

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ

ЗВІТ
до практичного завдання №2
з дисципліни «Архітектура програмного забезпечення»

Тема: Архітектура програмної системи Telegram

Виконав: студент групи ПЗПІ-22-2
Д'яченко Микита

Харків – 2025

1 ІСТОРІЯ ЗМІН

№	Дата	Версія звіту	Опис змін та виправлень
1	14.06.2025	0.1	Створено звіт на основі презентації Telegram

2 ЗАВДАННЯ

Підготувати доповідь про архітектуру однієї з відомих складних програмних систем. Оформити презентацію, зняти відеозапис, написати текстовий і PDF-звіт згідно з вимогами ДСТУ 3008:2015.

3 ОПИС ВИКОНАНОЇ РОБОТИ

У межах практичного завдання №2 було обрано систему Telegram як приклад складної, розподіленої програмної системи. Було досліджено її архітектуру, включаючи протокол MTProto, клієнт-серверну модель, роботу дата-центрів, кешування, зберігання медіа через CDN, а також API для ботів. Зроблено акцент на її мікросервісну структуру, масштабованість і засоби безпеки. Створено презентацію з 10 слайдів, яка містить ключові технічні характеристики системи. У доповіді охоплено як архітектурні, так і технологічні аспекти: реалізація черг, кешування, обробка подій, захист від DDoS-атак, використання баз даних і засобів CI/CD. Telegram показано як сучасну, стабільну та відкриту до інтеграцій систему.

4 ВИСНОВКИ

Telegram — це один із найкращих прикладів високонавантаженої хмарної системи. Її архітектура побудована на принципах модульності, безпеки та масштабованості. Система здатна обслуговувати сотні мільйонів користувачів по всьому світі, використовуючи георозподілену інфраструктуру, оптимізовані внутрішні протоколи та сучасні технології. Завдяки відкритому API Telegram також служить платформою для сторонніх розробників і ботів, що робить її ще більш універсальною. Отримані знання про архітектуру Telegram можуть бути корисними для аналізу інших сучасних складних систем.

ДОДАТОК Б

Відеозапис доповіді на YouTube

<https://www.youtube.com/watch?v=AILEioig91sss>

- 00:00 Вступ, загальна характеристика Telegram
- 01:00 Протокол MTProto та модель клієнт-сервер
- 02:15 Основні компоненти системи
- 03:20 Обробка повідомлень і кешування
- 04:30 Хмарне зберігання та CDN
- 05:15 API для ботів
- 06:10 Безпека, масштабування
- 07:10 Технології Telegram
- 08:00 Висновки

ДОДАТОК Б

Слайди презентації



Архітектура програмної системи Telegram

ПЗПІ-22-2 Д'яченко Микита 2025 рік

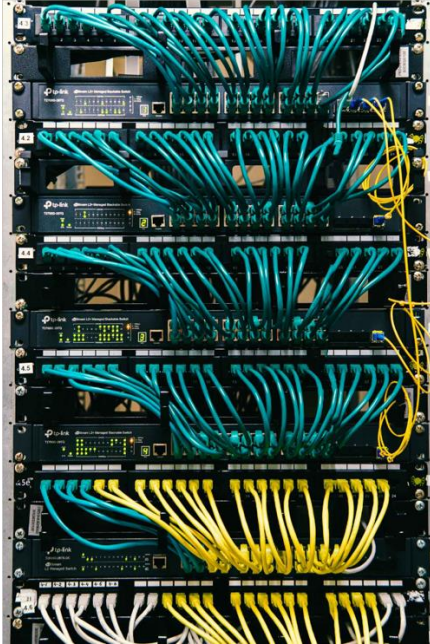
Рисунок Б.1 – Титульний слайд

Загальна характеристика Telegram

Telegram — це масштабований, безпечний, багатофункціональний хмарний месенджер, запущений у 2013 році Павлом Дуровим. Система підтримує кросплатформену синхронізацію, канали, групи, боти, відеодзвінки та хмарне зберігання медіа.

Архітектура Telegram побудована з урахуванням потреб сотень мільйонів користувачів у всьому світі, і базується на власному протоколі MTProto, що забезпечує високу швидкість передачі даних та шифрування. Telegram сьогодні — це не лише додаток, а розподілена глобальна система з власною CDN, георозподіленими дата-центрами, API для сторонніх розробників і внутрішньою мікросервісною інфраструктурою.

Рисунок Б.2 – Загальна характеристика Telegram



Клієнт-серверна архітектура та протокол MTProto

1

Клієнт-серверна модель

Telegram реалізує клієнт-серверну модель: клієнти ініціюють з'єднання, а всі дії — авторизація, передача повідомлень, завантаження файлів — обробляються серверною інфраструктурою.

2

Протокол MTProto

Для передачі даних використовується власний протокол MTProto, який забезпечує ефективне стиснення, шифрування та перевірку цілісності.

3

Шифрування повідомлень

Повідомлення шифруються на боці клієнта й надсилаються до сервера, який виконує маршрутизацію. У секретних чатах використовується end-to-end шифрування, яке не дозволяє серверам бачити зміст повідомлень. Це забезпечує баланс між безпекою і можливістю хмарного зберігання та синхронізації.

Рисунок Б.3 – Клієнт-серверна архітектура

Основні компоненти системи

Клієнтські застосунки

Мобільні, десктопні, веб.

Дата-центри (DC1–DC5)

Кожен дата-центр обслуговує частину користувачів, що дозволяє географічно розподіляти навантаження.

Сервери API

Відповідають за взаємодію клієнта з ядром системи, обробку повідомлень, автентифікацію та передачу медіа до сховища.

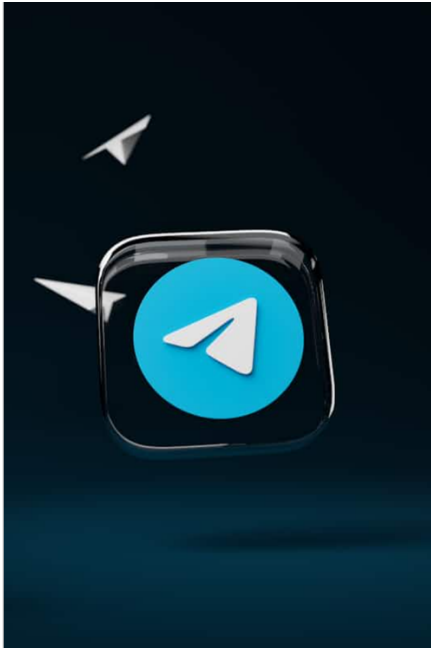
Сервери зберігання та кеші

Система кешування (на основі Redis) дозволяє пришвидшити доставку повторюваних даних.

Черги повідомлень і шлюзи для ботів

Забезпечують ефективну обробку та маршрутизацію даних.

Рисунок Б.4 – Основні компоненти



Механізм обробки повідомлень

Шифрування та передача

Спочатку клієнт шифрує повідомлення та передає його на сервер.

Отримання та класифікація

Сервер отримує дані, розшифровує (у межах дозволеного), класифікує за чергою подій і зберігає в базі або тимчасовому кеші.

Доставка

Якщо одержувач онлайн — повідомлення доставляється негайно, якщо ні — очікує підтвердження з іншого пристрою.

Подієво-орієнтована архітектура

Telegram використовує подієво-орієнтовану архітектуру з чергами (зокрема Redis або власні брокери), що забезпечує асинхронну доставку та високу продуктивність при великій кількості підключень.

Рисунок Б.5 — Обробка повідомлень



Хмарне зберігання та CDN



Хмарне зберігання

У Telegram реалізовано хмарне зберігання всіх мультимедійних файлів.



Розподіл по CDN

При завантаженні відео чи документа, дані розподіляються по спеціальних серверах CDN, що фізично наближені до користувачів.



Прискорення завантаження

Це дозволяє прискорити завантаження, уникнути дублювання файлів і забезпечити відмовостійкість.



Зашифроване зберігання

Кожен файл має унікальний ідентифікатор, асоційований з повідомленням. Усе зберігається в зашифрованому вигляді.

Рисунок Б.6 — Хмарне зберігання

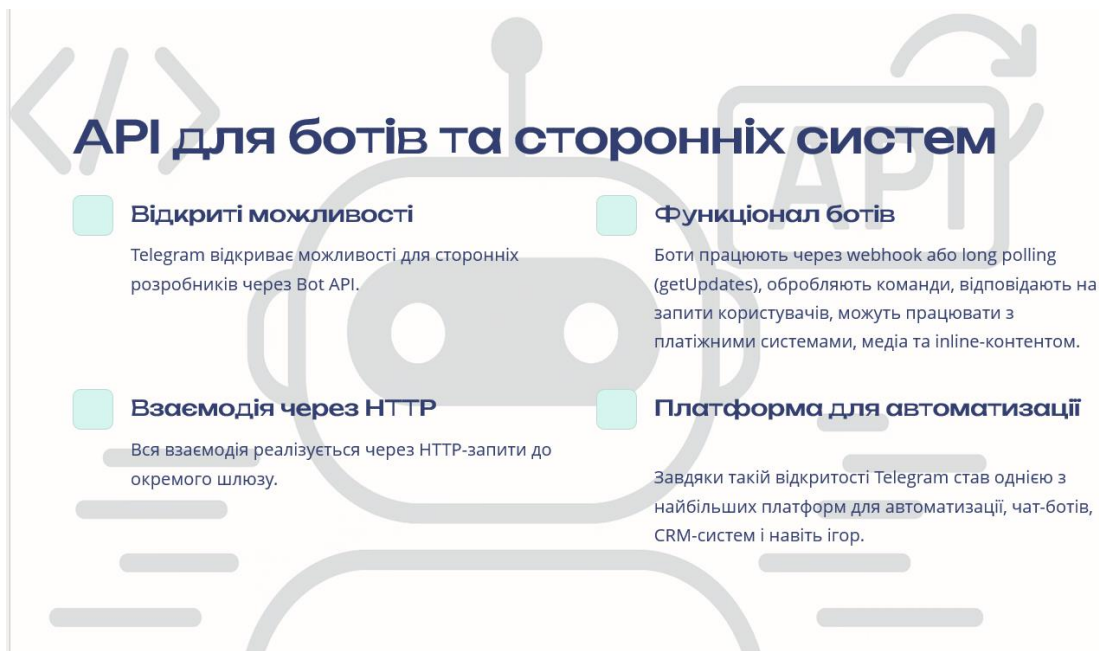


Рисунок Б.7 – Bot API

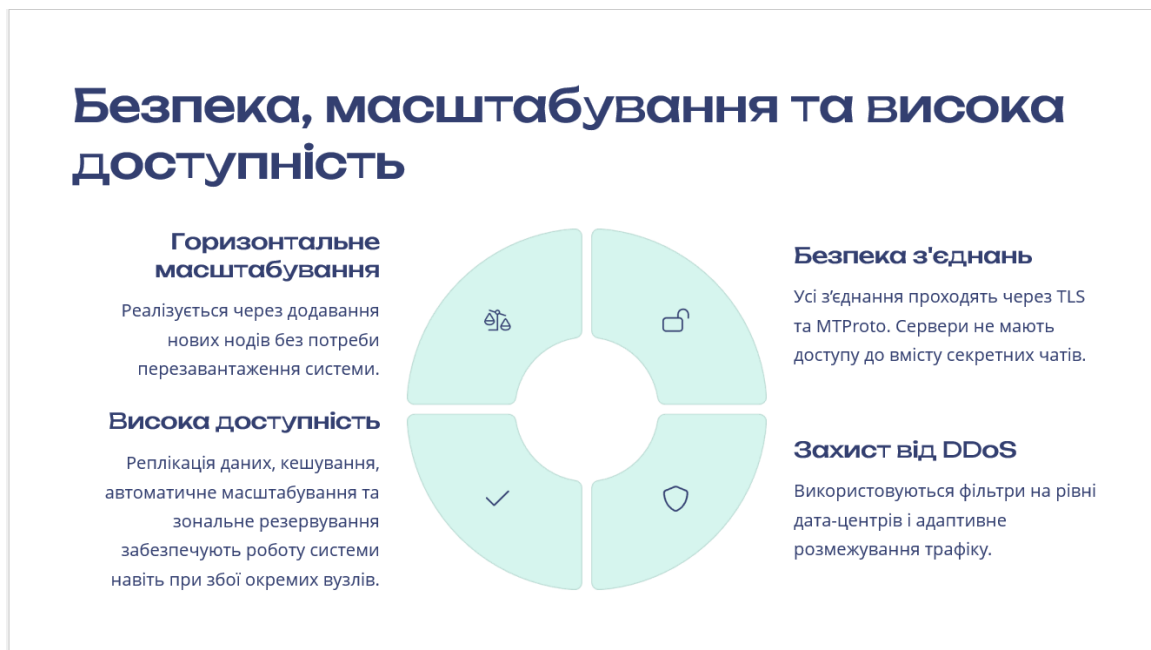


Рисунок Б.8 – Масштабування та безпека

Технології Telegram



Рисунок Б.9 – Технології Telegram

Висновки

Telegram — приклад сучасної високонавантаженої програмної системи з гнучкою та масштабованою архітектурою. Завдяки власному протоколу MTProto, розподіленій інфраструктурі, відкритому API і великому спектру функцій Telegram забезпечує швидку, безпечну та надійну комунікацію. Ця система демонструє, як можна поєднати масштабованість, зручність для користувача, безпеку і відкритість до розробників у єдиній архітектурній моделі.

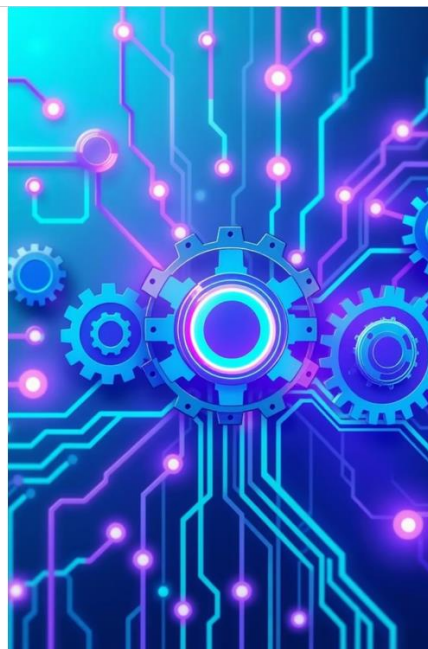


Рисунок Б.10 – Висновки

ДОДАТОК В

Для ілюстрації архітектурного підходу Telegram у додатку
представлено спрощений приклад Telegram-бота.

```
1  import telebot
2
3  bot = telebot.TeleBot('YOUR_BOT_TOKEN')
4
5  @bot.message_handler(commands=['start'])
6  def greet(message):
7      bot.reply_to(message, 'Вітаю! Це приклад
Telegram-бота.')
8
9  bot.polling()
```