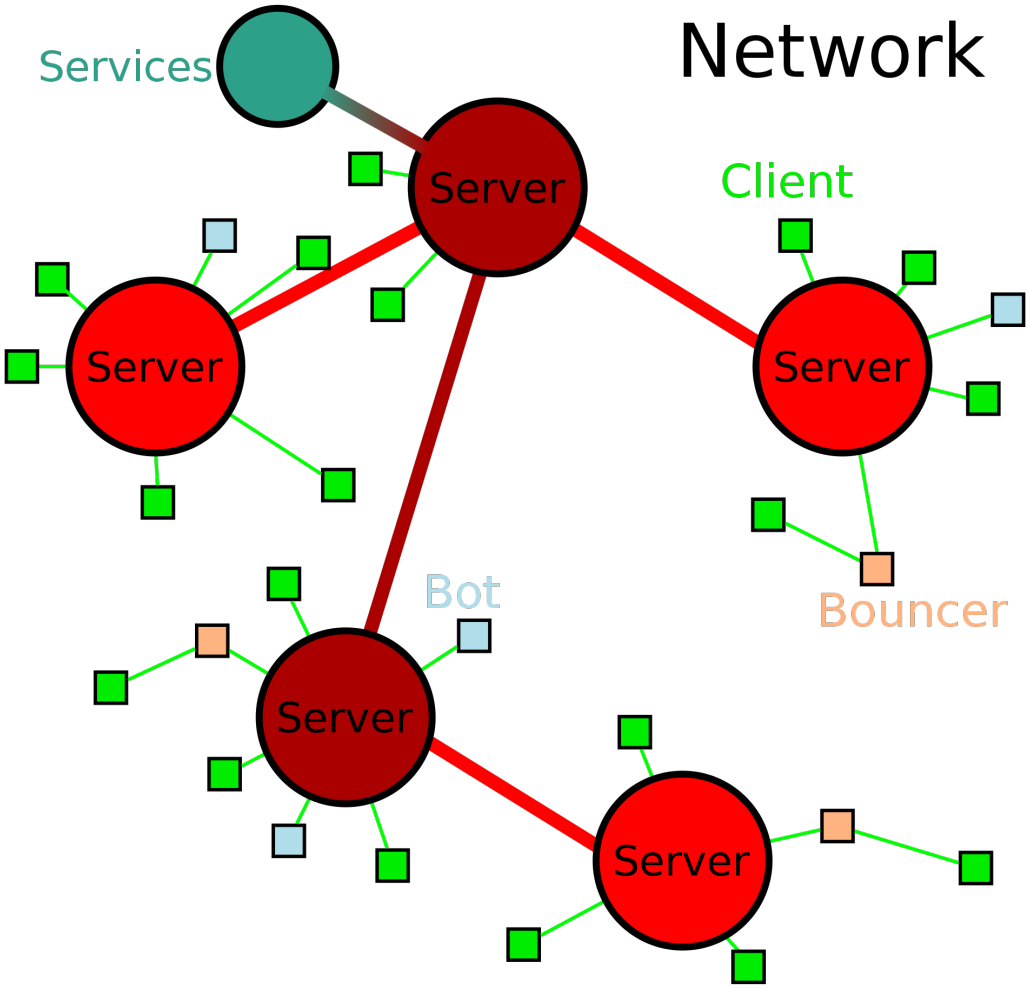

Рисунок 1.3 – Схема мережі IRC

XMPP (Extensible Messaging and Presence Protocol) – розроблений у 1999 році, є розширюваним протоколом для обміну повідомленнями і підтримки присутності в мережі. Він забезпечує більш сучасні можливості порівняно з IRC, включаючи підтримку різних типів даних та шифрування [5]. Схема мережі XMPP зображена на рисунку 1.4.

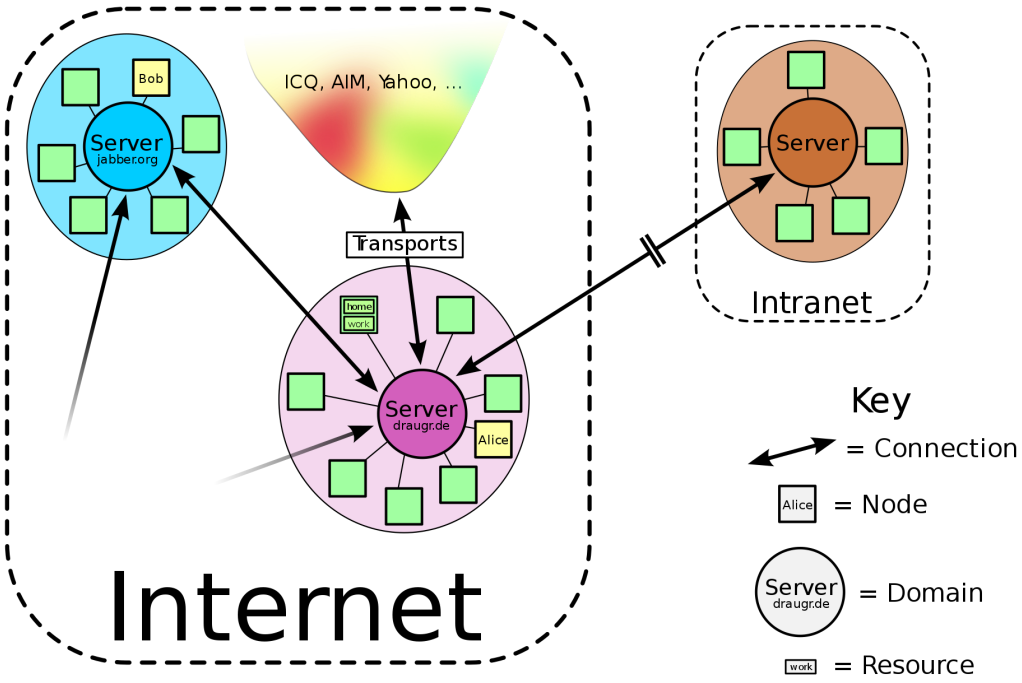

Рисунок 1.4 – Схема мережі XMPP

Переваги XMPP:

– розширюваність: XMPP дозволяє додавати нові функції за допомогою розширень (XEPs), що робить його гнучким;

– підтримка шифрування: XMPP підтримує наскрізне шифрування через розширення, що забезпечує високий рівень безпеки.

Недоліки XMPP:

– складність налаштування: XMPP може бути складним для налаштування та управління, особливо для новачків;

– проблеми зі сумісністю: існують деякі проблеми зі сумісністю між різними XMPP-клієнтами та серверами.

Signal – сучасний протокол, розроблений для забезпечення високого рівня безпеки та конфіденційності. Він використовує наскрізне шифрування для всіх повідомлень і дзвінків, забезпечуючи захист даних користувачів [6]. Інтерфейс даного додатку зображено на рисунку 1.5.

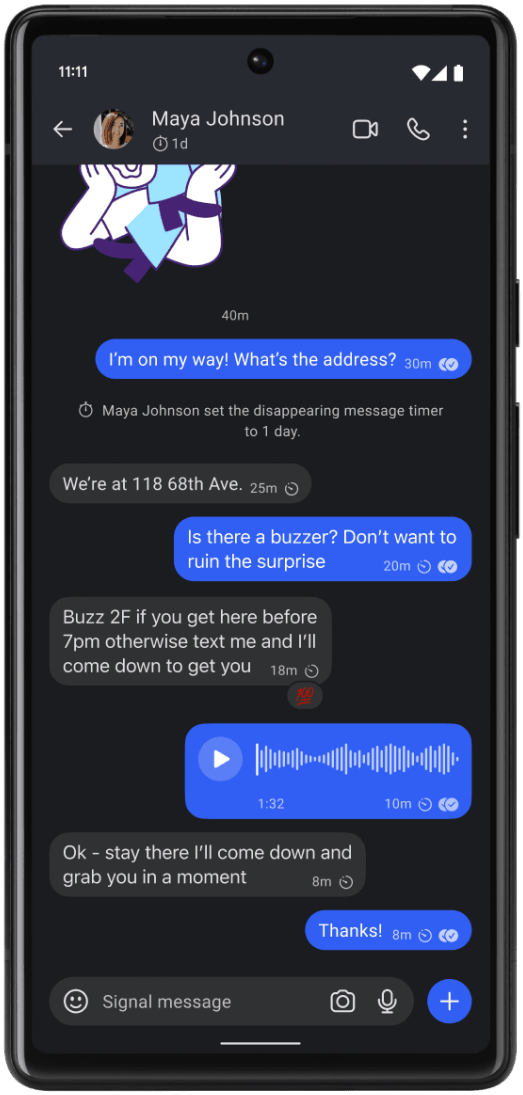

Рисунок 1.5 – Інтерфейс додатку Signal

Переваги Signal:

– висока безпека: Signal використовує потужні методи шифрування, що забезпечує захист даних навіть у випадку компрометації серверів;

– відкритий вихідний код: код Signal є відкритим, що дозволяє незалежним експертам перевіряти його на наявність уразливостей.

Недоліки Signal:

– центральна структура: незважаючи на високий рівень безпеки, Signal залежить від центрального сервера, що може створювати певні ризики;

– обмежена підтримка: Signal не підтримує деякі функції, такі як федерація між різними серверами, що обмежує його масштабованість.

Порівняння цих протоколів за ключовими параметрами, такі як безпека:

– IRC: відсутність вбудованого шифрування робить IRC менш безпечним порівняно з іншими протоколами;

– XMPP: підтримка розширень для шифрування забезпечує високий рівень безпеки, проте потребує додаткової налаштування;

– Signal: використання наскрізного шифрування робить Signal одним з найбезпечніших протоколів.

Підтримка децентралізації:

– IRC: центральна структура серверів обмежує можливості децентралізації;

– XMPP: підтримка федерації між серверами забезпечує високий рівень децентралізації;

– Signal: центральний сервер обмежує можливості децентралізації.

Функціональність:

– IRC: простий текстовий обмін повідомленнями без підтримки складних функцій;

– XMPP: підтримка різних типів даних та розширень, що робить його гнучким;

– Signal: високий рівень безпеки та підтримка текстових, голосових та відеоповідомлень.

Аналіз існуючих протоколів показує, що кожен з них має свої переваги та недоліки. IRC є простим та стабільним, але не забезпечує достатнього рівня безпеки. XMPP є гнучким та розширюваним, але складним у налаштуванні. Signal забезпечує високий рівень безпеки, але залежить від центрального сервера.

Це підкреслює потребу в нових підходах до розробки протоколів для месенджерів, які можуть поєднувати переваги існуючих рішень та уникати їхніх недоліків. У наступних розділах буде розглянуто, як протокол Matrix вирішує ці завдання і які його основні переваги порівняно з іншими протоколами.

2 ТЕОРЕТИЧНІ ОСНОВИ ТА ПРИНЦИПИ РОБОТИ ПРОТОКОЛУ MATRIX

2.1 Принципи роботи протоколу Matrix

Протокол Matrix є одним з найсучасніших і найперспективніших рішень для федеративних месенджерів. Він був створений з метою забезпечення децентралізованої комунікації, високого рівня безпеки і широких можливостей інтеграції. У цьому розділі ми детально розглянемо архітектуру та принципи роботи протоколу Matrix, його основні компоненти, механізми шифрування, федерацію, а також його переваги та недоліки порівняно з іншими протоколами.

Протокол Matrix базується на децентралізованій моделі, де кожен сервер може взаємодіяти з іншими серверами, утворюючи розподілену мережу, як це зображено на рисунку 2.1. Це дозволяє уникнути централізованої точки відмови та підвищує загальну стійкість системи.

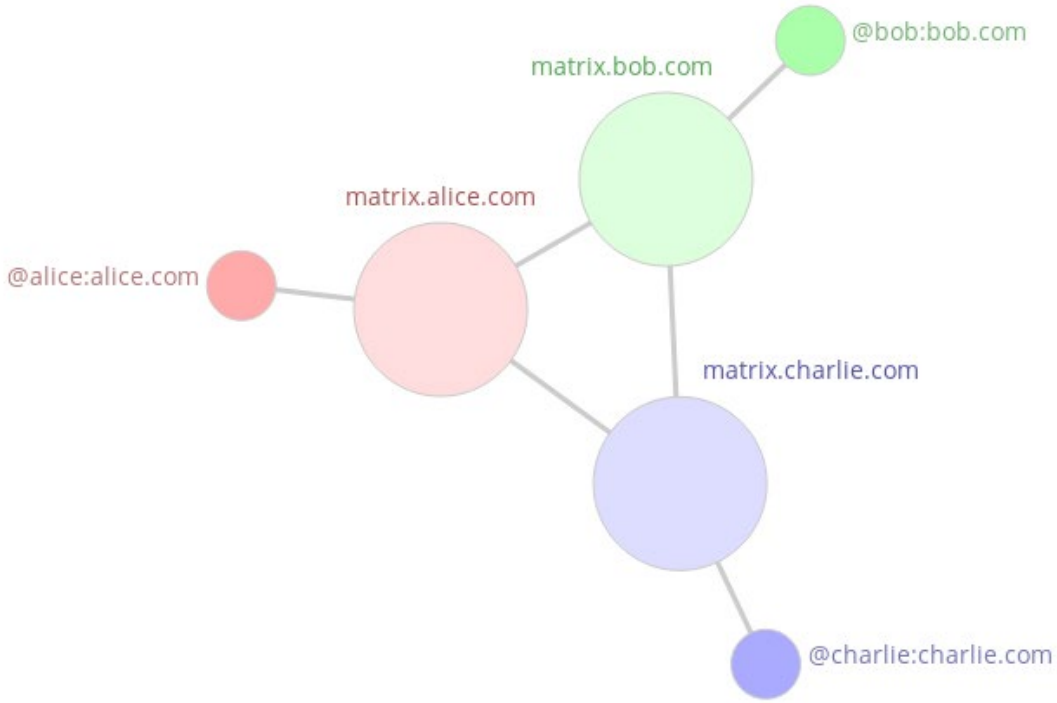

Рисунок 2.1 – Архітектура протоколу Matrix

Ключові компоненти архітектури Matrix включають:

– сервери: вузли мережі, які зберігають дані та забезпечують комунікацію між клієнтами;

– клієнти: додатки або пристрої, які використовують користувачі для надсилання та отримання повідомлень;

– API: інтерфейси для взаємодії між клієнтами та серверами, а також між серверами.

Основні принципи роботи протоколу:

– відкритість та децентралізація: однією з головних переваг протоколу Matrix є його децентралізована природа. Кожен сервер може взаємодіяти з іншими серверами без необхідності в центральному вузлі. Це забезпечує високу стійкість до збоїв і атак, а також дозволяє зберігати дані локально на кожному сервері, що підвищує рівень конфіденційності;

– синхронізація даних: протокол Matrix використовує модель «глобального журналу» для зберігання всіх повідомлень та подій. Кожен сервер зберігає копію цього журналу, що дозволяє їм синхронізувати дані між собою. Це забезпечує консистентність даних у всій мережі та дозволяє користувачам отримувати доступ до своїх повідомлень з будь-якого пристрою.

Matrix підтримує наскрізне шифрування, що забезпечує високий рівень безпеки комунікацій. Наскрізне шифрування гарантує, що тільки відправник і отримувач можуть прочитати повідомлення, навіть якщо дані перехоплені під час передачі.

Федерація є ключовою особливістю протоколу Matrix, яка дозволяє різним серверам обмінюватися повідомленнями та подіями. Це дозволяє створювати глобальну мережу комунікацій, де користувачі можуть спілкуватися незалежно від того, на якому сервері вони зареєстровані.

Кожен сервер у мережі Matrix може обмінюватися подіями з іншими серверами через федераційний API. Це дозволяє забезпечити безперервну комунікацію між користувачами на різних серверах та забезпечує високу доступність даних.

Протокол Matrix має ряд переваг, які роблять його привабливим для використання у федеративних месенджерах:

– децентралізація: відсутність єдиної точки відмови підвищує стійкість системи;

– безпека: наскрізне шифрування забезпечує високий рівень конфіденційності;

– відкритість: відкритий вихідний код дозволяє незалежним експертам перевіряти програмне забезпечення на наявність уразливостей;

– сумісність: підтримка інтеграції з іншими сервісами та протоколами розширює можливості використання.

Однак, є й деякі недоліки:

– складність налаштування: встановлення та налаштування серверів Matrix може бути складним завданням для новачків;

– продуктивність: підтримка глобального журналу може вимагати значних ресурсів для синхронізації даних між серверами.

Протокол Matrix є потужним інструментом для створення федеративних месенджерів. Його децентралізована архітектура, високий рівень безпеки та підтримка федерації роблять його привабливим вибором для сучасних комунікаційних систем. Незважаючи на деякі складнощі з налаштуванням та продуктивністю, Matrix забезпечує надійність та функціональність, необхідні для сучасних месенджерів. У наступних розділах ми детально розглянемо конкретні приклади використання протоколу Matrix та способи його інтеграції з іншими системами.

2.2 Архітектура та структура даних у Matrix

Архітектура протоколу Matrix є однією з ключових особливостей, що забезпечує його гнучкість, масштабованість та надійність. Вона включає домашні сервери, кімнати та події, які взаємодіють між собою для забезпечення

стабільної роботи системи. У цьому розділі ми детально розглянемо кожен з цих компонентів, їхні функції та принципи роботи.

Домашні сервери є основними компонентами мережі Matrix. Кожен користувач підключений до свого домашнього сервера, який відповідає за зберігання даних користувача, управління автентифікацією та авторизацією, а також за обмін повідомленнями з іншими серверами.

Основні функції домашніх серверів (див. рисунок 2.2) включають:

– зберігання даних користувача, таких як історія повідомлень, контакти;

– забезпечення автентифікації та авторизації користувачів;

– передача повідомлень між користувачами та іншими серверами у мережі;

– синхронізація даних з іншими серверами для забезпечення цілісності та доступності інформації.

Домашні сервери взаємодіють між собою через федеративну мережу. Вони обмінюються повідомленнями та подіями, використовуючи стандартні протоколи передачі даних. Це забезпечує можливість користувачам спілкуватися між собою, незалежно від того, на якому сервері вони знаходяться.

Кімнати (rooms) є віртуальними просторами, де користувачі можуть спілкуватися один з одним. Кожна кімната має унікальний ідентифікатор та певні налаштування, що визначають її властивості та поведінку.

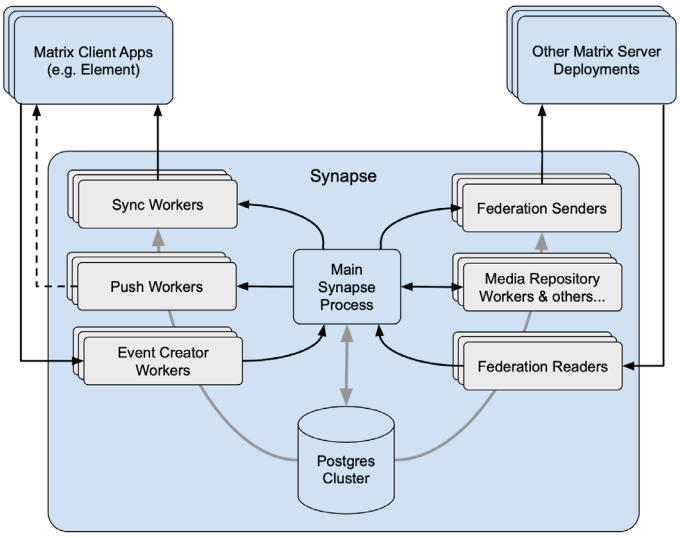

Рисунок 2.2 – Архітектура домашніх серверів у Matrix

Користувачі можуть створювати нові кімнати, запрошувати інших учасників, змінювати налаштування кімнати та керувати правами доступу. Це дозволяє створювати приватні та публічні кімнати, які можуть бути використані для різних цілей, від особистого спілкування до проведення групових дискусій.

Історія повідомлень у кімнатах зберігається на домашніх серверах, що дозволяє користувачам переглядати старі повідомлення та синхронізувати історію між різними пристроями. Це забезпечує зручність використання та доступність інформації у будь-який час.

Події (events) є основним елементом даних у Matrix. Кожна подія представляє окрему дію, таку як відправка повідомлення, додавання користувача до кімнати або зміна налаштувань.

Кожна подія має унікальний ідентифікатор та включає інформацію про тип події, відправника, одержувача та вміст. Події зберігаються у вигляді JSON-об’єктів (див. рисунок 2.3), що дозволяє легко передавати та обробляти їх у мережі. Протокол передачі даних Matrix забезпечує надійну та ефективну синхронізацію подій між серверами, гарантуючи цілісність та послідовність даних.

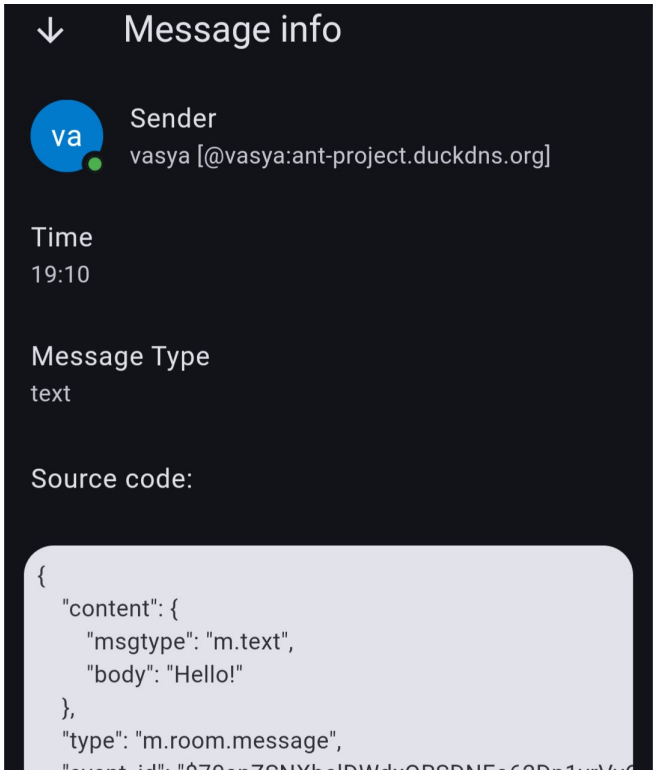

Рисунок 2.3 – Source code події у форматі JSON

Matrix підтримує різні типи подій, включаючи:

– повідомлення (message) – текстові, мультимедійні та інші повідомлення, які відправляються у кімнатах;

– станові події (state events) – події, що змінюють стан кімнати, такі як зміна теми, додавання або видалення учасників;

– події реакцій (reaction events) – події, що представляють реакції на інші повідомлення, такі як лайки або емодзі.

Архітектура та структура даних у протоколі Matrix забезпечують високу гнучкість та масштабованість системи. Домашні сервери, кімнати та події взаємодіють між собою для забезпечення надійної та безперебійної роботи мережі. Завдяки такій архітектурі Matrix може забезпечити ефективну та безпечну комунікацію між користувачами у федеративній мережі.

2.3 Принципи федерації в Matrix

Федерація є однією з основних характеристик протоколу Matrix, яка дозволяє забезпечити децентралізовану та масштабовану систему комунікацій. Федерація дозволяє незалежним серверам взаємодіяти один з одним, створюючи єдину мережу для обміну повідомленнями та іншими даними. У цьому розділі ми детально розглянемо поняття федерації, принципи взаємодії серверів у федеративній мережі, а також переваги такого підходу.

Федерація у контексті Matrix означає об’єднання окремих серверів у єдину мережу. Кожен сервер у федеративній мережі є незалежним і може мати власну політику безпеки, конфіденційності та управління користувачами. Це забезпечує децентралізацію та підвищує стійкість системи до збоїв та атак.

Федерація є важливою з кількох причин:

– децентралізація: відсутність центрального контролюючого органу робить систему менш вразливою до атак і збоїв. У разі виходу з ладу одного сервера, інші сервери продовжують працювати незалежно;

– масштабованість: додавання нових серверів до мережі дозволяє розподіляти навантаження та обслуговувати більшу кількість користувачів без значного зниження продуктивності;

– конфіденційність: користувачі можуть вибирати сервери з політиками конфіденційності, які відповідають їхнім потребам, забезпечуючи тим самим контроль над власними даними.

Взаємодія серверів у федеративній мережі базується на стандартизованих протоколах передачі даних, які забезпечують цілісність та доступність інформації для всіх учасників мережі.

Принципи взаємодії серверів:

– синхронізація даних: сервери обмінюються даними для забезпечення синхронізації повідомлень та подій між користувачами на різних серверах. Це гарантує, що всі учасники кімнати отримують однакові повідомлення незалежно від того, на якому сервері вони зареєстровані;

– передача подій: кожен сервер передає події, такі як нові повідомлення або зміни стану, іншим серверам, що беруть участь у відповідних кімнатах. Події зберігаються у вигляді ланцюжків, що забезпечує їх послідовність та цілісність;

– верифікація: сервери перевіряють автентичність подій та правомірність їх передачі, щоб забезпечити цілісність даних. Це включає верифікацію підписів подій та контроль доступу до кімнат.

Федерація у Matrix реалізована за допомогою серверних API, які дозволяють серверам обмінюватися подіями. Коли користувач відправляє повідомлення у кімнаті, його домашній сервер передає цю подію іншим серверам, які беруть участь у кімнаті. Це забезпечує синхронізацію даних між серверами та дозволяє всім користувачам отримувати актуальну інформацію.

Федерація у Matrix надає кілька ключових переваг, що роблять цей протокол привабливим для різних сценаріїв використання.

Федеративна архітектура дозволяє легко масштабувати систему, додаючи нові сервери для обслуговування зростаючої кількості користувачів. Кожен сервер може обслуговувати свою групу користувачів, розподіляючи

навантаження та знижуючи ризик перевантаження окремих вузлів. Це особливо важливо для великих організацій та спільнот, де кількість користувачів може значно варіюватися.

Завдяки децентралізованій природі федерації, система є більш стійкою до збоїв. Якщо один сервер виходить з ладу, інші сервери продовжують працювати, забезпечуючи безперебійний доступ до даних та функціональності. Це підвищує надійність системи та знижує ризик втрати даних.

Федерація дозволяє користувачам вибирати сервери з політиками конфіденційності та безпеки, які відповідають їхнім потребам. Це забезпечує більший контроль над особистими даними та дозволяє користувачам уникати централізованого контролю з боку великих компаній. Користувачі можуть вибирати сервери, які вони вважають найбільш надійними та безпечними, що підвищує рівень довіри до системи.

Незважаючи на численні переваги, федерація також має свої виклики, які потрібно враховувати під час впровадження та використання протоколу Matrix.

Одним з викликів федерації є забезпечення належного управління правами доступу та контролю за автентифікацією користувачів. Кожен сервер повинен мати надійні механізми контролю доступу для захисту даних та забезпечення безпеки комунікацій.

Оскільки федерація передбачає обмін даними між різними серверами, можуть виникати затримки під час передачі подій. Це може впливати на продуктивність системи, особливо у великих мережах з високою інтенсивністю трафіку. Для мінімізації затримок необхідно оптимізувати механізми передачі даних та забезпечити ефективну роботу серверів.

Забезпечення сумісності між різними реалізаціями серверів та клієнтів є ще одним викликом федерації. Сервери та клієнти повинні дотримуватися стандартів протоколу Matrix для забезпечення безперебійної взаємодії. Це вимагає ретельного тестування та постійного оновлення програмного забезпечення.

Отже, федерація у протоколі Matrix є ключовою особливістю, що забезпечує децентралізацію, масштабованість та підвищену безпеку комунікаційної мережі. Завдяки можливості об'єднання незалежних серверів у єдину мережу, Matrix пропонує гнучке та надійне рішення для сучасних комунікаційних систем. Незважаючи на певні виклики, такі як керування правами доступу та забезпечення продуктивності, переваги федерації значно переважають, роблячи Matrix привабливим вибором для різних сценаріїв використання.

2.4 Безпека та конфіденційність у Matrix

Безпека та конфіденційність є критично важливими аспектами будь-якої комунікаційної системи, особливо у федеративних мережах, таких як Matrix. У цьому розділі ми розглянемо принципи наскрізного шифрування, методи захисту даних на сервері та питання конфіденційності, включаючи збереження анонімності користувачів та захист від втручання третіх осіб.

Наскрізне шифрування (end-to-end encryption, E2EE) є основним методом забезпечення конфіденційності у Matrix. Цей метод забезпечує шифрування повідомлень від моменту їх відправки до моменту отримання, що гарантує, що тільки відправник та одержувач можуть прочитати вміст повідомлення.

Принципи роботи:

– ключі шифрування: для кожної сесії комунікації генерується унікальна пара ключів шифрування. Відправник шифрує повідомлення за допомогою публічного ключа одержувача, а одержувач розшифровує його за допомогою свого приватного ключа;

– механізми автентифікації: для забезпечення цілісності повідомлень та захисту від атак посередника (MITM) використовуються механізми автентифікації, такі як протоколи ключової домовленості та цифрові підписи;

– менеджмент ключів: Matrix використовує протоколи управління ключами, що забезпечують безпечне зберігання та обмін ключами між користувачами.

Переваги:

– конфіденційність: наскрізне шифрування гарантує, що навіть адміністратори серверів не можуть прочитати вміст повідомлень, забезпечуючи тим самим високу ступінь конфіденційності;

– цілісність: захищає повідомлення від несанкціонованих змін під час передачі;

– безпека: захищає дані від перехоплення та зловмисних атак.

Приклади використання:

– приватні повідомлення: наскрізне шифрування забезпечує конфіденційність приватних повідомлень між користувачами;

– чат-кімнати: захист конфіденційної інформації в групових чатах;

– обмін файлами: безпечний обмін файлами та іншими даними.

Захист даних на сервері є важливим аспектом забезпечення безпеки у федеративних мережах. Сервери повинні мати надійні механізми захисту від несанкціонованого доступу, втрати даних та зловмисних атак.

Методи захисту:

– шифрування даних: використання шифрування для зберігання даних на сервері, що захищає інформацію у випадку компрометації сервера;

– контроль доступу: впровадження строгих механізмів контролю доступу для обмеження доступу до даних тільки авторизованим користувачам;

– регулярні аудити: проведення регулярних аудитів безпеки для виявлення та усунення потенційних вразливостей.

Управління доступом включає в себе аутентифікацію користувачів та контроль за їхніми правами доступу до різних ресурсів. Це забезпечує, що тільки авторизовані користувачі можуть взаємодіяти з конфіденційними даними.

Конфіденційність користувачів є ключовим аспектом, який потребує особливої уваги у федеративних мережах. Це включає в себе як захист особистих даних, так і забезпечення анонімності користувачів.

Збереження анонімності користувачів може бути досягнуто за допомогою різних методів, таких як використання псевдонімів та анонімних облікових записів:

– псевдоніми: користувачі можуть використовувати псевдоніми замість реальних імен для збереження анонімності;

– анонімні облікові записи: можливість створення облікових записів без прив'язки до особистих даних користувача.

Захист від втручання третіх осіб включає в себе захист від зловмисних атак, перехоплення даних та інших загроз:

– захист від атак посередника (MITM): використання протоколів автентифікації та шифрування для запобігання перехопленню даних;

– моніторинг безпеки: постійний моніторинг мережевої активності для виявлення підозрілих дій та атак.

Отже, забезпечення безпеки та конфіденційності у протоколі Matrix є складним та багатогранним завданням, яке включає в себе наскрізне шифрування, захист даних на сервері та питання конфіденційності користувачів. Наскрізне шифрування забезпечує конфіденційність повідомлень, захист даних на сервері гарантує їх безпеку, а методи збереження анонімності та захист від втручання третіх осіб роблять протокол Matrix надійним та безпечним рішенням для сучасних комунікацій. Незважаючи на виклики, такі як управління ключами та контроль доступу, переваги безпеки та конфіденційності роблять Matrix привабливим вибором для різних сценаріїв використання.

3 ПРАКТИЧНА ЧАСТИНА

3.1 Попередні вимоги для встановлення сервера

Для встановлення власного сервера месенджера на основі протоколу Matrix потрібна машина, на якій він буде розміщений. Це може бути як і власний локальний сервер, так і віртуальний сервер у хмарного провайдера, наприклад, AWS, Azure Cloud, Google Cloud або Oracle Cloud. На сервері повинна бути встановлена стабільна версія операційної системи Ubuntu або її еквівалент. На рисунку 3.1 зображений вхід у віртуальний приватний сервер, розміщений у Oracle Cloud, на якому запущена ОС Ubuntu 22.04 LTS.

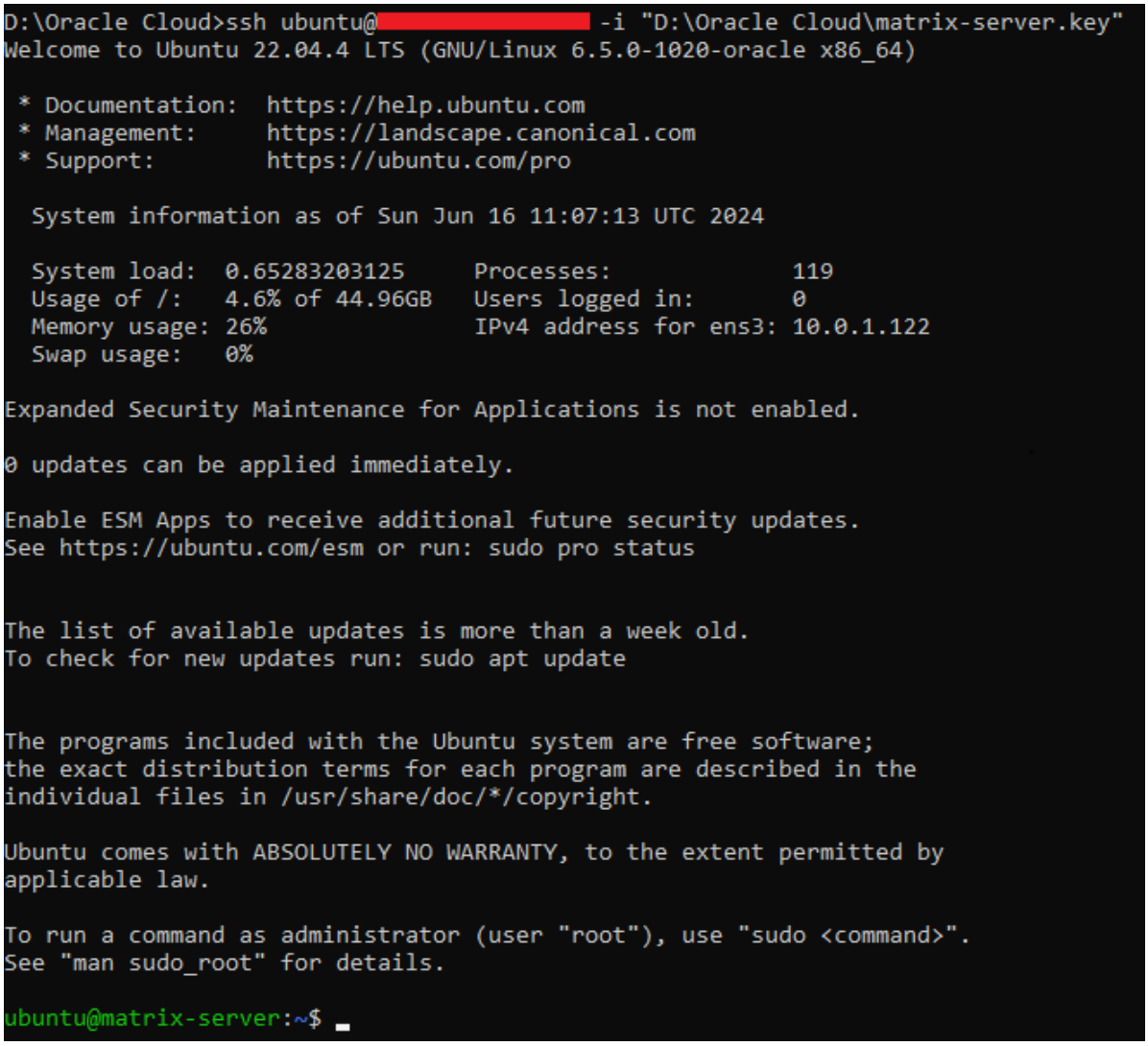

Рисунок 3.1 – Вхід по SSH до віртуального сервера

Коли ми знаходимося в нашій консолі сервера, потрібно переконатися, що ми використовуємо саме новіше програмне забезпечення, безпосередньо оновивши його, як це зображено на рисунку 3.2.

```
ubuntu@matrix-server:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  linux-headers-6.5.0-1023-oracle linux-image-6.5.0-1023-oracle linux-modules-6.5.0-1023-oracle
  linux-modules-extra-6.5.0-1023-oracle linux-oracle-6.5-headers-6.5.0-1023
The following packages will be upgraded:
  bind9-dnsutils bind9-host bind9-libs cloud-init cpio distro-info-data git git-man klibc-utils landscape-common less
  libarchive13 libc-bin libc6 libglib2.0-0 libglib2.0-bin libglib2.0-data libklibc libnghttp2-14 libtss2-esys-3.0.2-0
  libtss2-mu0 libtss2-sys1 libtss2-tcti-cmd0 libtss2-tcti-device0 libtss2-tcti-mssim0 libtss2-tcti-swtpm0
  linux-headers-oracle linux-image-oracle linux-oracle locales openssh-client openssh-server openssh-sftp-server
  python3-idna python3-jinja2 snapd ubuntu-advantage-tools ubuntu-pro-client ubuntu-pro-client-l10n vim vim-common
  vim-runtime vim-tiny xxd
44 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
27 standard LTS security updates
Need to get 187 MB of archives.
After this operation, 727 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Рисунок 3.2 – Оновлення пакетів в Ubuntu

Також для сервера потрібно мати власне доменне ім’я. Безкоштовно його можна отримати через сервіс Duck DNS (див. рисунок 3.3), який надає у безоплатне користування піддомен свого сервісу, який ми і будемо використовувати для створення нашого сервера.

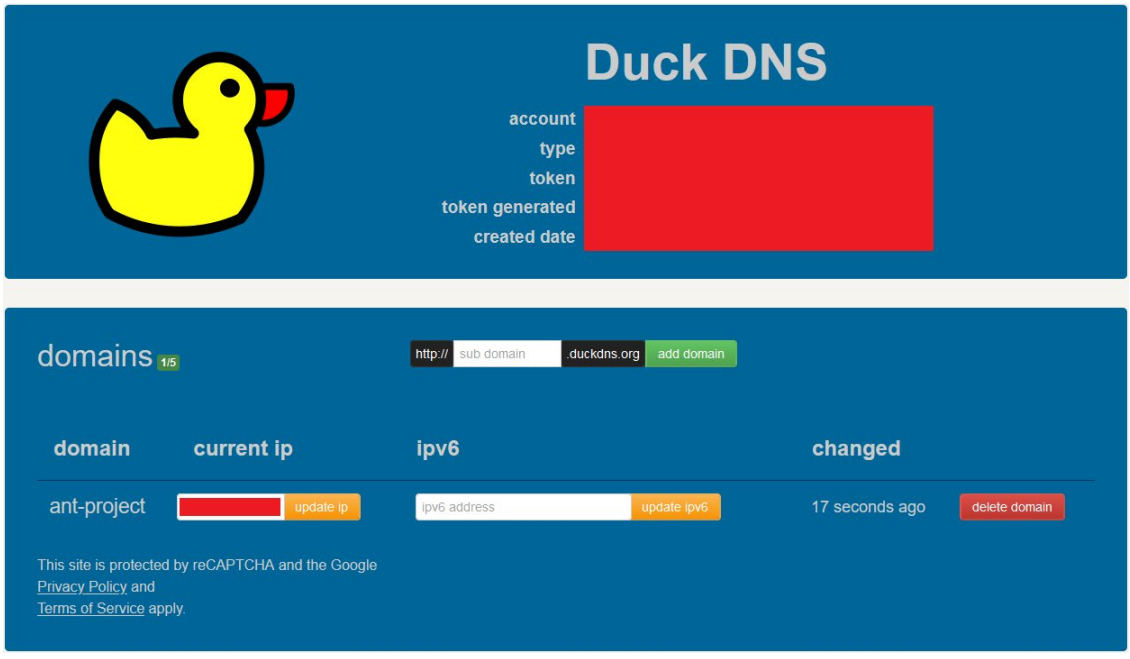

Рисунок 3.3 – Інтерфейс сервісу Duck DNS

Після виконаних дій можна переходити до встановлення самого сервера, а також подальшого його налаштування та тестування.

3.2 Встановлення Matrix Synapse

Synapse – це домашній сервер Matrix з відкритим вихідним кодом, який розроблявся з 2019 по 2023 рік як частина фонду Matrix.org Foundation [7]. Саме його ми будемо використовувати для нашого Matrix сервера. Процес встановлення зображений на рисунку 3.4, а усі команди для встановлення наведені у лістингу 3.1.

Лістинг 3.1 – Встановлення Matrix Synapse

```
$ sudo apt install -y lsb-release wget apt-transport-https
$ sudo wget -O \
  /usr/share/keyrings/matrix-org-archive-keyring.gpg \
  packages.matrix.org/debian/matrix-org-archive-keyring.gpg \
$ echo "deb \
  [signed-by=/usr/share/keyrings/matrix-org-archive-keyring.gpg] \
  https://packages.matrix.org/debian/ $(lsb_release -cs) main" | \
  sudo tee /etc/apt/sources.list.d/matrix-org.list
$ sudo apt update
$ sudo apt install matrix-synapse-py3
```

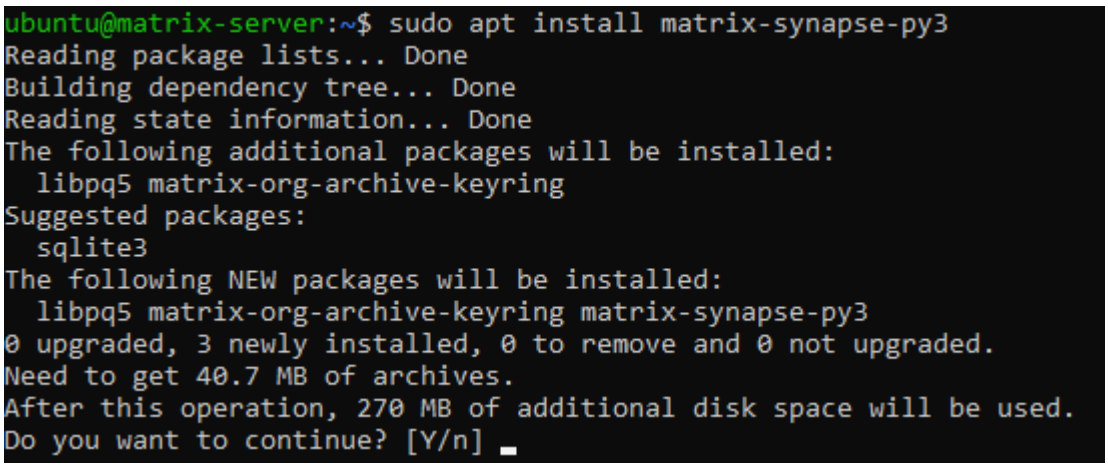

Рисунок 3.4 – Встановлення Matrix Synapse

Під час встановлення з’явиться вікно, у якому потрібно вказати доменне ім’я нашого серверу, яке ми отримали через сервіс Duck DNS (див. рисунок 3.5).

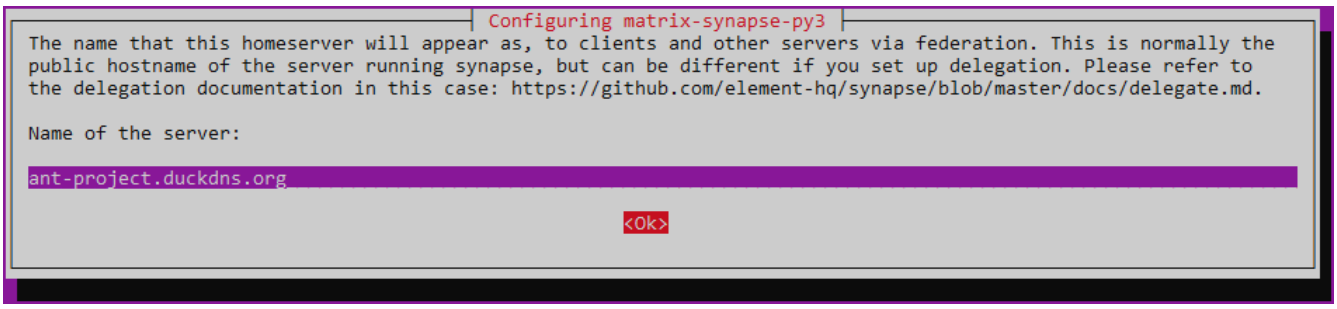

Рисунок 3.5 – Конфігурація Matrix Synapse

Також для роботи сервера потрібно встановити залежності, як це зображено на рисунку 3.6. Команда для встановлення:

```
$ sudo apt install build-essential python3-dev libffi-dev \
    python3-pip python3-setuptools sqlite3 \
    libssl-dev virtualenv libjpeg-dev libxslt1-dev libicu-dev
```

```
ubuntu@matrix-server:~$ sudo apt install build-essential python3-dev libffi-dev \
>                     python3-pip python3-setuptools sqlite3 \
>                     libssl-dev virtualenv libjpeg-dev libxslt1-dev libicu-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-setuptools is already the newest version (59.6.0-1.2ubuntu0.22.04.1).
The following additional packages will be installed:
  bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11 gcc-11-base
  icu-devtools javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6
  libatomic1 libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libdeflate0 libdpkg-perl libexpat1-dev
  libfakeroot libfile-fcntllock-perl libfontconfig1 libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjbig0
  libjpeg-turbo8 libjpeg-turbo8-dev libjpeg8 libjpeg8-dev libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0
  libmpc3 libnsl-dev libpython3-dev libpython3.10-dev libquadmath0 libstdc++-11-dev libtiff5 libtirpc-dev libtsan0
  libubsan1 libwebp7 libxml2-dev libxpm4 linux-libc-dev lto-disabled-list make manpages-dev python3-distlib
  python3-filelock python3-pip-whl python3-platformdirs python3-setuptools-whl python3-virtualenv python3-wheel
  python3-wheel-whl python3.10-dev rpcsvc-proto zlib1g-dev
Suggested packages:
  bzip2-doc cpp-doc gcc-11-locales debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf
  automake libtool flex bison gdb gcc-doc gcc-11-multilib apache2 | lighttpd | httpd glibc-doc bzr libgd-tools icu-doc
  libssl-doc libstdc++-11-doc pkg-config make-doc python2-pip-whl python2-setuptools-whl sqlite3-doc
The following NEW packages will be installed:
  build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11
  gcc-11-base icu-devtools javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libasan6 libatomic1 libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libdeflate0 libdpkg-perl
  libexpat1-dev libfakeroot libffi-dev libfile-fcntllock-perl libfontconfig1 libgcc-11-dev libgd3 libgomp1 libicu-dev
  libisl23 libitm1 libjbig0 libjpeg-dev libjpeg-turbo8 libjpeg-turbo8-dev libjpeg8 libjpeg8-dev libjs-jquery
  libjs-sphinxdoc libjs-underscore liblsan0 libmpc3 libnsl-dev libpython3-dev libpython3.10-dev libquadmath0
  libssl-dev libstdc++-11-dev libtiff5 libtirpc-dev libtsan0 libubsan1 libwebp7 libxml2-dev libxpm4 libxslt1-dev
  linux-libc-dev lto-disabled-list make manpages-dev python3-dev python3-distlib python3-filelock python3-pip
  python3-pip-whl python3-platformdirs python3-setuptools-whl python3-virtualenv python3-wheel python3-wheel-whl
  python3.10-dev rpcsvc-proto sqlite3 virtualenv zlib1g-dev
0 upgraded, 82 newly installed, 0 to remove and 0 not upgraded.
Need to get 90.5 MB of archives.
After this operation, 313 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Рисунок 3.6 – Встановлення залежностей

Після даного етапу сервер буде готовий до роботи, і його залишається тільки налаштувати для зручного використання та оптимальної роботи.

3.3 Встановлення PostgreSQL

Офіційний посібник з встановлення Matrix рекомендує нам використовувати Postgres замість SQLite в якості системи баз даних для подолання теперішніх та майбутніх проблем з продуктивністю [8]. Для встановлення Postgres потрібно виконати наступні команди:

```
$ sudo apt install libpq5 libpq-dev
$ sudo apt install postgresql postgresql-contrib
```

Далі потрібно створити користувача (див. рисунок 3.7) за допомогою наступних команд:

```
$ sudo -u postgres bash
$ createuser --pwprompt synapse_user
```

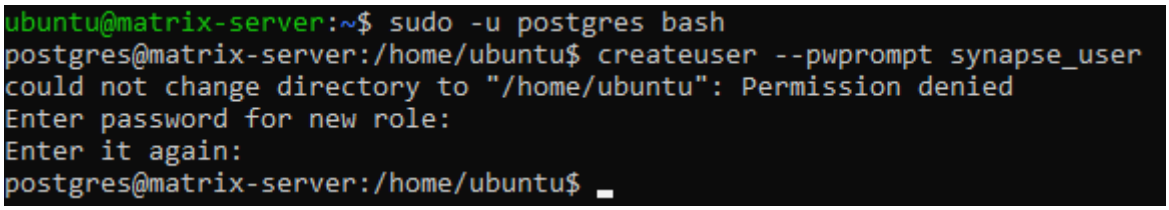

Рисунок 3.7 – Створення нового користувача в Postgres

Також потрібно створити нову базу даних за допомогою наступної команди:

```
$ createdb --encoding=UTF8 --locale=C -template=template0 \
    --owner=synapse_user synapse
```

Далі потрібно розкоментувати змінну у файлі /etc/postgresql/14/main/postgresql.conf (див. рисунок 3.8):

```
$ sudo nano /etc/postgresql/14/main/postgresql.conf
```

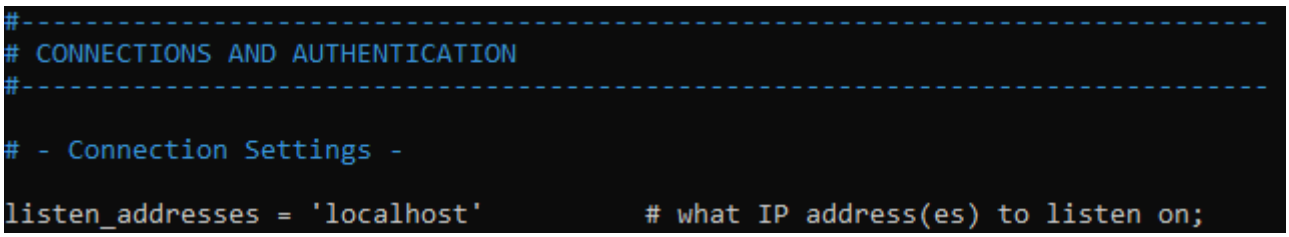

Рисунок 3.8 – Розкоментована змінна у файлі postgresql.conf

Включаємо автентифікацію по паролю, добавивши у файлі pg_hba.conf рядки, як зображено на рисунку 3.9:

```
$ sudo nano /etc/postgresql/14/main/pg_hba.conf
```

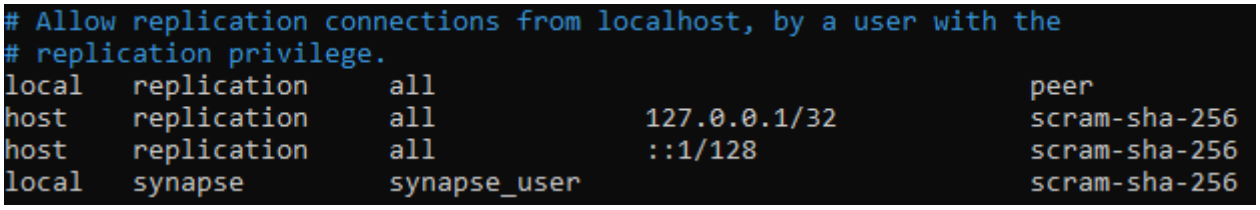

Рисунок 3.9 – Файл pg_hba.conf

Наступним кроком потрібно налаштувати підключення бази даних Postgres до Matrix Synapse. Для цього заходимо в налаштування сервера, бачимо стандартно підключену базу даних SQLite (див. рисунок 3.10):

```
$ sudo nano /etc/matrix-synapse/homeserver.yaml
```

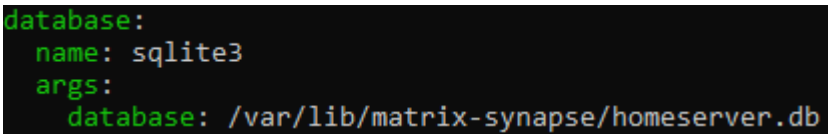

Рисунок 3.10 – Підключена база даних SQLite

Потрібно видалити дані рядки та ввести конфігурацію, наведену у лістингу 3.2 та зображену на рисунку 3.11.

Лістинг 3.2 – Налаштування підключення Postgres

```
database:
  name: psycopg2
  args:
    user: synapse_user
    password: <pass>
    database: synapse
    host: localhost
    cp_min: 5
    cp_max: 10
```

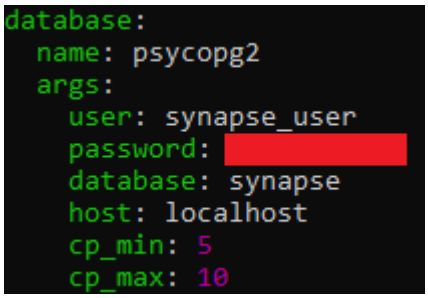

Рисунок 3.11 – Налаштування підключення Postgres

Після цього на нашому сервері для збереження даних буде використовуватись база даних PostgreSQL.

3.4 Налаштування proxy

Для простоти можна використовувати Caddy, веб-сервер з відкритим вихідним кодом, для налаштування нашого зворотного проксі. Спочатку встановлюємо Caddy за допомогою команд, наведених у лістингу 3.3.

Лістинг 3.3 – Налаштування підключення Postgres

```
$ sudo apt install -y debian-keyring debian-archive-keyring \
  apt-transport-https
$ curl -1sLf
'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' \
  | sudo gpg --dearmor \
  -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
$ curl -1sLf \
  'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' \
  | sudo tee /etc/apt/sources.list.d/caddy-stable.list
$ sudo apt update
$ sudo apt install caddy
```

Після встановлення Caddy видаляємо конфігурацію по замовчуванню:

```
$ sudo rm -rf /etc/caddy/Caddyfile
```

Створюємо новий конфігураційний файл та налаштовуємо його:

```
$ sudo touch /etc/caddy/Caddyfile
$ sudo nano /etc/caddy/Caddyfile
```

Потрібна конфігурація для сервера наведена у лістингу 3.4 та зображена на рисунку 3.12.

Лістинг 3.4 – Конфігурація Caddy

```
ant-project.ducksdns.org {
  reverse_proxy /_matrix/* localhost:8008
  reverse_proxy /_synapse/client/* localhost:8008
  reverse_proxy localhost:8008
}

ant-project.ducksdns.org:8448 {
  reverse_proxy localhost:8008
}
```

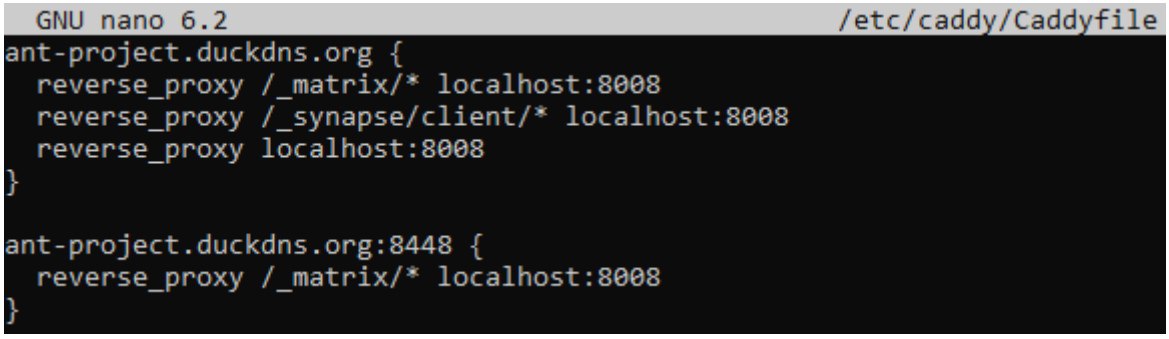

Рисунок 3.12 – Конфігурація Caddy

Після виконаннях всі перелічених дій наш сервер готовий до першого запуску, однак спочатку нам доведеться створити користувачів, використовуючи профілі яких ми будемо тестувати наш сервер.

3.5 Створення користувачів та перший запуск

Для створення користувачів, потрібно добавити паролю у поле registration_shared_secret у файлі /etc/matrix-synapse/homeserver.yaml, як це зображено на рисунку 3.13.

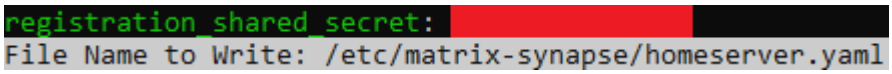

Рисунок 3.13 – registration_shared_secret у файлі homeserver.yaml

Далі створюємо нових користувачів:

```
$ register_new_matrix_user -u <user> \
    -c /etc/matrix-synapse/homeserver.yaml \
    -a https://ant-project.duckdns.org
```

Процес створення двох нових користувачів на сервері Matrix Synapse зображений на рисунку 3.14.

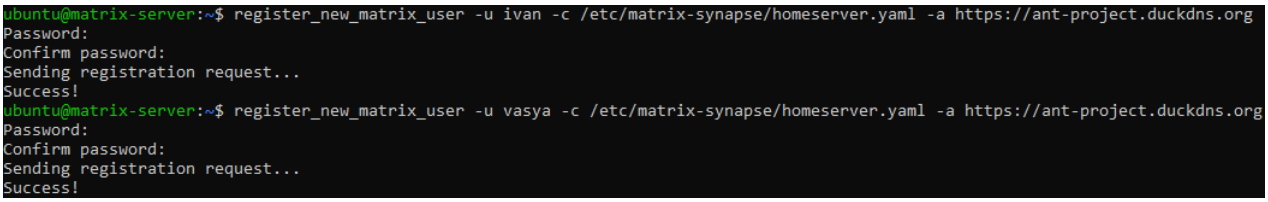

Рисунок 3.14 – Створення нових користувачів

Після цього потрібно перезапустити усі наші сервіси:

```
$ sudo systemctl restart caddy.service
$ sudo systemctl restart postgresql.service
$ sudo systemctl restart matrix-synapse
```

Після цього ми можемо перейти у веб-браузері по посиланню https://ant-project.duckdns.org, і побачимо сторінку, на якій нам повідомляється, що Matrix Synapse успішно запущено та працює (див. рисунок 3.15).

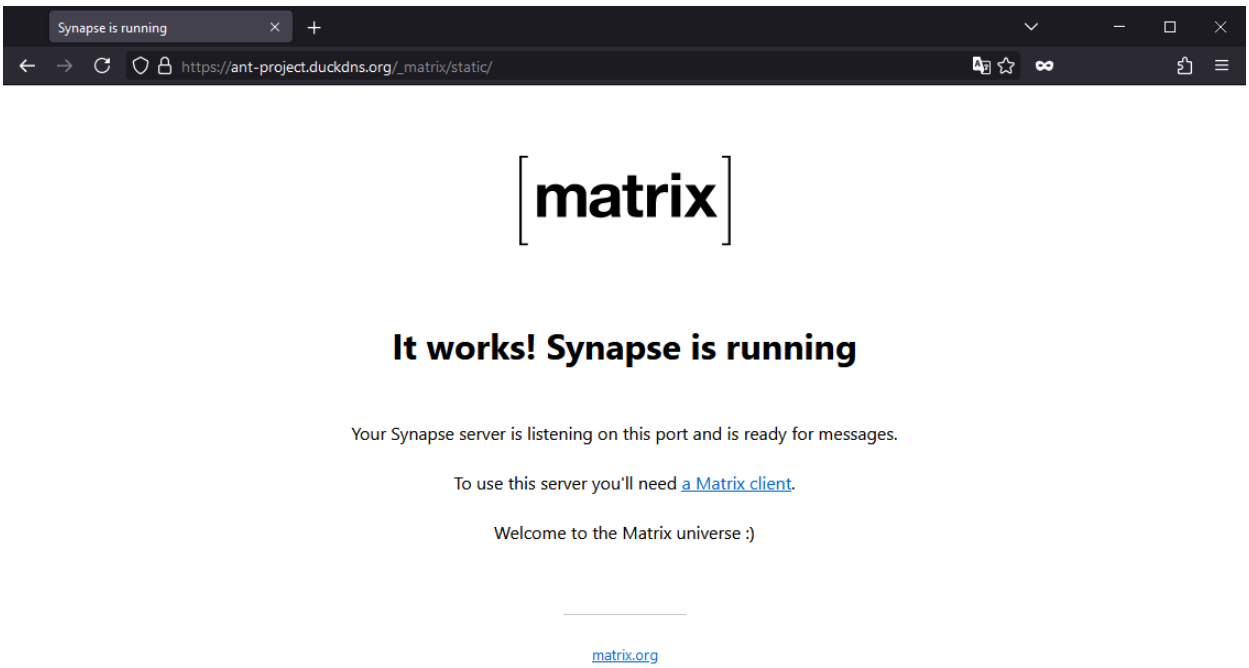

Рисунок 3.15 – Статична сторінка Matrix Synapse у веб-браузері

Після успішного тестового запуску можна перейти до повноцінного тестування роботи та функціональності нашого сервера.

3.6 Тестування роботи сервера

Для початку відкриємо веб-версію сервісу Element, який дозволяє підключитися до сервера через веб-браузер, незалежно на якому пристрої цей веб-браузер запущений, що робить використання у такий спосіб не залежним від платформи, на якому запущений цей сервіс. Підключимося до нашого сервера під користувачем ivan (див. рисунок 3.16).

Після успішного підключення до сервера, з’являється відповідний інтерфейс для комунікації через сервер, як це зображено на рисунку 3.17, на якому видно список усіх користувачів та кімнат, до яких користувач підключений та зразу надається можливість відправити повідомлення іншому користувачу, найти публічні каналі або створити свій груповий час.

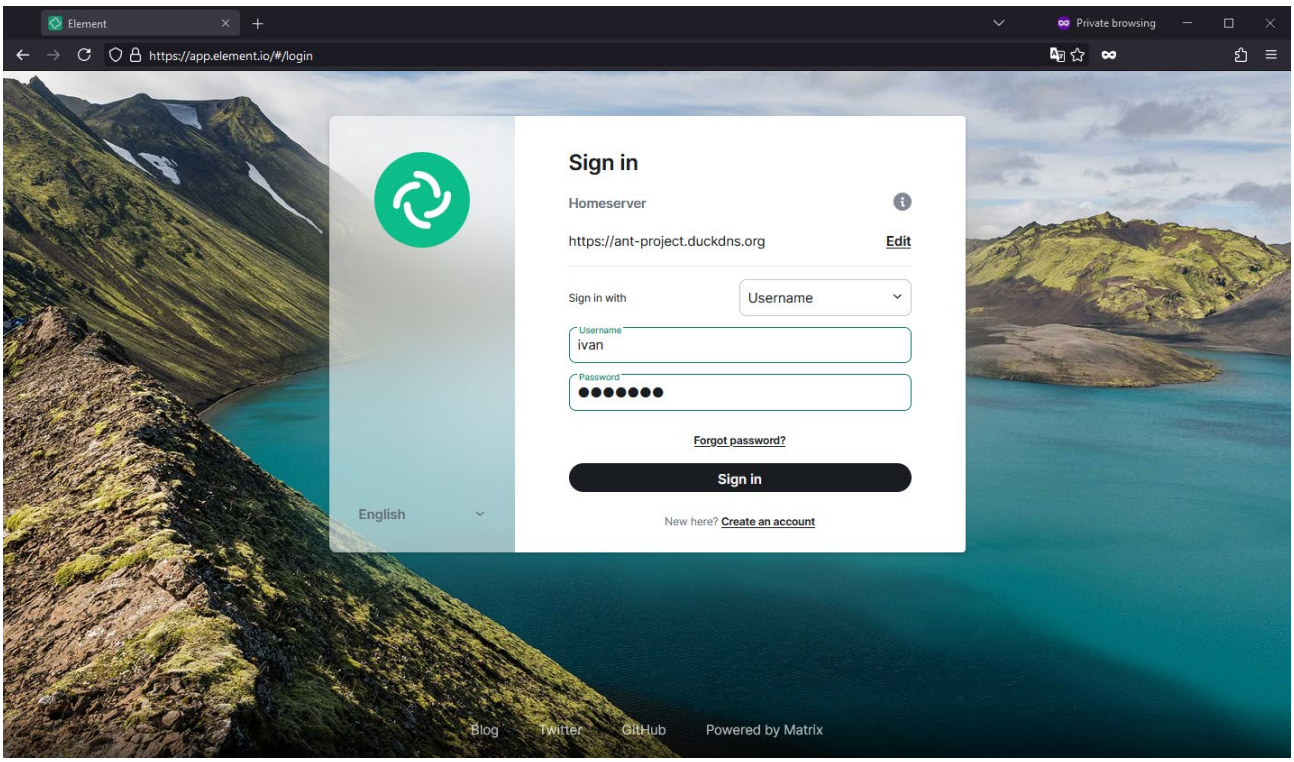

Рисунок 3.16 – Element Web Login

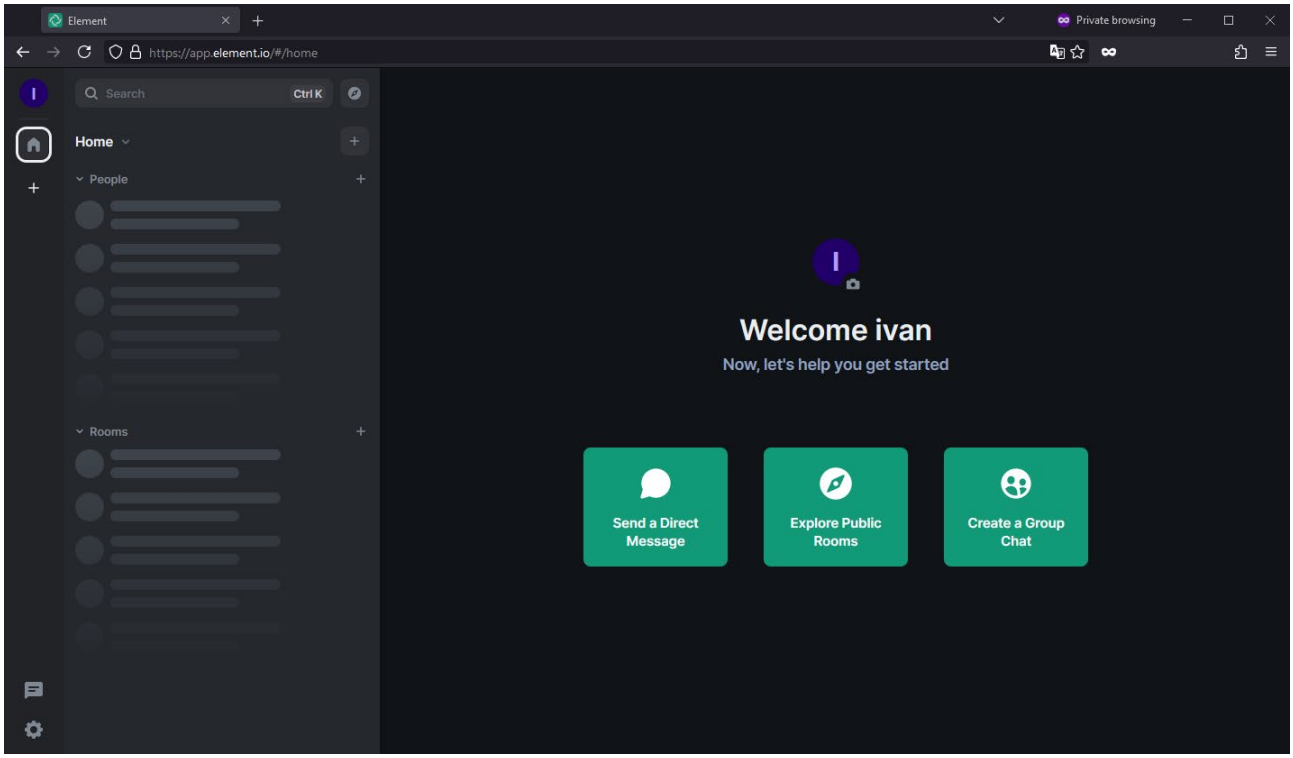

Рисунок 3.17 – Веб-інтерфейс Element підключеного до сервера користувача

З мобільного телефона спочатку підключимося через додаток FluffyChat, який аналогічно надає можливість підключитися до серверів на основні протоколу Matrix, під користувачем vasya (див. рисунок 3.18).

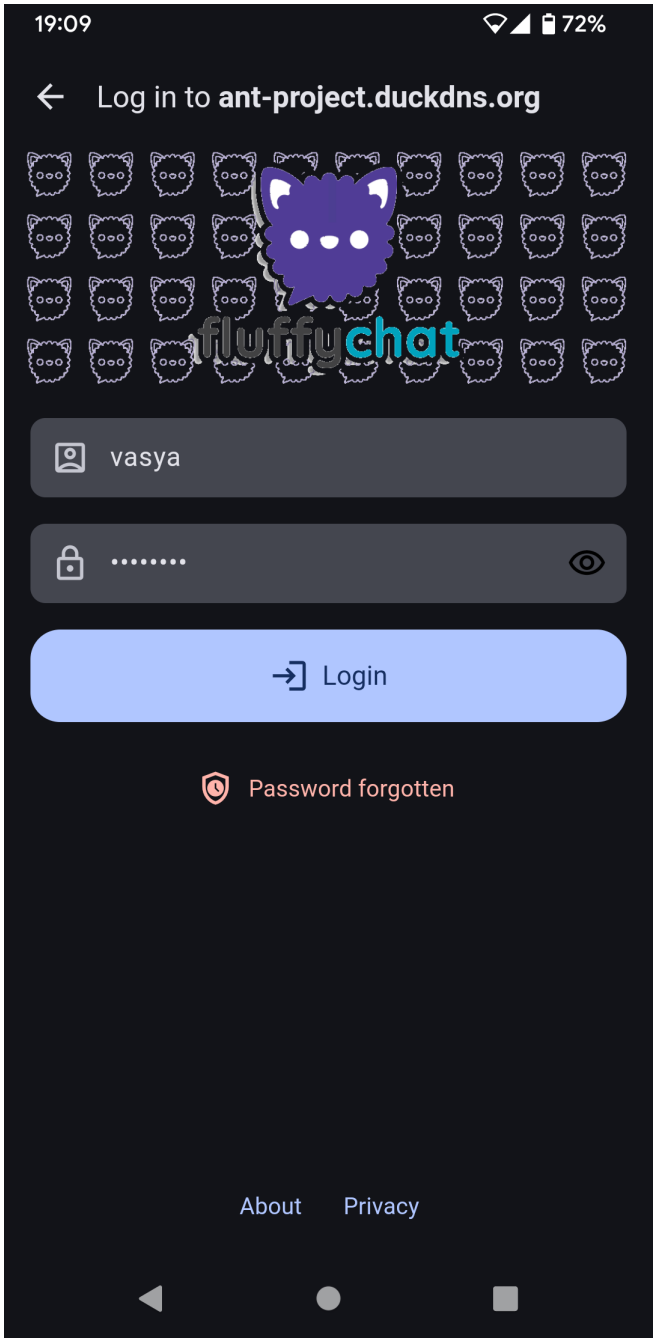

Рисунок 3.18 – FluffyChat Login

Після успішного підключення найдемо користувача ivan, який підключений до цього самого сервера (див. рисунок 3.19).

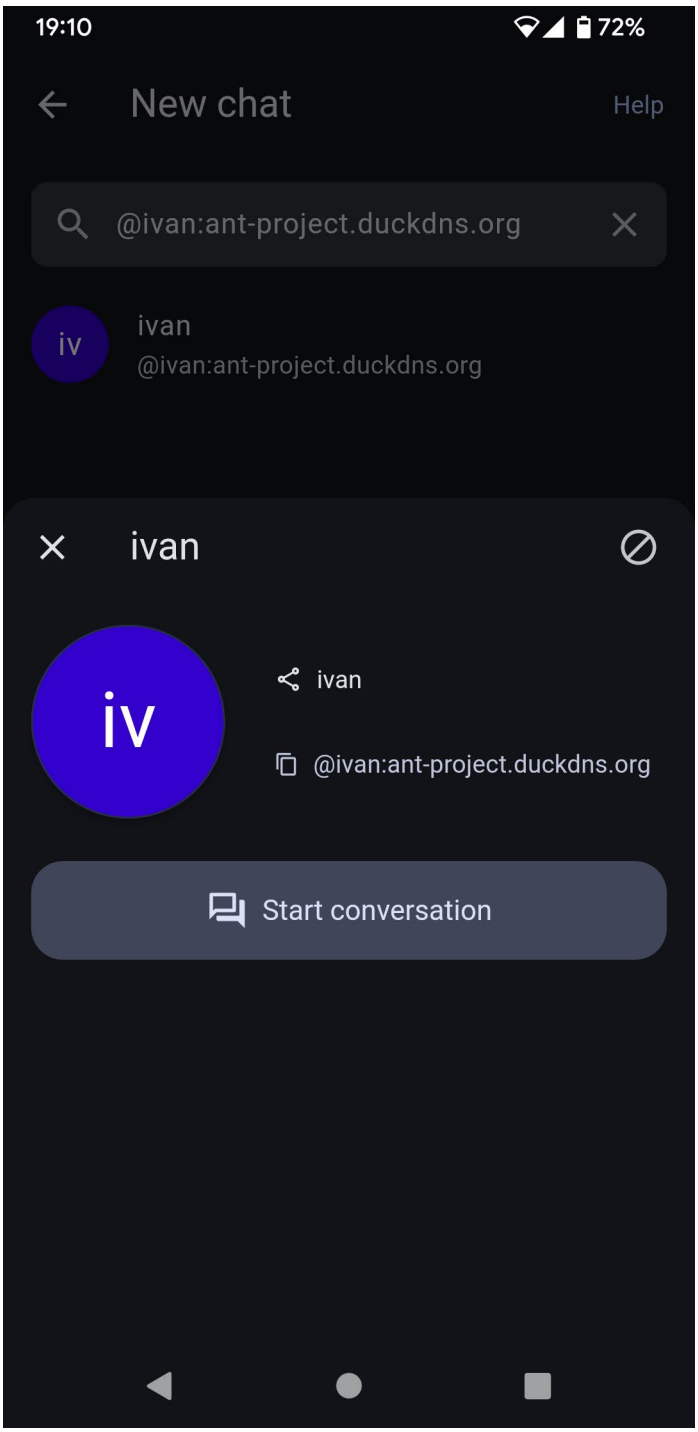

Рисунок 3.19 – Успішний пошук користувача ivan

Почнемо листування з користувачем ivan та напишемо йому перше повідомлення, як це зображено на рисунку 3.20. На цьому ж рисунку видно, що користувач ivan зараз активний, відображається вся інформація, яка відноситься до нашого листування та присутні такі можливості, як аудіодзвінок або відправлення аудіоповідомлення.

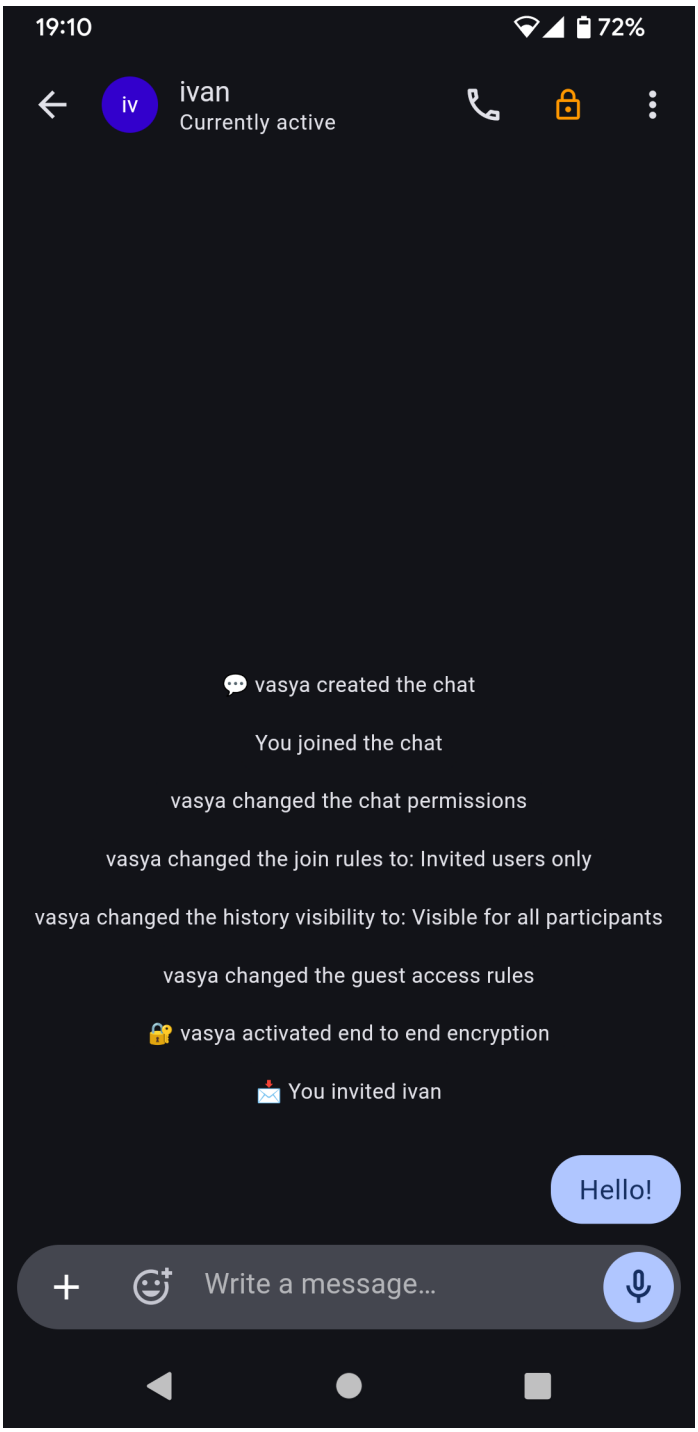

Рисунок 3.20 – Створене листування з першим повідомленням у FluffyChat

Після цього початкова сторінка інтерфейсу додатка буде виглядати, як і багато інших звичних інтерфейсів месенджерів (див. рисунок 3.21), що значно спрощує його використання та сприйняття. Також будуть доступні такі функції, як пошук користувачів та чатів, або створення нового листування.

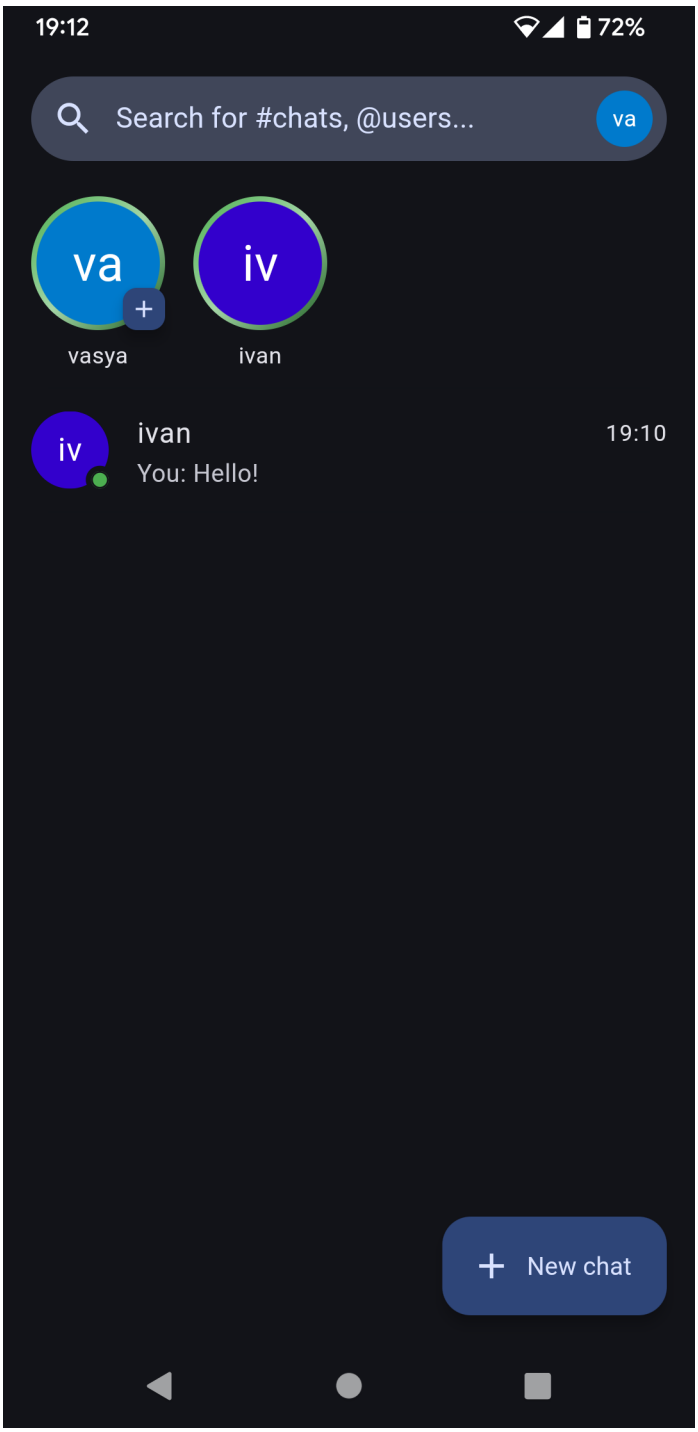

Рисунок 3.21 – Вигляд початкової сторінки додатка при запуску FluffyChat

Після успішного відправленого повідомлення у користувача ivan з’являється сповіщення (див. рисунок 3.22). Відкривши його, з’явиться інтерфейс листування з користувачем vasya (див. рисунок 3.23). У ньому буде відображена аналогічна інформація щодо поточного листування.

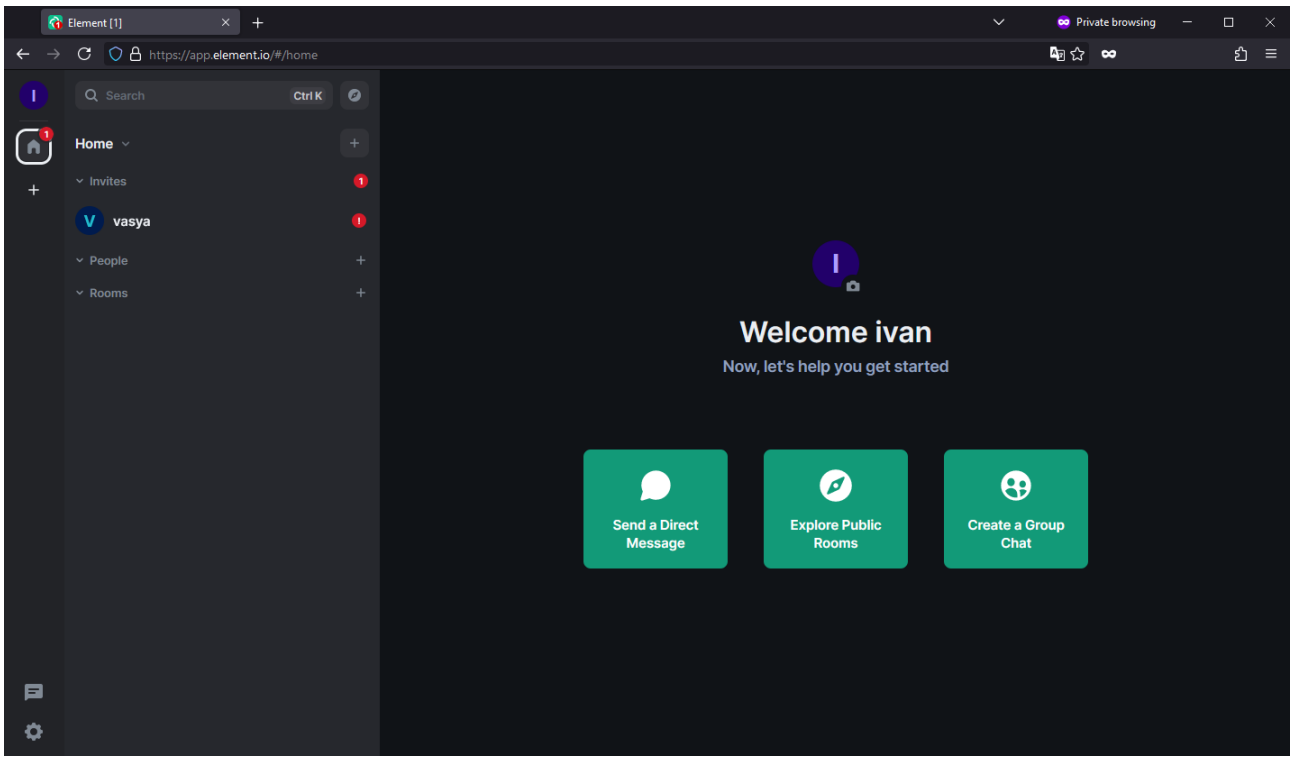

Рисунок 3.22 – Сповіщення у Element Web

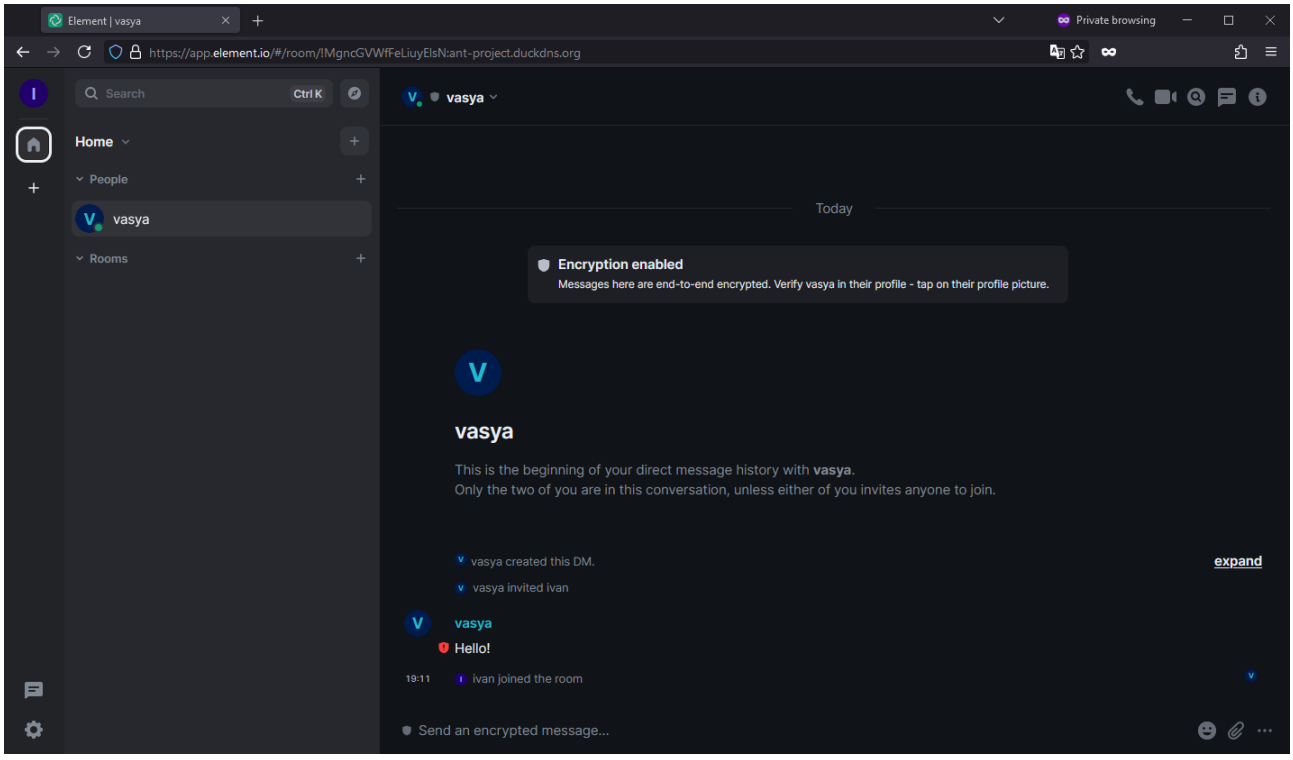

Рисунок 3.23 – Інтерфейс листування у Element Web

Використовувати можна і інші додатки, наприклад Element для Android. Спробуємо за допомогою нього підключитися до нашого сервера під тим самим користувачем (див. рисунок 3.24).

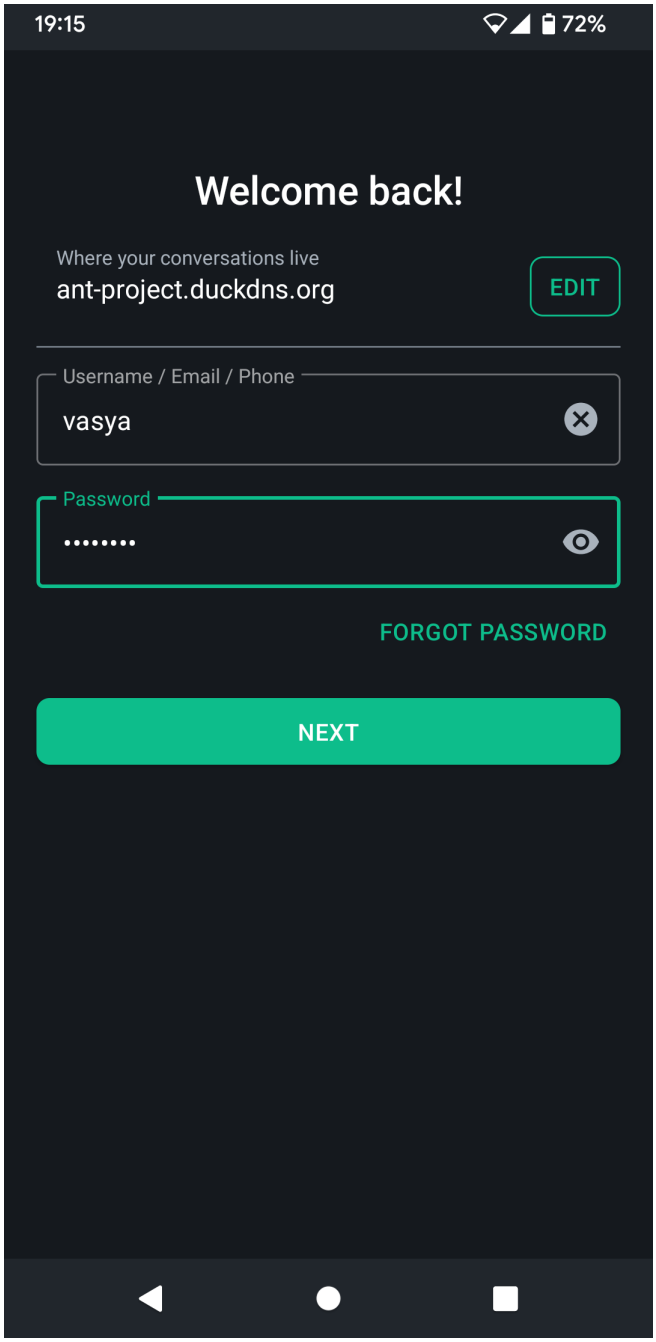

Рисунок 3.24 – Підключення через Element для Android

Відправляємо ще одне аналогічне повідомлення, інтерфейс початкової сторінки додатка зображено на рисунку 3.25.

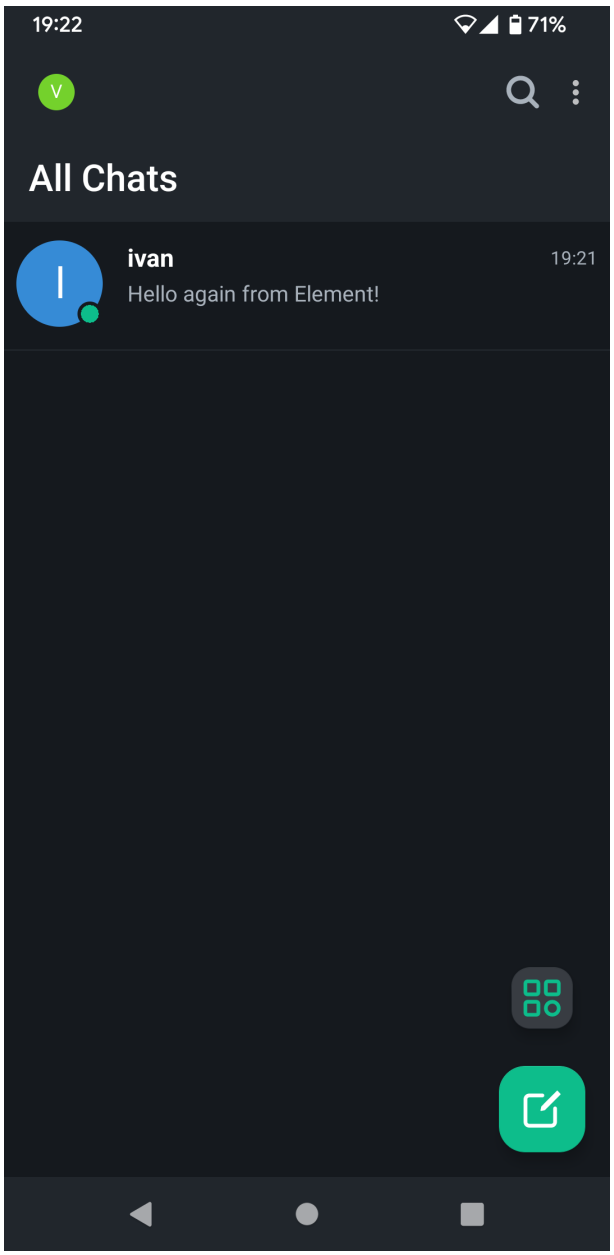

Рисунок 3.25 – Вигляд початкової сторінки при запуску Element для Android

Користувач ivan через веб-інтерфейс успішно бачить нове повідомлення (див. рисунок 3.26).

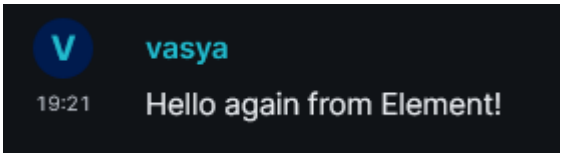

Рисунок 3.26 – Нове повідомлення у користувача ivan

Тепер спробуємо подзвонити. Коли користувач ivan починає дзвінок, у користувача vasya з’являється відповідне сповіщення, як це зображено на рисунку 3.27.

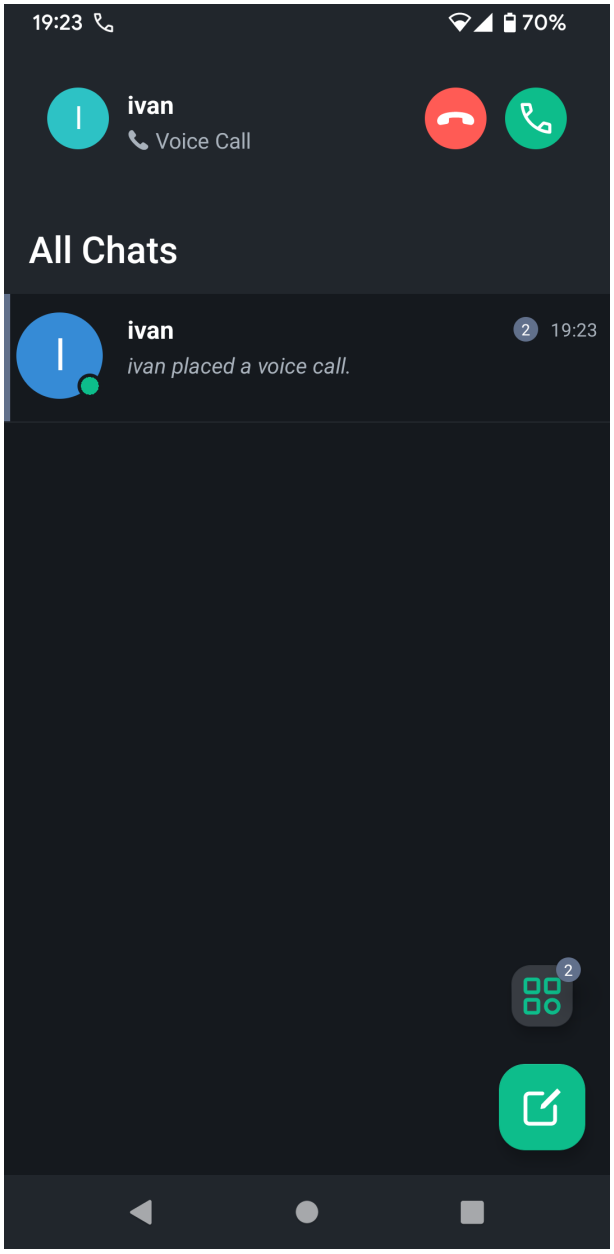

Рисунок 3.27 – Сповіщення вхідного дзвінка у інтерфейсі Element для Android

Інтерфейс активного дзвінка у додатку Element для Anrdoid зображено на рисунку 3.28.

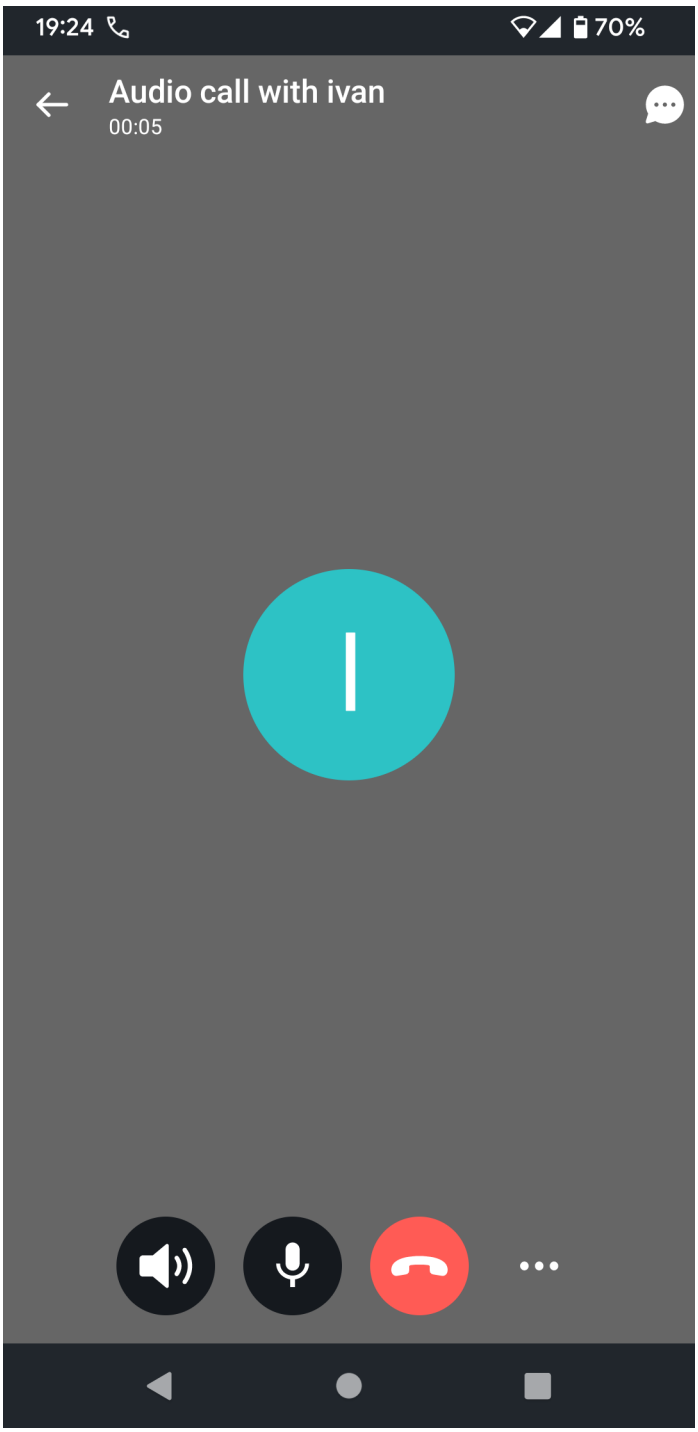

Рисунок 3.28 – Інтерфейс дзвінка у Element для Android

При цьому інтерфейс у користувача ivan, який підключений через веб-інтерфейс Element зображено на рисунку 3.29.

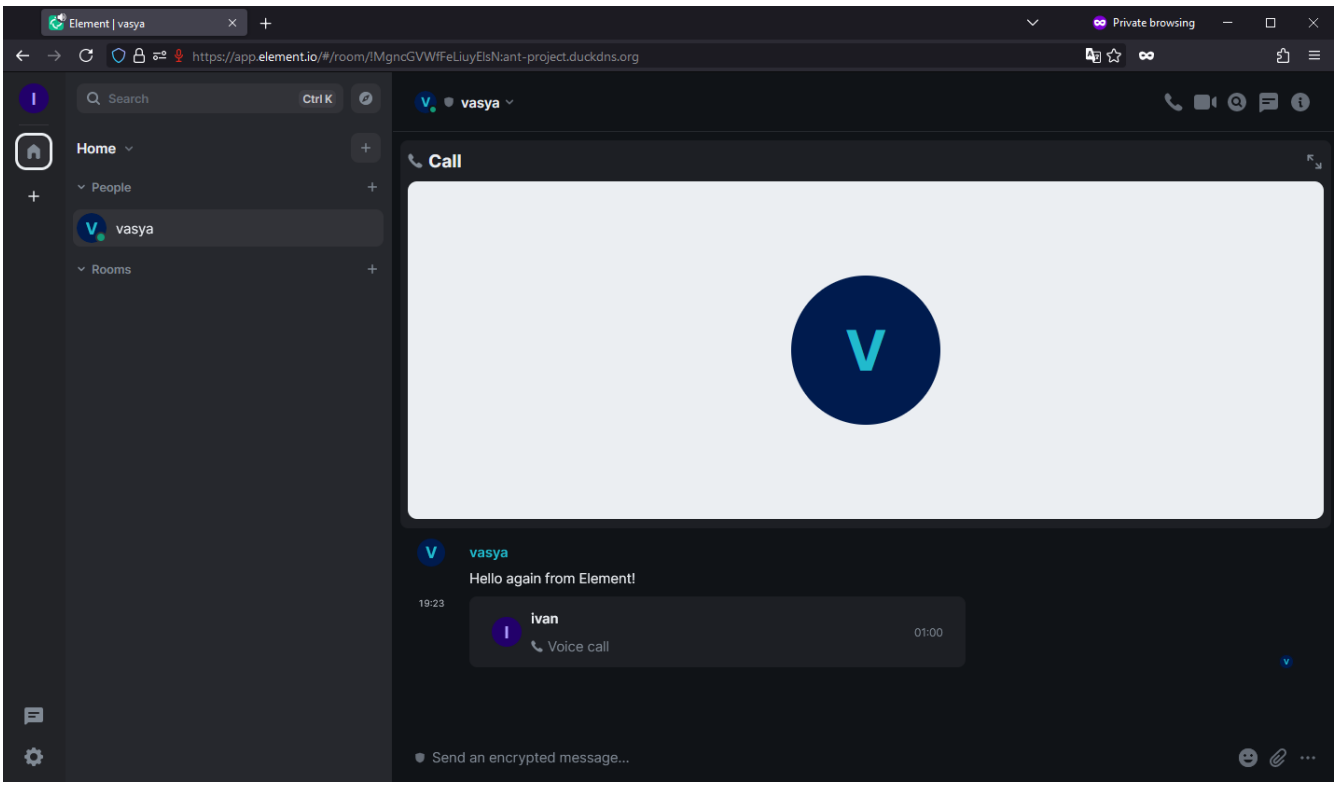

Рисунок 3.29 – Інтерфейс дзвінка у веб-інтерфейсі Element

Варто підмітити, що якість зв’язку є хорошою, немає посторонніх шумів чи значної затримки. Даний месенджер працює стабільно, тобто він є готовим для використання у повсякденному житті чи у корпоративній сфері, при цьому має значні переваги у порівнянні з іншими месенджерами, про які було розказано раніше.

Практичне налаштування сервера показало, що інсталяція та конфігурація були відносно простими і добре задокументованими, що сприяє швидкому впровадженню у будь-якому середовищі.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Значення адаптації в трудовому процесі

Праця людини безпосередньо пов’язана із виробничим середовищем. Працівник може нормально здійснювати трудову діяльність лише тоді, коли умови зовнішнього середовища відповідають оптимальним. Якщо вони змінюються, стають несприятливими, то на протидію їм організм людини включає спеціальний механізм, який зберігає постійність внутрішньою середовища, або змінює його в межах допустимого. Такий механізм називається адаптацією. Адаптація є важливим засобом попередження травмування, виникнення нещасних випадків у трудовому процесі і відіграє значну роль в охороні праці.

Адаптація – це динамічний процес пристосування організму та його органів до мінливих умов зовнішнього середовища.

Адаптація в трудовій діяльності поділяється на фізіологічну, психічну, соціальну та професійну.

Фізіологічна адаптація – це сукупність фізіологічних реакцій, які є в основі пристосування організму до змін зовнішніх умов, і направлені на збереження відносної постійності його внутрішнього середовища – гомеостазу.

Гомеостаз – це відносна динамічна постійність складу та властивостей внутрішнього середовища і стійкість основних фізіологічних функцій організму людини. Гомеостаз в організмі підтримується на усіх рівнях його організації і забезпечує динамічну рівновагу організму і зовнішнього середовища.

Суть механізму адаптації полягає у змінах меж чутливості аналізаторів, розширенні діапазону фізіологічних резервів організму та зміні в певних межах параметрів фізіологічних функцій. Завдяки фізіологічній адаптації фізичні та біохімічні параметри, які визначають життєдіяльність організму, змінюються у вузьких межах порівняно із значними змінами зовнішніх умов: підвищується стійкість організму до холоду, тепла, недостачі кисню, змін барометричного

тиску та інших факторів. Велике значення у фізіологічній адаптації має реактивність організму, його початковий функціональний стан (вік, тренованість тощо), в залежності від якого змінюються і відповідні реакції організму на різні дії. Процес фізіологічної адаптації до незвичайних, екстремальних умов проходить декілька стадій, або фаз: спочатку переважають явища декомпенсації (порушення функцій), потім неповного пристосування (активний пошук організмом стійких станів, що відповідають новим умовам середовища) і, нарешті, фаза відносного стійкого пристосування.

Фізіологічна адаптація до праці має активний характер і за сприятливих умов виробничого середовища та оптимальних навантажень веде до підвищення стійкості та працездатності організму, збільшення його резервних можливостей, зменшення захворювань і травматизму. Проте коливання умов середовища, в яких відбувається фізіологічна адаптація, має певну межу, характерну для кожного організму. Якщо працівник потрапляє в умови, коли інтенсивність впливу чинників виробничого середовища переважає можливості його адаптації, настають патологічні зміни фізіологічних систем, захворювання організму.

Психічна адаптація – це процес встановлення оптимальної відповідності особистості до навколишнього середовища в процесі діяльності. Зрозуміло, що такі властивості, як гальмування мислення та низька швидкість переробки інформації, обмежений діапазон сприйняття, порушення функції пам’яті гальмують адаптацію; висока рухливість нервових процесів, навпаки, її підвищує.

Психічна адаптація в процесі праці залежить від психічних властивостей працівника, його психічного стану, психологічних реакцій на стреси, що виникають на роботі, кваліфікації та культури людини, особливостей професійної діяльності, конкретних умов праці тощо.

Соціальна адаптація – це пристосування працюючої людини до системи відносин у робочому колективі з його нормами, правилами, традиціями, ціннісними орієнтаціями. Під час соціальної адаптації працівник поступово отримує різнобічну інформацію про колектив, де він працює, про систему ділових та особистих взаємовідносин.

При несприятливому протіканні соціальної адаптації підвищується рівень стресу на роботі, наслідки якого позначаються на поведінці працівника та можуть призвести до міжособових конфліктів, нещасних випадків.

Професійна адаптація – це адаптація до трудової діяльності з усіма її складовими: адаптація до робочого місця, знарядь та засобів праці, об'єктів та предметів праці, особливостей технологічного процесу, часових параметрів роботи тощо.

Професійна адаптація виражається у розвитку стійкого позитивного ставлення працівника до своєї професії, певного рівня оволодіння ним специфічними навичками та уміннями, у формуванні необхідних для якісного виконання роботи властивостей. Професійна адаптація визначається необхідним мінімумом знань та навичок, яких працівник набув при одержанні спеціальності, ступенем відповідальності, практичністю, діловитістю тощо. Адаптація вважається завершеною тоді, коли працівник досягає кваліфікації, відповідної існуючим стандартам.

Кожен із розглянутих видів адаптації впливає на працездатність та здоров'я працівника, формує у нього певний рівень чутливості та стійкості до психоемоційних перевантажень, внаслідок розвитку яких може істотно змінитися надійність професійної діяльності.

4.2 Вимоги ергономіки до організації робочого місця оператора ПК

Робоче місце – це зона простору, що оснащена необхідним устаткуванням, де відбувається трудова діяльність одного працівника чи групи працівників.

Раціональне планування робочого місця має забезпечувати: найкраще розміщення знарядь і предметів праці, не допускати загального дискомфорту, зменшувати втомлюваність працівника, підвищувати його продуктивність праці. Площа робочого місця має бути такою, щоб працівник не робив зайвих рухів і не відчував незручності під час виконання роботи. Важливо мати також можливість змінити робочу позу, тобто положення корпуса, рук, ніг. Проте доцільно

виключати або мінімізувати всі фізіологічно неприродні і незручні положення тіла.

Проведені дослідження показують, що при раціональній організації робочих місць продуктивність праці зростає знати на 15-25%.

Гігієнічні вимоги визначають умови життєдіяльності і працездатності людини у процесі взаємодії з технікою і середовищем; показниками є рівень освітлення, температура, вологість, шум, вібрація, токсичність, загазованість тощо.

Антропометричні вимоги визначають відповідність конструкцій техніки антропометричним характеристикам людини (зріст, розміри тіла та окремі рухові ланки). Показниками є раціональна робоча поза, оптимальні зони досягнення, раціональні трудові рухи.

Фізіологічні та психофізіологічні вимоги визначають відповідність техніки і середовища можливостям працівника щодо сприйняття, переробки інформації, прийняття і реалізації рішень.

Організація робочого місця передбачає:

– правильне розміщення робочого місця у виробничому приміщенні;

– вибір ергономічно обґрунтованого робочого положення, виробничих меблів з урахуванням антропометричних характеристик людини;

– раціональне компонування обладнання на робочих місцях;

– урахування характеру та особливостей трудової діяльності.

Загальні принципи організації робочого місця:

– на робочому місці не повинно бути нічого зайвого. Усі необхідні для роботи предмети мають бути поряд із працівником, але не заважати йому;

– ті предмети, якими користуються частіше, розташовуються ближче, ніж ті предмети, якими користуються рідше;

– предмети, які беруть лівою рукою, повинні бути зліва, а ті предмети, які беруть правою рукою – справа;

– якщо використовують обидві руки, то місце розташування пристосувань вибирається з урахуванням зручності захоплювання його двома руками;

– робоче місце не повинно бути захаращене;

– організація робочого місця повинна забезпечувати необхідну оглядовість.

Статичні напруження працівника в процесі праці пов’язані з підтриманням у нерухомому стані предметів і знарядь праці, а також підтриманням робочої пози.

Робоча поза – це основне положення працівника у просторі: зручна робоча поза має забезпечувати стійкість положення корпуса, ніг, рук, голови працівника під час роботи, мінімальні затрати енергії та максимальну результативність праці.

Найпоширенішими у процесі праці є пози сидячи і стоячи. Проектуючи робоче місце, потрібно враховувати, що при виконанні роботи з фізичним навантаженням бажана поза стоячи, а при малих зусиллях – сидячи.

Робоча поза стоячи втомлює людину більше, ніж сидяча. Вона вимагає на 10 % більше енергії, спричиняє підвищення артеріального і венозного тиску крові, розширення вен на ногах, пошкодження ступень, викривлення хребта.

Під час роботи сидячи нижня частина корпуса розслаблена, а основне статичне навантаження припадає на м’язи шиї, спини, таза, стегон. Неправильна сидяча поза може викликати застій крові в ногах, а якщо виконується великий обсяг роботи для пальців рук – запалення суглобів.

ВИСНОВКИ

В процесі виконання цієї кваліфікаційної роботи було успішно реалізовано та налаштовано сервер федеративного месенджера на основі протоколу Matrix. Результати роботи підтверджують, що протокол Matrix є перспективним рішенням для створення децентралізованих комунікаційних мереж, які відповідають високим вимогам безпеки та надійності.

Під час дослідження було проведено детальний аналіз сучасних підходів до побудови месенджерів. Виявлено, що централізовані месенджери мають переваги у простоті управління, але суттєво поступаються децентралізованим рішенням у питаннях безпеки та конфіденційності.

Теоретична частина роботи включала огляд та аналіз принципів роботи протоколу Matrix, його архітектури та особливостей федерації. Визначено, що протокол Matrix дозволяє створювати мережі, де кожен сервер може взаємодіяти з іншими, забезпечуючи при цьому високу ступінь конфіденційності та захисту даних.

У практичній частини було налаштовано сервер на основі Matrix Synapse, використано базу даних PostgreSQL та налаштовано проксі-сервер для забезпечення додаткового рівня безпеки. Тестування підтвердило стабільну роботу сервера, високий рівень якості зв’язку, відсутність сторонніх шумів та значних затримок.

Завершення даного проекту демонструє практичну цінність та можливості протоколу Matrix у створенні безпечних та надійних комунікаційних мереж. Отримані результати можуть бути корисними для фахівців у галузі кібербезпеки, а також для організацій, які прагнуть підвищити рівень захисту своїх комунікаційних систем. У майбутньому планується продовжити дослідження у напрямку оптимізації та розширення функціональності федеративних месенджерів на основі протоколу Matrix, що дозволить ще більше підвищити їх безпеку та надійність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mesh World P2P Simulation Hypothesis [Електронний ресурс] – Режим доступу до ресурсу: https://www.delphitools.info/DWSH/. Дата доступу: 01.06.2024.

2. Matrix // DEVPEW [Електронний ресурс] – Режим доступу до ресурсу: https://devpew.com/blog/matrix/. Дата доступу: 01.06.2024.

3. Букатка, С., & Тимощук, В. (2023). ХЕШ-алгоритм шифрування паролів користувачів ос Linux. Матеріали VI Міжнародної студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання “, 112-113.

4. What is end-to-end encryption (E2EE)? [Електронний ресурс] – Режим доступу до ресурсу: https://atsign.com/resources/articles/what-is-end-to-end-encryption-e2ee/. Дата доступу: 01.06.2024.

5. Тимощук, В., & Стебельський, М. (2023). Шифрування даних в операційних системах. Матеріали VI Міжнародної студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання “, 183-184.

6. IRC [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/IRC. Дата доступу: 01.06.2024.

7. Karnaukhov, A., Tymoshchuk, V., Orlovska, A., & Tymoshchuk, D. (2024). USE OF AUTHENTICATED AES-GCM ENCRYPTION IN VPN. Матеріали конференцій МЦНД, (14.06. 2024; Суми, Україна), 191-193. https://doi.org/10.62731/mcnd-14.06.2024.004

8. XMPP [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/XMPP. Дата доступу: 01.06.2024.

9. Nataliya Zagorodna, Iryna Kramar (2020). Economics, Business and Security: Review of Relations. Business Risk in Changing Dynamics of Global Village BRCDGV-2020: Monograph / Edited by Pradeep Kumar, Mahammad Sharif. India, Patna: Novelty & Co., Ashok Rajpath,. 446 p., pp.25-39.

10. Signal [Електронний ресурс] – Режим доступу до ресурсу: https://signal.org/. Дата доступу: 01.06.2024.

11. Tymoshchuk, V., Dolinskyi, A., & Tymoshchuk, D. (2024). MESSENGER BOTS IN SMART HOMES: COGNITIVE AGENTS AT THE FOREFRONT OF THE INTEGRATION OF CYBER-PHYSICAL SYSTEMS AND THE INTERNET OF THINGS. Матеріали конференцій МЦНД, (07.06. 2024; Луцьк, Україна), 266-267. https://doi.org/10.62731/mcnd-07.06.2024.004

12. Synapse: Matrix homeserver written in Python/Twisted. [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/matrix-org/synapse. Дата доступу: 01.06.2024.

13. Setup Matrix Synapse Home-server [Електронний ресурс] – Режим доступу до ресурсу: https://medium.com/@dassomnath/setup-matrix-synapse-home-server-ba54e20f8290. Дата доступу: 01.06.2024.