НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Інститут телекомунікаційних систем

Кафедра Інформаційно-телекомунікаційних мереж

До захисту допущено:

Завідувач кафедри

_____ Лариса ГЛОБА

«___»_____2020 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційно-комунікаційні технології»

спеціальності 172 «Телекомунікації та радіотехніка»

на тему: «Удосконалена архітектура мережі Інтернет Речей»

Виконав:
студент IV курсу, групи ТІ- 62
Пороло Євгеній Олександрович _____

Керівник:
Асистентент кафедри ІТМ ІТС,
Курдеча В.В. _____

Рецензент:
Посада, науковий ступінь, вчене звання,
Прізвище, ім’я, по батькові _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
Кафедра Інформаційно-телекомунікаційних мереж

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інформаційно-комунікаційні технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Лариса ГЛОБА
«___»_____2020 р.

ЗАВДАННЯ
на дипломну роботу студенту
Пороло Євгенію Олександровичу

1. Тема роботи «Удосконалена архітектура мережі Інтернет Речей», керівник роботи асистент кафедри інформаційно-телекомунікаційних мереж ІТС Курдеча Василь Васильович, затверджені наказом по університету від «30» березня 2020 р. № 924-с

2. Термін подання студентом роботи 8 червня 2020 р.

3. Вихідні дані до роботи метод узгодження політик, наукові статті про архітектури Інтернету Речей.

4. Зміст роботи

1. Провести аналіз збереженості даних в мережі Інтернету речей.
2. Проаналізувати існуючі способи забезпечення конфіденційності інформації в мережі Інтернету речей.
3. Удосконалити архітектуру мережі інтернету речей за рахунок концепції Data Bank.
4. Запропонувати спосіб управління інформацією для модифікованої архітектури Інтернету речей.
5. Провести оцінку запропонованого рішення.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

1. Титульний слайд
2. Актуальність
3. Основні задачі
4. Проблеми безпеки та конфіденційності в мережі IoT
5. Загальна архітектура мережі IoT
6. Огляд існуючих рішень
7. Модифікована архітектура
8. Модифіквана архітектура
9. Реалізація запропонованого методу узгодження
10. Результати моделювання
11. Оцінка запропонованого методу
12. Загальні висновки

6. Дата видачі завдання 2 жовтня 2019 року

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблем збереженості даних в мережі Інтернту речей	06.10.2019 – 15. 11. 2019	Виконано
2	Аналіз існуючих рішень, що забепечують конфіденційність даних	16.11.2019 – 28.12.2019	Виконано
3	Порівняння методів узгодження політик конфіденційності користувачів	29.12.2019 - 04.02.2020	Виконано
4	Проведення моделювання запропонованого методу узгодження політик	05.02.2020 – 04.03.2020	Виконано
5	Написання наукової статті (ПРІТС-2020) та виступ	05.03.2020 – 13.04.2020	Виконано
6	Аналіз результатів запропонованого методу	14.04.2020 – 10.05.2020	Виконано
7	Оформлення дипломної роботи	12.05.2020 – 08.06.2020	Виконано

Студент Євгеній ПОРОЛО

Керівник Василь КУРДЕЧА

РЕФЕРАТ

Робота містить 59 сторінки, 18 рисунків, 5 таблиць. Було використано 22 джерела.

Актуальність: актуальність роботи полягає у тому, що на даний час отримав розвитку спосіб надання безкоштовних послуг в обмін на дані користувачів. Технології Інтернету речей приносять багато переваг користувачам, але також викликають занепокоєння щодо безпеки та конфіденційності. Більшість пристроїв можуть передавати особисті дані третім особам. Внаслідок цього, існує явна потреба в розробці хмарної інфраструктури Інтернету речей, щоб дозволити користувачам контролювати свої дані.

Мета роботи: підвищити ефективність управління персональною інформацією та забезпечити збереженість даних за рахунок удосконалення архітектури мережі Інтернету речей.

Ключові слова: Інтернет речей, політика конфіденційності, архітектура мережі, метод узгодження, персональні дані.

ABSTRACT

The work contains 59 pages, 18 figures, 5 tables. 22 sources were used.

Relevance: the relevance of the work is that currently developed a way to provide free services in exchange for user data. IoT technologies bring many benefits to users, but also cause security and privacy concerns. Most devices can transfer personal data to third parties. As a result, there is a clear need to develop an Internet of Things cloud infrastructure to allow users to control their data.

Purpose: to increase the efficiency of personal information management and ensure data safety by improving the architecture of the Internet of Things.

Keywords: Internet of Things, privacy policy, network architecture, reconciliation method, personal data.

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1 10
АНАЛІЗ ХАРАКТЕРИСТИК ТА ПРОБЛЕМ ЗБЕРЕЖЕНОСТІ ДАНИХ В МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ 10
1.1 Технології хмарних сервісів та Інтернету речей 10
1.2 Класифікація моделей хмарних інфраструктур Інтернету речей 12
1.3 Проблеми безпеки та конфіденційності в мережі IoT 16
1.4 Ризики та випробування пов’язані з використанням пристроїв IoT 18
Висновки: 22
РОЗДІЛ 2 23
ПІДХОДИ ДЛЯ ВИРІШЕННЯ ПРОБЛЕМ ЗАХИСТУ ТА КОНФІДЕНЦІЙНОСТІ ІНФОРМАЦІЇ 23
2.1 Концепція Data Box 23
2.2 Концепція Personal Data Vaults 26
2.3 Концепція Data Bank 30
Висновки: 33
РОЗДІЛ 3 35
МОДИФІКАЦІЯ АРХІТЕКТУРИ МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ ЗА РАХУНОК КОНЦЕПЦІЇ DATA BANK 35
3.1 Впровадження концепції Data Bank 35
3.2 Вимоги щодо методу узгодження конфіденційності 37
3.3 Метод узгодження конфіденційності 38
3.4 Загальна структура використання Data Bank 45
Висновки: 49
РОЗДІЛ 4 50
РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОГО МЕТОДУ ТА АНАЛІТИЧНА ОЦІНКА 50
4.1 Натурне моделювання запропонованого методу 50
4.2 Аналітична оцінка методу 52
Висновки: 55
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ 56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 57

ПЕРЕЛІК СКОРОЧЕНЬ

IoT	Internet of Things
AWS	Amazon Web Services
WSN	Wireless Sensor Network
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
CSP	Content Security Policy
ACL	Access Control List
DAS	Data Collection System
API	Application Programming Interface
ABAC	Attribute-Based Access Control
GDPR	General Data Protection Regulation
XML	eXtensible Markup Language
P3P	Platform for Privacy Preferences

ВСТУП

Актуальність: актуальність роботи полягає у тому, що з кожним роком зростає кількість користувачів Інтернет, кількість вузлів міжмашинного обміну інформації та об’єм передаваних даних. А тому з’явилась тенденція, за якої користувачі оплачують інформаційні послуги за рахунок доступу до своїх даних та відповідної їх статистичної обробки.

Ситуація ускладнюється появою Інтернету речей (IoT), який дозволяє збирати величезну кількість персональних даних пристроями та передавати їх на хмарні сервери.

Внаслідок цього, існує явна потреба в розробці хмарної інфраструктури Інтернету речей, щоб дозволити користувачам контролювати свої дані.

Мета й завдання дослідження

Мета роботи: підвищити ефективність управління персональною інформацією та забезпечити збереженість даних за рахунок удосконалення архітектури мережі Інтернету речей.

Для досягнення мети дослідження було поставлено та вирішено такі **основні задачі:**

1. Провести аналіз збереженості даних в мережі Інтернету речей.
2. Проаналізувати існуючі способи забезпечення конфіденційності інформації в мережі Інтернету речей
3. Удосконалити архітектуру мережі інтернету речей за рахунок концепції Data Bank
4. Запропонувати спосіб управління інформацією для модифікованої архітектури Інтернету речей
5. Провести оцінку запропонованого рішення

Об’єкт дослідження: процеси управління персональною інформацією та забезпечення збереженості даних.

Предмет дослідження: архітектури мережі Інтернету речей.

Теоретичний результат дослідження: модифікована архітектура для збереженості конфіденційності користувачів, що відрізняється наявністю методу узгодження політик.

Практичний результат роботи: можливість управління політиками на основі застосування модифікованої архітектури з використанням методу узгодження.

РОЗДІЛ 1

АНАЛІЗ ХАРАКТЕРИСТИК ТА ПРОБЛЕМ ЗБЕРЕЖЕНОСТІ ДАНИХ В МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Технології хмарних сервісів та Інтернету речей

Інтернет речей (IoT) відкриває нові можливості для розвитку програм, які покращують життя людини. Ці програми охоплюють різні сфери, такі як розумні міста, промислові додатки, домашня автоматизація та медицина. Крім того, постійні інновації в розробці обладнання, програмного забезпечення та технологій бездротового зв’язку протягом останнього десятиліття надавали IoT статусу інноваційної концепції і призвели до розширення розумних об'єктів, оскільки кількість підключених пристроїв збільшується з кожним днем.

Ці технології обіцяють полегшити наше життя, заощадити наш час, звільнить наш мозок від запам'ятовування логістичних даних, таких як маршрути, час прийому ліків тощо. Постійний ріст використання цих об'єктів призводить незліченної кількості генерованих даних. У цьому світлі, враховуючи постійно зростаючі здібності платформ зондування, обчислювальну та комунікаційну здатність смарт-пристроїв, Cisco Internet Business Solutions Group передбачила, що з кінця 2019 року розумні пристрої IoT щорічно генеруватимуть понад 500 зеттабайт неструктурованих та структурованих даних. Крім того, очікується, що ця кількість буде експоненціально зростати. Більше того, прогнози галузей передбачають понад 50 мільярдів підключених пристроїв до Інтернету на кінець 2020 року[1].

Більшість додатків IoT засновані на розгортанні високого числа датчиків, щоб найкраще контролювати програмне середовище. Датчики в IoT є обмеженими пристроями, тобто мають низьку потужність та обмежену пам'ять, що призводить до складнощів в зберіганні та обробці інформації.

Використання хмарних сервісів IoT вирішує таку проблему, оскільки надається достатня кількість пам’яті для зберігання даних. Завдяки хмарному IoT величезний обсяг даних захоплений через WSN може зберігатися в хмарі та бути доступним для кількох користувачів. Використання такого підходу зменшує загальну вартість збору даних як для системи, так і для користувачів. Оскільки різні WSN мають різних власників, тому на хмарних сервісах розміщуються різні програми. Використовуючи послуги хмарних обчислень, користувачі звертають увагу, що затримка є критичним моментом при забезпеченні потоку даних в режимі реального часу. Наприклад, датчики які встановлені на інфраструктурі дорожнього руху, повинні надавати дані потоку руху в режимі реального часу, щоб водії, які перебувають у надзвичайній ситуації, користувалися цими даними для вибору менш перевантажених доріг, щоб якнайшвидше прибути до місця призначення. Ці дані повинні ефективно зберігатися та відправлятися.

Однією з головних проблем розподілених серверів є забезпечення високої доступності. Досягнення такої послуги є важкою задачею, коли кількість даних, що зберігаються у хмарі стрімко зростає. В основному проблеми з доступністю даних відбуваються через неефективну організацію зберігання даних. Відсутність ефективної організації зберігання даних, в довгостроковому результаті призводить до перевантаження деяких серверів з точки зору процесора та пам'яті, а інші сервери залишаються неактивними[2].

Для виконання ефективних запитів введення та виведення інформації, слід розгорнути балансування навантаження, що надходить на сервери. Балансування навантаження - це техніка оптимізації, при якій прагнуть розподілити навантаження між серверами так, щоб забезпечити найкращу доступність даних на запит, що відправив користувач. Таким чином, це зменшує затримку і збільшує пропускну здатність. Завжди забезпечувати високу доступність даних у критичних програмах, ефективний моніторинг даних, резервне копіювання та відновлення даних дуже важливо.

1.2 Класифікація моделей хмарних інфраструктур Інтернету речей

Інтернет речей отримує користь від масштабованості, продуктивності та характеру оплати інфраструктури хмарних обчислень. Оскільки програми IoT виробляють великі обсяги даних і містять декілька обчислювальних компонентів (наприклад, алгоритми обробки даних та аналітики), їх інтеграція в інфраструктуру хмарних обчислень може забезпечити можливість економічно вигідного масштабування. Подібно до інфраструктури хмарних обчислень, хмарні інфраструктури IoT та пов'язані з ними сервіси можна класифікувати на наступні моделі:

– Програмне забезпечення як сервіс (SaaS).

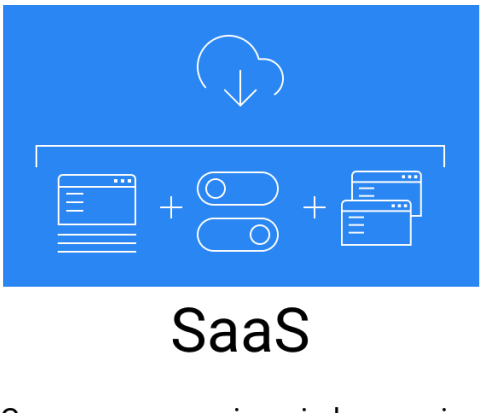

Рис.1.1 Програмне забезпечення як сервіс

Програмне забезпечення як послуга, також відоме як хмарні послуги додатків, являє собою найбільш часто використовуваний варіант для підприємств на хмарному ринку. SaaS використовує Інтернет, щоб доставляти додатки, якими керує сторонній постачальник, своїм користувачам. Більшість програм SaaS запускаються безпосередньо через ваш веб-браузер, а це означає, що вони не потребують завантаження або встановлення на стороні клієнта.

Завдяки своїй моделі доставки веб-сайтів, SaaS усуває необхідність завантаження та встановлення програм IT-персоналу на кожному окремому комп’ютері. За допомогою SaaS постачальники керують усіма потенційними

технічними проблемами, такими як дані, проміжне програмне забезпечення, сервери та сховища, що призводить до спрощеного обслуговування та підтримки бізнесу[3].

Найнижча вартість серед інших хмарних рішень пов'язана з високим рівнем залежності від постачальника програмного забезпечення. Всі компоненти, включаючи додатки та дані користувачів керуються за допомогою CSP. Це може привести до проблем з дотриманням конфіденційності і безпеки даних. Крім того, часто доводиться жертвувати налаштуванням заради високої доступності, якщо ви вирішили використовувати готове рішення.

– Платформа як сервіс (PaaS).

Рис. 1.2 Платформа як сервіс

Послуги хмарної платформи надають хмарні компоненти певному програмному забезпеченню, що використовується в основному для додатків. PaaS пропонує структуру для розробників, яку вони можуть будувати та використовувати для створення спеціальних додатків. Усіма серверами, сховищами та мережами може керувати підприємство або сторонній постачальник, тоді як розробники можуть підтримувати управління додатками.

Модель доставки PaaS схожа на SaaS, за винятком того, що замість доставки програмного забезпечення через Інтернет, PaaS пропонує платформу для створення програмного забезпечення. Ця платформа надається через Інтернет, що дає розробникам свободу сконцентруватися на створенні програмного забезпечення, не турбуючись про операційні системи, оновлення програмного забезпечення, зберігання чи інфраструктуру. PaaS дозволяє бізнесу розробляти та створювати додатки, вбудовані в PaaS за допомогою спеціальних програмних компонентів. Ці програми, які іноді називають середнім програмним забезпеченням, є масштабованими та широко доступними, оскільки вони набувають певних характеристик хмари[4].

В разі невдалого вибору постачальника PaaS, ви будете мати проблеми з несправностями сервера або дотриманням вимог управління даними. Ще один з ризиків полягає в тому, що ці рішення потребують дуже стабільного з'єднання та достатньої пропускної здатності для безперебійної роботи в будь-який час.

– Інфраструктура як сервіс (IaaS).

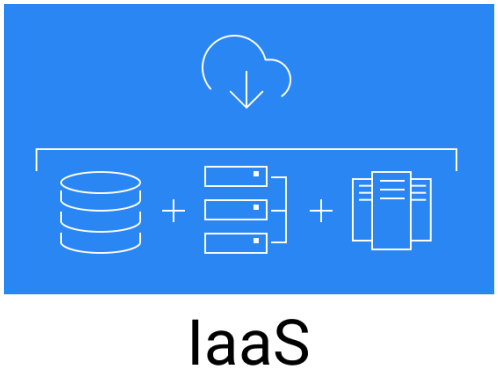

Рис. 1.3 Інфраструктура як сервіс

Послуги хмарної інфраструктури складаються з високомасштабованих та автоматизованих обчислювальних ресурсів. IaaS - це повністю самостійний сервіс для доступу та моніторингу комп'ютерів, мереж, зберігання та інших послуг. IaaS дозволяє підприємствам купувати за потребою та в міру необхідності ресурси, а не обладнання безпосередньо. IaaS забезпечує інфраструктуру хмарних обчислень, включаючи сервери, мережу, операційні системи та сховище, за допомогою технології віртуалізації.

Ці хмарні сервери зазвичай надаються організації через інформаційну панель або API, що дає клієнтам IaaS повний контроль над усією інфраструктурою. IaaS надає ті самі технології та можливості, що і традиційний центр обробки даних, не потребуючи фізичного обслуговування або управління ним. Клієнти IaaS все ще можуть отримувати доступ до своїх серверів та сховищ даних безпосередньо, але все це передається через "віртуальний центр даних" у хмарі.[5]

Проблеми можуть виникати в дотриманні регламенту по управлінню даними. Потенційна втрата даних – це ще один ризик, при використанні IaaS, необхідно мати комплексний план відновлення, в разі випадку таки інцидентів.

Основні відмінності між даними архітектурами полягають в управлінні компонентами, що входять до їх скаду.

Рис. 1.4 Основні відмінності IaaS, PaaS та SaaS

1.3 Проблеми безпеки та конфіденційності в мережі IoT

У IoT кожний підключений пристрій може бути потенційним дверним прорізом в інфраструктуру IoT або до особистих даних. Проблеми безпеки та конфіденційності даних дуже важливі, але потенційні ризики, пов'язані з IoT, вийдуть на новий рівень, оскільки функціональна сумісність, гібридні програми і автономне прийняття рішень починають додавати складність системи, лазівки в системі безпеки і потенційну вразливість. У IoT виникнуть ризики пов’язані з конфіденційністю, оскільки складна структура мережі може створити більше вразливостей. У IoT велика частина інформації пов'язана з нашими особистими даними, такі як дата народження, місце розташування, особові рахунки і т.д. Це один з аспектів проблем з великими даними і професіонали в галузі безпеки повинні будуть переконатися, що вони враховують всі потенційні ризики, пов'язані з конфіденційністю даних.

IoT повинен бути реалізований законним, етичним, соціально і політично прийнятним способом, в якому повинні враховуватися правові проблеми, систематичні підходи, технічні проблеми і проблеми бізнесу[6]. Основні проблеми продемонстровані на рисунку 1.5.

Рисунок. 1.5 – Проблеми пов’язані з Інтернетом речей

Такі методи, як програмування пристроїв, шифрування пам’яті та контрольовані списки доступу (ACL) мають досить інертний характер. Мало уваги приділяється застосуванню ефективних заходів безпеки та конфіденційності для захисту даних і контролю доступу відповідно до динаміки ресурсів, середовища, користувачів або програм. Крім того, посилення втручання користувача в захист його власних даних є ще однією проблемою[7].

Таким чином, необхідний тонкий контроль доступу та різноманітні варіанти конфіденційності для користувача. Безпека повинна вирішуватися на протязі всього життєвого циклу IoT, від початкового проекту до працюючих служб. Отже, цілком важливо усвідомити, що методи безпеки та конфіденційності мають бути більш динамічними та надійними за своєю суттю, щоб мати можливість відповідати на зростаючі потреби цифрового світу.

1.4 Ризики та випробування пов’язані з використанням пристроїв IoT

Сенсорні пристрої IoT, такі як фітнес трекери, пристрої моніторингу стану здоров'я і автоматичні платники рахунків, обробляють високочутливі споживчі та фінансові дані. Оптимістичне зростання обсягу даних і підключених пристроїв або складних систем, таких як інтелектуальні пристрої і транспортні засоби, монітори працездатності і датчики, викликають великий сплеск проблем, з точки зору обробки даних, транспорту, безпеки і контролю доступу.

Підключені до мережі речі, люди і пристрої являють собою типові, керовані даними, сценарії IoT. Хоча повністю підключений до інтернету світ з інтелектуальними обчислювальними системами і інтелектуальними пристроями, що забезпечують повсякденне життя, звучить досить перспективно, але відсутність належних механізмів безпеки і конфіденційності для захисту персональних даних несе, як потенційні фізичні ризики так і ризики пов’язані з безпекою.

Мережа пристроїв записує, керує, зберігає і транспортує величезну кількість конфіденційних даних. З постійним розвитком способів обробки і аналізу згенерованих даних, проблеми конфіденційності та безпеки неминучі. У той час як несанкціонований доступ і неправильне використання даних з датчиків, таких як глюкометри, ваги, монітори серцевого ритму і артеріального тиску і таких пристроїв, як кардіостимулятор, можуть привести до серйозної небезпеки для здоров'я (або навіть призвести до смертельного результату), мережеві пристрої, такі як побутові прилади (дверні замки, камери, печі) можуть бути доступні зловмисникам. Невідповідний доступ до кредитної карти або контроль над транспортним засобом, небезпечний як за фізичними, так і за фінансовими причинами.

Сучасні методи забезпечення безпеки і конфіденційності в Інтернеті не дозволяють вирішити проблеми безпеки даних, заснованих на Інтернеті речей, такі як різна власність, управління ключами, закони і правила безпеки,

різнорідне підключення і життєвий цикл різних пристроїв [8, 9]. Пристрої з вбудованими датчиками розрізняються по марці, типу і використанням.

Пристрій IoT може використовуватися одним або декількома користувачами для виявлення, зберігання або обробки конфіденційних даних, отже, застосування відповідної системи контролю доступу та автентифікації особистості є досить складним завданням. На відміну від традиційних систем даних, дані з пристроїв IoT неоднорідні і можуть передаватися з непередбачуваною швидкістю. У класичному веб-налаштуванні, записані дані можуть бути зашифровані та надіслані по захищених каналах зв'язку для обробки на стороні сервера.

Однак в інтелектуальних пристроях IoT такі операції на стороні сервера, як аналіз, обробка і виведення даних, більшою мірою децентралізовані. Дані, що передаються по каналах зв'язку, варіюються від показань датчиків до аналітичної інформації. Це робить канали передачі даних більш уразливими для злому і атаки зловмисників.

У порівнянні з типовою клієнт-серверною або розподіленою веб-інфраструктурою, архітектура IoT відрізняється з точки зору обсягу та різноманітності даних, можливості підключення, управління контролем і розподілу обчислень між різними рівнями. Отже, дуже важливо розуміти проблеми безпеки, з якими стикається мережа IoT щодо її багаторівневої архітектури, рисунок 1.6.

Рис. 1.6 Загальна архітектура мережі Інтернету речей

На рисунку 1.6 зображена архітектура, що складається з чотирьох частин, яка сьогодні підтримує багато систем IoT[10].

— Рівень Датчиків та приводів

Відмінною особливістю датчиків є їх здатність перетворювати інформацію, отриману з зовнішнього світу, в дані для аналізу. Іншими словами, важливо почати з включення датчиків на 4 етапі структури архітектури IoT, щоб отримати інформацію, яка може бути фактично оброблена. Для приводів процес йде ще далі - ці пристрої здатні втручатися в фізичну реальність. Наприклад, вони можуть вимкнути світло і відрегулювати температуру в кімнаті. Через це рівень датчиків та приводів

охоплює і коригує все, що необхідно в фізичному світі, щоб отримати необхідну інформацію для подальшого аналізу.

— Рівень Збору даних

Незважаючи на те, що цей етап архітектури IoT як і раніше має на увазі роботу в безпосередній близькості з датчиками і виконавчими механізмами, тут також присутні міжмережеві інтерфейси і системи збору даних (DAS). Зокрема, останні підключаються до мережі датчиків, в той час як інтернет-шлюзи працюють через Wi-Fi, провідні локальні мережі та виконують подальшу обробку. Життєво важливе значення цього етапу полягає в обробці величезної кількості інформації, зібраної на попередньому етапі і стисненні її до оптимального розміру для подальшого аналізу. Крім того, тут відбувається необхідне перетворення з точки зору структури.

— Рівень Систем обробки та обчислення

На цьому етапі архітектури IoT, підготовлені дані переносяться в мультисервісну систему локального зберігання, обчислень або логічного обробки. Зокрема, сучасні IT-системи виконують розширену аналітику і попередню обробку. Наприклад, це відноситься до технологій машинного навчання і візуалізації. У той же час тут може статися деяка додаткова обробка до етапу входу в центр обробки даних. Крім того, тут повинні бути створені відповідні служби безпеки для захисту небезпечних кінцевих точок.

— Рівень Центру обробки даних

Основні процеси на останньому етапі архітектури IoT відбуваються в центрі обробки даних або в хмарі. В хмарному центрі обробки даних розміщуються додатки, що надають послуги для управління архітектурою Інтернету речей. Служби безпеки мають вирішальне значення на цьому рівні для запобігання проникнення і експлуатації кінцевих точок від зловмисників.

Життєво важливо керувати об’ємом і неоднорідністю даних, а також забезпечувати конфіденційність даних для декількох споживачів. Використання класичного криптографічного шифрування для безпечного зберігання даних може бути складним в управлінні, оскільки вони можуть

бути зламані за короткий час. Для пом’якшення загроз, таких як скомпрометовані системи, повторні транзакції або відмова в обслуговуванні, потрібні відповідні методи забезпечення безпеки. Для спрощення дуже різноманітного середовища IoT і пов’язаних з цим проблем безпеки, обов’язково потрібна гнучка структура безпеки. Загрози на рівнях IoT слід усувати за допомогою спеціальних механізмів забезпечення конфіденційності і безпеки даних, розроблених на основі чіткого розуміння вимог безпеки, специфічних для IoT.

Висновки:

1) Проаналізовано технології та їхні характеристики, що використовуються в Інтернеті речей та хмарних обчисленнях. Визначено ряд проблем, що можуть виникати в результаті використання хмарних рішень IoT.
2) Проведено аналіз моделей хмарних інфраструктур, що використовуються в Інтернеті речей. Проаналізовані відмінності між ними та проблеми, що виникають в кожній із моделей.
3) Проведено аналіз проблем збереження та конфіденційності даних в мережі Інтернету речей, розглянуто загальну архітектуру та проблеми на кожному з рівнів.

РОЗДІЛ 2

ПІДХОДИ ДЛЯ ВИРІШЕННЯ ПРОБЛЕМ ЗАХИСТУ ТА КОНФІДЕНЦІЙНОСТІ ІНФОРМАЦІЇ

2.1 Концепція Data Box

Серед підходів, що вирішують питання безпеки та конфіденційності персональних даних користувачів при використанні хмарних рішень Інтернету речей, виділяються наступні три DataBox, Personal Data Vaults, Data Bank.

Розробники концепції Databox ділять її характеристику на чотири частини: це повинна бути *надійна платформа*, що забезпечує засоби для *управління даними*, а також *контрольований доступ* для інших сторін, які бажають використовувати свої дані та *підтримує стимули* для всіх сторін[11].

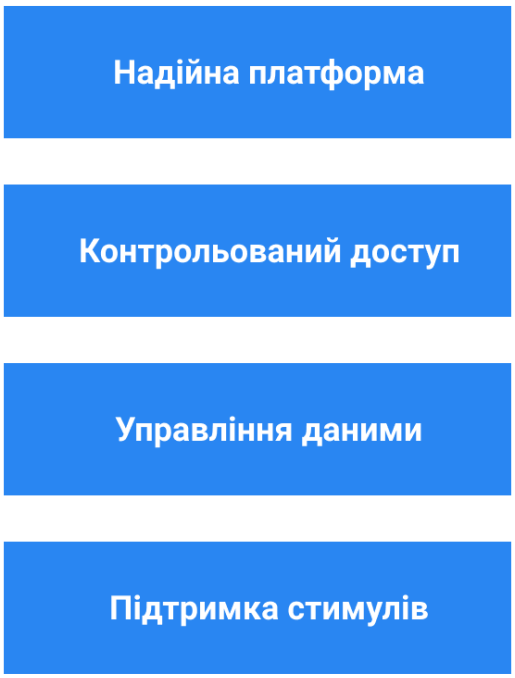

Рис. 2.1 Характеристика концепції Databox

— Надійна платформа

Databox знаходиться в основі вашої присутності в Інтернеті. Він фіксує, індексує, зберігає та керує даними про вас і тими, що були створені вами.

Для цього доведеться довіряти цій платформі, а також вручну додавати дані та індекси у вашому Databox, дані виводити з різних джерел, таких як встановлені додатки, уподобання перегляду та поведінка в Інтернеті. Це потенційно робить його набагато більш обізнаною та нав'язливою системою (хоча і кориснішою).

Довіра до платформи також вимагає надійної поведінки як частини інфраструктури. Тобто, Databox повинен мати доступ, щоб він міг корисно допомагати користувачеві керувати їхніми онлайн-взаємодіями. У той же час це повинно забезпечити простий спосіб для втручання користувача в операції збору даних та обміну ними, що проводяться для запобігання порушенням у випадках, коли автоматичні дії, отримані внаслідок конфігурації та політики, мають непередбачені наслідки.

Нарешті, всі ці дії та поведінка повинні підтримується повсюдним веденням журналу, за допомогою відповідних інструментів, щоб користувачі та (потенційно) сторонні аудитори могли створити довіру до того, що система працює як очікувалося і якщо трапиться щось непередбачене, то результати помилки можна було відслідкувати.

— Контрольований доступ

Мета Databox не просто зібрати всі ваші особисті дані в одне місце, але і забезпечити контроль доступу до цих даних. Користувачі повинні мати детальний контроль над тим, які дані надаються третім сторонам. Більш складні можливості включають підтримку збереження конфіденційності таких методів аналізу даних, як диференціальна конфіденційність та гомоморфне шифрування.

Один важливий фактор, який часто враховується поточними системами - це необхідність контролювати період доступу в кожному конкретному випадку. Зокрема, потреба відкликати раніше наданий доступ. У системі де доступ надається для обробки даних локально, це відносно просто; але в системі, де дані за замовчуванням копіюються в стороннє співробітництво це досить складне завдання. Завдання полягає в підтримуванні або отримуванні

існуючої та майбутньої інформації про стани всіх потенційних третіх сторін, які можуть отримати доступ до нещодавніх даних.

— Управління даними

Так само як збір ваших даних та надання засобів для контрольованого доступ до них, Databox має забезпечувати засоби для взаємодії з користувачами та відображати дані, які він містить. Це дасть змогу користувачам приймати більш обізнані рішення щодо поведінки, яку вони впроваджують, чи безпосередньо самі, чи опосередковано, передаючи контроль іншим. В рамках цих взаємодій і для підтримки довіри, на платформі користувачі повинні мати можливість редагувати та видаляти дані зі свого Databox, розглядаючи це як спосіб вирішити неминучі випадки.

Аналогічно, це може бути доречним для деяких даних і для деяких користувачів, щоб Databox не дозволив проявляти звичну цифрову тенденцію ідеального запису. Це означає, що Databox отримає дозвіл автоматично забувати дані, які вже не є актуальними або стали неправдивими. Навіть якщо дані раніше використовувалися, їх все одно потрібно буде "поставити" за межі використання користувачам, які бажають редагувати їх в майбутньому. Такі локальні та глобальні поняття, як право бути забутим вимагають дотримання узгоджених протоколів і інші форми співпраці з сторонніми службами та агрегаторами даних.

— Підтримка стимулів

Для розвитку інноваційного використання персональних даних потрібний стимул. Наслідком вищезазначеного контрольованого доступу є те, що користувачі можуть відмовити стороннім службам (наприклад, рекламодавцям або постачальникам хмарних послуг) отримати доступ до своїх даних. У найпростішому випадку це може призвести до того, що користувачі просто більше не зможуть скористатися цими послугами. Однак більш вигідно було б забезпечити засоби для цих послуг, що стягували б плату з користувача іншими способами: ті, хто бажає платити доступом до

своїх даних, можуть це зробити, тоді як ті, хто не хоче розкривати персональну інформацію, можуть платити традиційними засобами.

Тобто, Databox має бути в змозі дозволити користувачам відстежувати платежі поряд із потоком даних до та від інших сторонніх послуг, доступних через певну форму магазину додатків. Databox також може діяти як механізм зменшення експозиції для комерційних організацій, які більше не мають наміру безпосередньо зберігати та контролювати цілий спектр приватних даних (наприклад, медичних записів). Комерційна організація все ще може отримати доступ та запитувати дані, як описано раніше. Це особливо актуально для міжнародних організацій, які в іншому випадку повинні знати про безліч правових рамок.

2.2 Концепція Personal Data Vaults

Personal Data Vaults (PDV) - це магазин персональних даних, який підтримує механізми власності персональних даних, селективний обмін та аудит, щоб забезпечити видимість в обміні даними. Система спирається на два припущення[12].

Перше полягає в тому, що кожен власник даних має логічно виразний PDV.

Друге припущення полягає в тому, що коли власник обмінюється даними з іншим суб'єктом господарювання, існує неявна або явна законодавчо закріплена угода про те, як суб'єкт господарювання буде використовувати отримані дані. Важливий виклик в структурі PDV є розробка такого механізму. Ці принципи проектування вимагають, щоб індивідуальне прийняття рішень щодо керованого обміну даними не було занадто складним та трудомістким, оскільки це зробить систему непридатною. Для досягнення принципів PDV озміщується між користувачем та постачальниками послуг контентного вмісту, як показано на рисунку 2.2.

Рис. 2.2 Принцип взаємодії PDV

PDV містить три механізми управління політикою даних: Гранульовані списки контролю доступу (ACL), Trace-аудит та Систему рекомендацій правил. Обмін даними опосередковано через PDV має багато переваг перед використанням централізованого стороннього зберігання даних.

До переваг можна віднести:

1) дозвіл користувачам володіти персональними даними;

2) дозвіл постачальникам послуг контентної служби отримувати доступ до даних з дозволу користувача;

3) зменшення можливості надмірності даних;

4) зменшення ймовірності проблем із втратою даних та обмеженням ресурсів, зберігаючи дані з мобільних пристроїв.

— Гранульований список контролю доступу

Традиційні ACL вказують, яким користувачам або системним процесам надається доступ до даних, а також які операції дозволені над даними. ACL для соціальних мереж були зосереджені на управлінні групами, хто має доступ до яких даних і менше приділяють уваги контролю даних для обміну.

На відміну від цього, розробники PDV намагаються підкреслити важливість детального (гранульованого) контролю даних для спільного використання.

Гранульований ACL містить суб’єкт та набір фільтрів. Суб'єкт визначає тип та назву третіх сторін, які отримують доступ до даних з PDV. Фільтр містить перелік обмежень, які визначають дані для обміну: кожне обмеження описується за типом та його атрибутами (див. Таблицю 2 для більш детальної інформації). Щоб бути дуже конкретними, використовується JavaScript Object Notation (JSON) для форматування ACL. Коли третя сторона, наприклад постачальник послуг контенту, запитує дані в PDV, то спочатку визначаються фільтри, пов’язані з відповідним записом, щоб вирішити, яку деталізацію даних надсилати.

Прототип PDV в даний час підтримує наступні три обмеження: Bounds (значення), Precision (ступінь точності), Frequency (частота повторення), таблиця 2.1, надалі буде добавлено більше обмежень для випадків загального користування у міру їх появи.

Таблиця 2.1.

Типи обмежень, що підтримуються в PDV.

Обмеження	Тип	Атрибути
Значення	час	starttime, endtime
	місцезнаходження	format(in-circle, out-circle), center(GPS coordinates),radius(in km)
	число	lower, lowersymbol(=, <, <=), upper, uppersymbol(=, >, >=)
	текст	attrname, text, symbol(=, !=)
Ступінь точності	час	value(private, second, minute, hour)
	місцезнаходження	value(private, exact, street, zipcode, state, country)
	число	value(private, average), timeframe(minute, hour, day, week, month)

Продовження таблиці 2.1

Обмеження	Тип	Атрибути
Частота повторення	час	unit(second, minute), value

— Trace-аудит

Прототип PDV системи дозволяє два типи механізмів Trace-аудиту: локальний Trace-аудит для журналу операцій, що виконуються всередині PDV і Trace-аудит для сторонніх додатків, щоб відстежувати дії, що відбуваються поза PDV (тобто сторонні додатки, які зчитують дані з PDV). Локальні журнали Trace-аудиту читають доступ до даних користувачів, що знаходяться у сховищі даних.

Кожен запис журналу представлено у вигляді наступного кортежу: <timestamp, appId, opType, dataTable, dataField1, dataField2, ..., startRow, endRow>, з інтерпретацією appId додаток виконано opType операція над таблицею даних dataTable в такий час timestamp і поля, що стосуються dataField1, dataField2, … починаючи з рядка startRow до endRow. Потім згенерована інформація про реєстрацію даних візуалізується та подається власникам PDV, щоб допомогти інтерпретувати, які дані були обмінені з якими програмами, враховуючи поточний набір політики.

Модуль Trace-аудиту для сторонніх додатків забезпечує видимість того, що відбувається з даними користувача, отриманими від їх PDV, всередині програми. Як і локальний Trace-аудит, програма реалізує Trace-аудит модуль для реєстрації даних, доступ до яких здійснюється, ця операція виконується разом із організацією, від імені якої отримано доступ до даних.

— Система рекомендацій правил

Система рекомендацій правил надає інтерфейс на високому рівні для встановлення політик обміну. Зокрема, вона попередньо обчислює значення

обмежень (ACL) для набору загальних намірів користувача на високому рівні Ці значення обмежень, отримані на основі історичних даних система рекомендацій робить розраховані значення доступними для користувачів (або довірених опікунів від імені користувачів), що полегшує повторну конфігурацію ACLs.

Так, використовуючи Систему рекомендацій правил, користувачі можуть більш активно брати участь у контрольованому обміні даними, а також постійний досвід роботи з системою дасть користувачам більш повне уявлення про наслідки політики конфіденційності. Використання лише системи рекомендацій недостатньо, оскільки з часом, запропоновані політики можуть фактично не відповідати намірам користувача. Користувачам потрібно буде перераховувати обмеження періодично або відповідно на аналіз Trace-аудиту.

2.3 Концепція Data Bank

Data Bank представлений платформою для керування даними, що надходять від пристроїв IoT та передаються в хмарні сервіси, рисунок 2.3. Він надає користувачам механізмами для визначення політики збору даних на рівні пристрою та політики обміну даними на хмарному рівні.

Архітектура цієї концепції складається з чотирьох рівнів: Прикладний рівень, Хмарний рівень, Локальне сховище (Data Pocket), Рівень Датчиків[13].

Прикладний рівень: це верхній рівень який складається з призначеного для користувача інтерфейсу, що визначає, як користувачі повинні взаємодіяти з Data Bank (він може бути розгорнутий на пристроях, підключених до Інтернету, через веб-сайт або мобільний додаток) та інтерфейсу / менеджеру обміну даними, який розглядається як прикладний програмний інтерфейс (API) і керує даними для сервісів.

Він діє як месенджер, приймаючи запити від сервісів, передаючи їх в модуль контролю доступу та повертаючи результати назад. У цій системі

користувачі є власниками даних, а служби - зовнішніми користувачами (третьою стороною), які хотіли б отримати доступ до даних або додатків, встановлених в Data Bank.

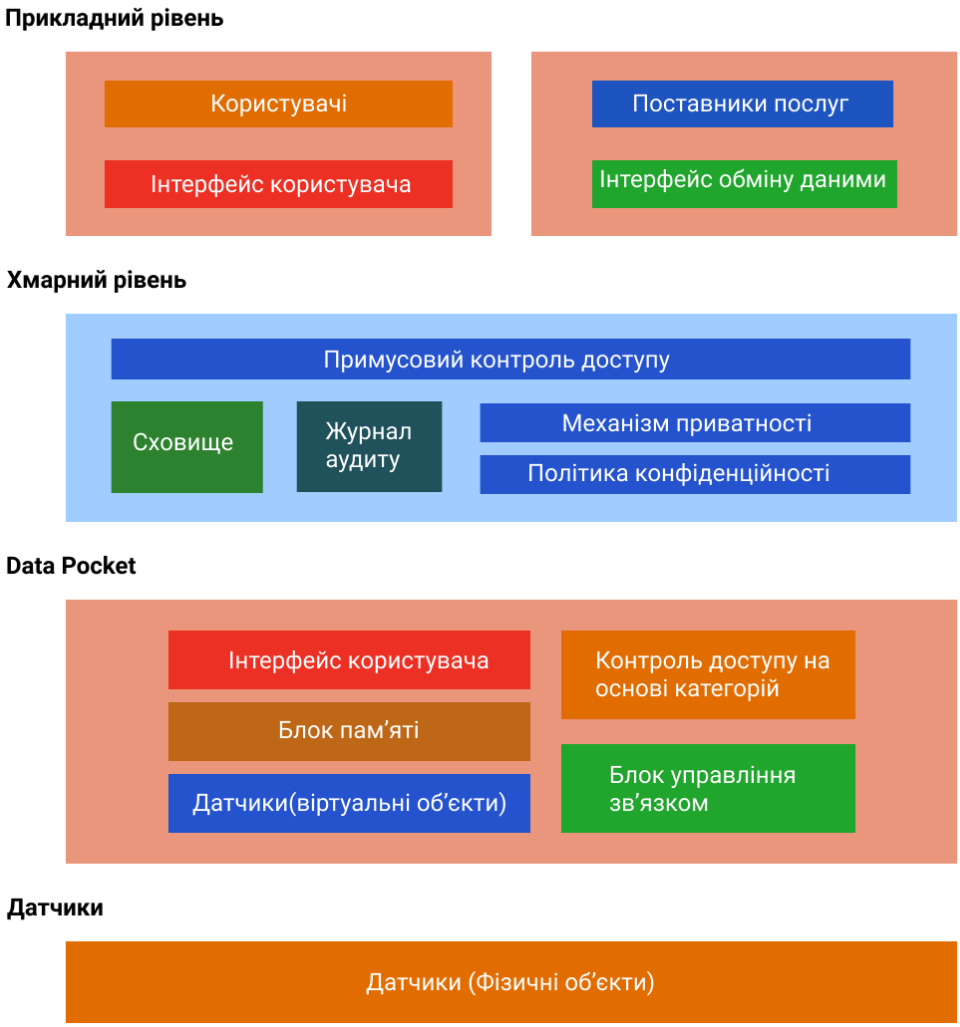

Рис. 2.3 Архітектура концепції Data Bank

Хмарний рівень: цей шар містить п'ять основних компонентів.

— Модуль контролю доступу отримує запити від сервісів і перевіряє, чи авторизований сервіс для доступу до запитуваних даних відповідно до політики. Цей контролер також готує дані для відповіді на запити.

— Модуль аудиту веде журнал всіх транзакцій, які відбуваються в Data Bank, наприклад, надано доступ до даних, заборонений доступ до даних, транзакції передачі даних і так далі.

— Сховище міститься в хмарі і зберігає всі дані користувачів. в формі, яка визначається політикою збору даних (наприклад, дані можуть бути відфільтровані та оброблені перед їх збереженням в хмарному сховищі). На додаток, сховище може також включати посилання на зовнішні сховища, підключені до Data Bank (наприклад, хмарне сховище третьої сторони).

— Механізм приватності пропонує послуги користувачеві з урахуванням зазначених налаштувань конфіденційності. Створює токени для доступу відповідних служб та сервісів. Також механізм виконує аналіз даних, наприклад, щоб ідентифікувати запити, які регулярно приймаються.

— Політика конфіденційності - це довідкова схема, запропонована механізмом утиліт конфіденційності і налаштовується користувачем. Ця політика призначена для модуля контролю доступу, який перевіряє, чи авторизований сервіс для доступу до запитуваних даних і для механізму утиліт конфіденційності для розподілення токенів між сервісами.

Рівень Pocket Data: призначений для користувача інтерфейс, працює в локальному обчислювальному пристрої (наприклад, концентратор IoT) і має справу з політиками збору даних. Призначення цього інтерфейсу - захопити налаштування конфіденційності користувача.

Рівень датчиків (фізичні об'єкти): цей рівень містить пристрої IoT, підключені через Інтернет або локальну мережу. Фізичні пристрої підключаються до Data Bank через вбудовані драйвери. Об'єктам в цьому шарі заборонено спілкуватися один з одним. Зв'язок між пристроями дозволений через віртуальні об'єкти, які керуються блоком управління зв'язком на рівні Data Pocket.

Порівняльна характеристика підходів для вирішення проблем захисту, управління та конфіденційності персональних даних користувача, продемонстрована в таблиці 2.2.

Таблиця 2.2

Технічні характеристики

	DataBox	PDV	Data Bank
Модель збору даних	—	✓	✓
Політики обміну даними	✓	✓	✓
Контроль збору даних на рівні пристроїв	✓	—	✓
Можливість анонімізації даних	✓	—	✓
Підтримка оновлень політик	✓	✓	✓
Специфікація політик	—	✓	✓

Як видно з таблиці, Data Bank надає користувачам тонкий контроль над своїми даними і дозволяє їм обмінюватися ними з сервісами відповідно до визначених користувачем політик, максимізуючи переваги для користувачів.

Відповідаючи на виклик конфіденційності, застосовується контроль даних як для збору так і для обміну даними: він включає механізми для визначення політики збору даних на рівні пристроїв, механізми для фільтрації та обробки даних перед передачею їх у хмарне сховище (замість скидання всіх необроблених даних в хмару), також включає в себе контроль доступу, який забезпечує, що зовнішні служби не мають прямого доступу до збережених даних. Користувач має контроль над політикою збору та обміну даними, що застосовується в Data Bank і може оновлювати політику в будь-який час.

Висновки:

1) Проведений аналіз існуючих методів, що забезпечують збереження конфіденційності та управління даними в мережі Інтернету речей. Проаналізовано особливості кожного з методів. Здійснено порівняння методів за технічними характеристиками та обрано відповідний, що

задовільняє потреби в конфіденційності, збереженості та управлінні персональними даними.

РОЗДІЛ 3

МОДИФІКАЦІЯ АРХІТЕКТУРИ МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ ЗА РАХУНОК КОНЦЕПЦІЇ DATA BANK

3.1 Впровадження концепції Data Bank

Використовуючи концепцію Data Bank можна модифікувати існуючу архітектуру Інтернету речей для відповідного захисту персональних даних користувачів та конфіденційності при використанні послуг в повсякденному житті. Опис модифікованої архітектури представлений нижче.

Рівень датчиків залишається, але з іншими компонентами всередині. Цей шар містить пристрої IoT, між якими заборонений зв’язок один з одним. Пристрої підключають до Data Bank через вбудовані драйвери та взаємодіють з віртуальними об’єктами через декілька стандартних протоколів, таких як Wi-Fi, Bluetooth, Z-хвиля, ZigBee.

Рівень відповідальний за збір та попередню підготовку даних замінюється на Data Pocket. В цьому шарі, користувачу надається інтерфейс, метою якого є захоплення налаштувань конфіденційності користувача. Для цього використовуються графічне представлення для політики контролю доступу та збору даних, які відображаються в зручному для користувача форматі. Користувачеві пропонується політика за замовчуванням, яку він налаштовує під свої потреби.

Для визначення політики використовується модель на основі категорій, яка є загальною моделлю, здатною представити високодинамічні політики (наприклад, політики ABAC). Блок управління зв’язком на цьому етапі забезпечує конфігурацію функцій, включаючи додавання або видалення датчиків та пристроїв і автоматичне оновлення політик.

Віртуальні об’єкти в цьому шарі містять методи та атрибути для представлення фізичних датчиків або пристроїв. Зв'язок серед віртуальних об'єктів здійснюється за допомогою блока управління зв’язком. Блок пам'яті

забезпечує зберігання необроблених даних з датчиків, після чого дані фільтруються відповідно до політики, перед завантаження в хмарне сховище.

Наступні два рівні, а саме Рівень Систем обробки та обчислення і Центру обробки даних створюють собою Хмарний рівень. Цей шар містить п'ять основних компонентів.

Модуль забезпечення контролю доступу отримує запити від служб і перевіряє чи службі дозволено отримувати доступ до запитуваних даних відповідно до політики. Цей контролер також готує дані для відповіді на запити.

Модуль аудиту зберігає журнал усіх транзакцій, що відбуваються в Data Bank, наприклад, наданий доступ до даних, заборонений доступ до даних, трансакції передачі даних тощо.

Сховище або Репозиторій розташований у хмарі і зберігає всі дані користувачів у формі, визначеній політикою збору даних (наприклад, дані можуть бути відфільтровані або оброблені до того, як перенесуться в хмару). Крім сховища, що надається Data Bank, репозиторій також може містити посилання на зовнішні сховища, додані до Data Bank (наприклад, хмарне сховище третьої сторони).

Механізм приватності пропонує послуги користувачеві з урахуванням зазначених налаштувань конфіденційності та переваг. Також механізм виконує аналіз даних, наприклад для ідентифікації запитів, які регулярно надходять. Вимоги щодо конфіденційності GDPR вирішуються, коли механізм пропонує політику за замовчуванням, а також наданням користувачеві контролю за обміном даними з постачальниками послуг.

Політика конфіденційності - це довідкова схема, яка пропонується механізмом конфіденційності та коригується користувачем. Це специфікація для модуля контролю доступу, щоб перевірити, чи служба уповноважена отримувати доступ до запитуваних даних, а також для механізму корисності-конфіденційності, щоб розподіляти маркери(токени) для служб.

Також в цій концепції до існуючої архітектури додається новий рівень – прикладний. Цей найвищий рівень складається з інтерфейсу призначеного для користувача, щоб взаємодіяти з Data Bank (він може бути розгорнутий на пристроях, підключених до Інтернету, через веб-сайт або мобільний додаток) та інтерфейсу / менеджеру обміну даними , який розглядається як інтерфейс прикладної програми (API) і контролює дані, видимі для служб. Він діє як месенджер, приймаючи запити від служб, передаючи їх до модуля контролю доступу та повертаючи результати назад до служб.

3.2 Вимоги щодо методу узгодження конфіденційності

Механізм приватності рисунок 3.1 спрямований на визначення необхідного для користувача рівня конфіденційності між персональними даними, що передаються та отриманими послугами в результаті цього обміну. Для цього потрібно застосовувати відповідну модель переговорів, що дозволяє інтерактивно встановити правильний баланс.

Важливо, щоб запропонований метод узгодження не тільки охоплював вимоги щодо конфіденційності користувачів IoT (тобто споживачів послуг IoT), але і розширював це покриття для узгодження та задоволення вимог конфіденційності постачальників послуг IoT (відтепер власника IoT).

Вимоги, які повинні бути виконані при розробці методу узгодження.

1) Немає участі користувача: все узгодження має відбуватися у фоновому режимі і без втручання користувача. Коли користувачі переміщаються по загальнодоступним або приватним місцям, вони, як правило, зісштовхуються з платформами IoT і беруть участь в обміні даними з ними для використання певної послуги. Через складність залучення користувача IoT, в такій ситуації пристрій який має налаштовані вимоги конфіденційності користувача, має діяти безперешкодно, щоб узгодити ці вимоги з постачальником послуг IoT. Крім того, налаштування конфіденційності постачальника IoT будуть повідомлені користувачеві для забезпечення відповідності між цими вимогами.

2) Мінімальні витрати: метод узгодження повинен накладати мінімальні витрати часу і енергії на виконання завдання IoT. Послуги IoT зазвичай надаються користувачам швидко. Тому ці служби чутливі до будь-яких тимчасових затримок. Крім того, оскільки пристрої IoT живляться від батареї, метод повинен бути енергоефективним, щоб не виснажувати джерела енергії задіяних пристроїв.
3) Гнучкість вибору: метод повинен уникати поточного повідомлення про конфіденційність і моделі вибору, що стосується прийняття сервісу в цілому або відмови від нього, оскільки ми бачимо, що ця модель жорстка і не буде працювати з ситуаціями IoT, які вимагають більшої гнучкості. Користувачам IoT повинні бути запропоновані різні варіанти, щоб вони могли продовжувати користуватися послугою, не жертвуючи при цьому своїми вимогами до конфіденційності.

Запропоновано використовувати метод узгодження конфіденційності для контролю доступу до даних користувачів [14].

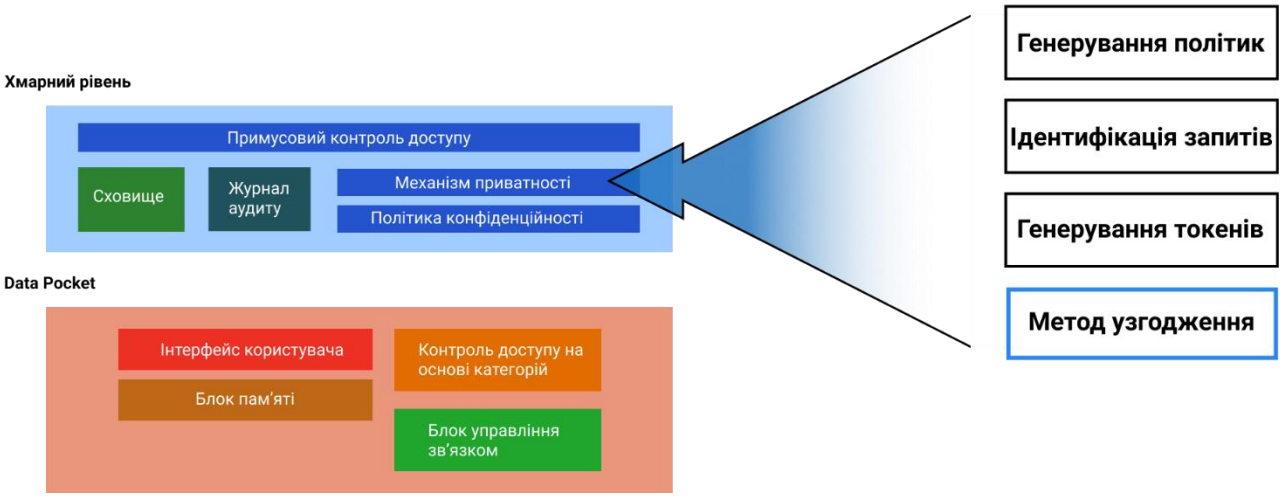

Рис. 3.1 Механізм приватності

3.3 Метод узгодження конфіденційності

Метод узгодження, представлений тут з концепцією досягнення відкритості в середовищі IoT. Відкритість передбачає, що мережі IoT будуть

діяти як комунальна інфраструктура, аналогічна електропостачанню та водопостачанню, доступ до яких користувачі IoT отримують на ходу [15].

Відкрите середовище вимагає, щоб сторони які беруть участь в обміні інформацією, вказували свої вимоги до конфіденційності, які повинні бути узгоджені від їх імені. Це аналогічно протоколу P3P [16], в якому браузер погоджує вимоги конфіденційності користувачів від їх імені з відвідуваних веб-сайтів, щоб контролювати особисту інформацію, яку веб-сайт може збирати про користувача.

У цьому методі, користувач IoT і власник IoT збережуть свої вимоги конфіденційності в файлі політики, що зберігається локально і використовує мову розмітки XML. Приклад сценарію для політик конфіденційності для користувача IoT та власника IoT, показаний в таблиці 3.1 і таблиці 3.2 відповідно.

Таблиця 3.1

Політика конфіденційності користувача.

```
<privacy-policy>
  <data-in type = “image” priority = “1”>
    <retention>3–month</retention>
    <shared>no</shared>
    <inferred>yes</inferred>
  </data–in>
  <data–out>
  <data–out type = “video” priority = “1”>
    <retention>1-year</retention>
    <shared>no</shared>
    <inferred>no</inferred>
  </data-out>
</ privacy-policy>
```

Таблиця 3.2

Політика конфіденційності власника.

```
<privacy-policy>
  <data-in type = “video” priority = “1”>
    <retention>1–year </retention>
    <shared>no</shared>
    <inferred>yes</inferred>
  </data–in>
  <data–out>
  <data–out type = “face-detection” priority = “1”>
    <retention>1-year</retention>
    <shared>no</shared>
    <inferred>no</inferred>
  </data-out>
  <data–out type = “image” priority = “1”>
    <retention>1-year</retention>
    <shared>no</shared>
    <inferred>yes</inferred>
  </data-out>
</ privacy-policy>
```

Як видно з таблиці 3.1, політика конфіденційності для користувача IoT визначає теги <data-in>, що вказують тип даних, які користувач хотів би отримати від власника IoT, а також дочірні елементи, що визначають сценарій використання цих даних. Ці теги <data-in> з політики конфіденційності користувача IoT будуть зіставлятися з тегами <data-out> в політиці власника IoT, оскільки остання визначає методи збору даних, прийняті власником IoT. І навпаки, теги <data-in>, зазначені в політиці власника IoT в таблиці 3.2, будуть співставлені з тегами <data-out> в політиці

конфіденційності користувача IoT, щоб гарантувати, що рівень збору даних, що виконується власником IoT, прийнятний для користувача Інтернету речей.

Прогнозована політика для кожного користувача може бути передбачена шляхом аналізу декількох сценаріїв збору даних. Використовувати прогнозування корисно, щоб уникнути громіздкого і схильного до помилок заповнення екранів налаштувань конфіденційності. Це реалізується шляхом дозволу на узгодження багаторівневого обслуговування з різними масштабами сценаріїв збору даних. Здійснюється це наступним чином.

Метод узгодження моделює взаємозв'язок між збором даних і послугою IoT в якості опції компромісу конфіденційності і утиліт, заявленої як формула 1. Це дозволяє власнику IoT пропонувати множинний вибір рівнів обслуговування, що вимірюються утилітою, відповідно до того, який числом даних користувач готовий поділитися.

Використовуються чотири аспекти форм-фактору, що впливають на переваги конфіденційності в середовищах IoT, у формулі прогнозування переваг. Ці чотири фактори зберігаються в XML-представленні політики конфіденційності як дочірні елементи всередині кожного тега <data-in> та <data-out>

Опис кожного з цих чотирьох елементів виглядає наступним чином:

1) Тип даних (t). Тип датчика, до якого здійснюється доступ, може мати різну ступінь секретності для власника цього датчика. Датчики, такі як камера або мікрофон, за своєю природою чутливі. Отже, дозвіл доступу до цих датчиків повинен здійснюватися з обережністю. У літературі є методи, що дозволяють мінімізувати ступінь порушення конфіденційності при доступі до цих датчиків, наприклад розмиття обличчя з прямої трансляції камери [17] або ретельний вибір функцій звуку, щоб уникнути побудови мови із захоплених аудіо [18]. Якщо ці

методи використовуються, вони повинні бути додані в файл XML, щоб бути частиною процесу узгодження.

2) Утримання. (R) Політика зберігання визначає терміни зберігання журналів обмінюваних даних. У додатках реального часу, де зберігання даних не потрібно, власник IoT або користувач IoT може використовувати цей фактор для примусового очищення своїх даних, залишивши цей елемент порожнім.
3) Загальний доступ (s). Будь-сторонній одержувач повинен бути зазначений в разі, якщо власник IoT або користувач IoT діляться зібраними даними для завдання IoT.
4) Прогнозований результат (i). Одержувач даних повинен вказати, чи будуть методи виведення використовуватися для отримання додаткової інформації з даних. Наприклад, дані акселерометра можуть бути використані для моніторингу фізичних вправ користувача для медичних програм, але також можуть бути використані для визначення місця розташування користувача в приміщенні.

Інші форм-фактори, які також можуть враховуватися, включають в себе причину запиту даних, розташування, мету і переваги збору даних [19]. Після вивчення вищезазначених чинників, що впливають на конфіденційність, їх використовуватимуть як частину розрахунку утиліт конфіденційності. Для оцінки утиліти конфіденційності використовується функція, яка виглядає наступним чином:

$$U = -\gamma \, .Pe\,(t,r,s,i) + B(t,r,s,i) \quad (1)$$

Пояснення позначень, що використовуються у формулі:

U позначає загальну корисність, яка буде досягнута при здійсненні обміну інформацією для запуску програми IoT.

B це перевага від обміну даними з точки зору власника даних. Для власника IoT це може бути грошовим стимулом або соціальної вигодою від надання можливості запуску додатків IoT на своїй території. Що стосується користувача IoT, то перевагою буде послуга, що надається додатком IoT.

Pe - це ступінь конфіденційності для обраної політики конфіденційності. Різні політики приведуть до різних рівнів впливу на конфіденційність в залежності від обраних форм-факторів впливу в політиці. Наприклад, вищі періоди зберігання, зазначені r для високочутливих даних, зазначені t, приведуть до більш високих значень *Pe*.

y є загальним фактором сприйняття чутливості конфіденційності. Цей фактор може варіюватися в залежності від місця розташування або контексту користувача. Значення γ множиться на *Pe* для збільшення або зменшення загального витоку конфіденційності для конкретних даних.

Умова праворуч від оператора рівності негативна, щоб узгодити її з умовою вигоди *B*. Таким чином, корисність *U* буде позитивною, якщо значення умови вигоди *B* переважує значення негативної умови впливу на конфіденційність і навпаки.

Процес узгодження політик конфіденційності

Узгодження політик відбувається за двома сцинаріями. Однофазний сценарій узгодження, показанй на рисунку 3.3.

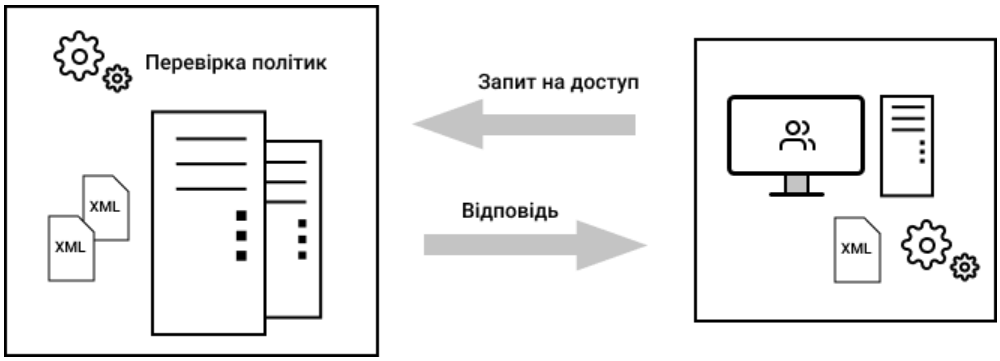

Рис. 3.2 Процес роботи методу узгодження

Сценарій починається з того, що користувач IoT відправляє запит на доступ до певного типу даних (датчика) власнику IoT. Дані беруться з елементу <data-in> для даних цього типу в політиці користувача. Отримавши запит на доступ, власник IoT перевіряє корисність запиту, підставляючи його в функцію. Припускаючи, що корисність запиту дорівнює або вище, ніж корисність елемента <data-out> на ті ж дані з політики конфіденційності власника IoT, запит приймається. Після цього власник IoT підключається до

користувача IoT і починає діяти як ретранслятор, пересилаючи інформацію датчика, отриману з інфраструктури IoT.

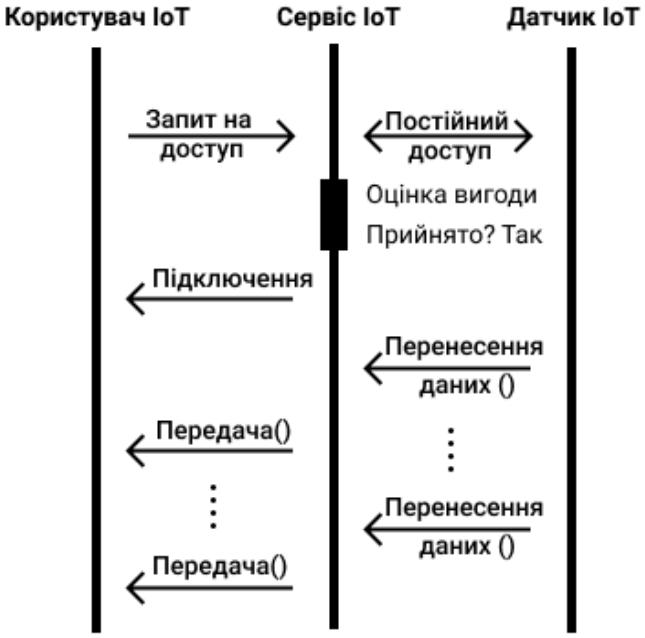

Рис. 3.3 Процес однофазного узгодження

Процес двофазного узгодження показано на рисунку 3.4. Цей сценарій починається аналогічним чином, коли користувач IoT відправляє запит на доступ до даних власнику IoT. Однак в цьому випадку корисність запиту вважається неприйнятною для власника IoT.

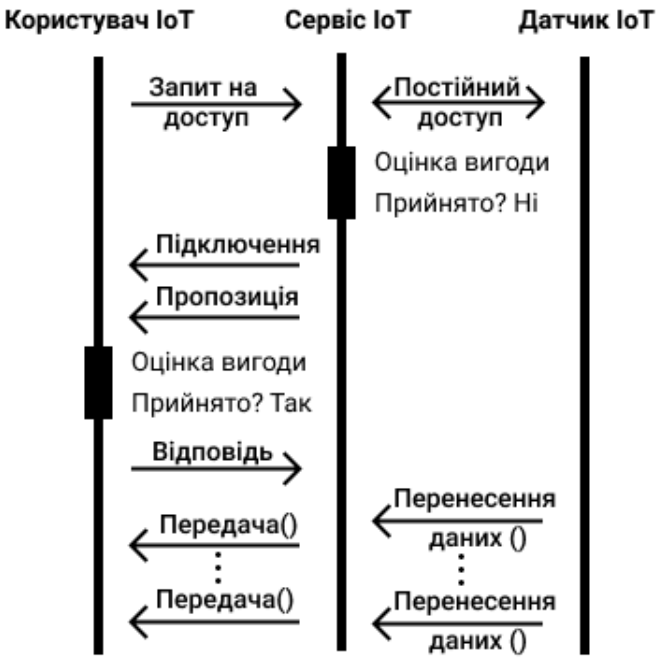

Рис. 3.4 Процес двофазного узгодження

Відповідно, власник IoT буде витягувати елемент <data-out> для запитуваного елемента даних зі своєї власної політики конфіденційності і відправлятиме його користувачу. Це відбувається тільки після підключення до користувача IoT.

Пристрій користувача IoT потім перевіряє корисність цієї пропозиції на відповідність політиці другого пріоритету, якщо вона визначена в файлі політики конфіденційності. Кожен тег <data-in "i" data-out> в політиці конфіденційності містить атрибут пріоритету, що дозволяє користувачеві IoT і власнику IoT визначати альтернативні політики, які будуть використовуватися під час узгодження.

Тільки визначення політики пріоритету 1 для елемента даних означає, що політика для цього елемента даних не підлягає обговоренню. Припускаючи, що користувач IoT прийняв альтернативну пропозицію, власник IoT почне пересилання необхідних даних, як тільки вони будуть отримані з джерела даних.

3.4 Загальна структура використання Data Bank

Рисунок 3.4 ілюструє загальну структуру, яка повинна бути спеціалізована для кожного конкретного сценарію використання Data Bank в Інтернеті речей. Діаграма класів показує загальні взаємозв'язки між компонентами в кожному рівні та між рівнями і пропонує шляхи реалізації компонентів.

Верхній прикладний шар містить чотири основні класи. UserInterface та DataSharingInterface відповідають за автентифікацію, представлення інформації та перевірку даних. Вони містять як функцію перегляду, так і контролери, що використовують модель даних, яка маніпулює даними про користувачів та сервіси.

Перегляд складається з усіх методів, які безпосередньо взаємодіють з користувачами та службами, наприклад, клас UserInterface має метод для створення графічного інтерфейсу користувача (GUI), який дозволяє

користувачам взаємодіяти з Data Bank, аналогічно, клас DataSharingInterface має метод для створення інтерфейсу програмування прикладних програм (API).

Крім того, у функції перегляду є деякі способи подання інформації у конкретних форматах (фігури, зображення, діаграми, текстовий файл тощо) для користувачів та служб. Частина контролера - це посередник, який зв'язується з функцією перегляду та моделлю даних; у нього є метод перевірки даних із перегляду та метод передачі цих даних у модель даних.

Клас User зберігає та керує даними про користувачів, а клас Service зберігає та керує даними про послуги. Прикладами даних про користувача та службу є ім’я, ідентифікатор тощо. Обидва класи User та Service також містять методи та конструктори для підтримки інкапсуляції, такі як отримання та встановлення методів для маніпулювання даними у класах. Усі класи цього шару працюють на веб-сервері.

Хмарний шар: цей шар включає класи в основному для маніпулювання даними. У Repository є метод створення схеми бази даних (зберігання, що надається Data Bank), способи створення з'єднань з БД, методи для виконання команд SQL та метод для створення API для підключення до зовнішнього сховища даних.

Через обмежену ємність пристроїв IoT клас DataRetentionPolicy керує часовою межею. Він визначає, як довго будуть зберігатися елементи даних у локальній пам'яті та сховищі. У цьому класі є метод очищення сховища.

Клас PrivacyPolicy містить детальну інформацію про політику та має методи оновлення політик та методи сповіщення користувача про зміну політики.

Клас PrivacyUtilityMechanism має методи для оптимізації конфіденційності та методи генерування політик за замовчуванням та методи генерування токенів для служб.

Клас EnforcementAccessControl має методи для оцінки авторизації послуг, методи для оцінки запитів та створення даних для відповіді на запити служб.

Клас AuditingLog має метод генерації, метод очищення та метод повернення файлів журналів, які зберігають історію транзакцій.

Рівень Data Pocket: основними класами цього шару є PolicyInterface, який має методи для створення графічного інтерфейсу, що призначений для захоплення налаштувань користувача та оновлення політик, а також інтерфейси для представлення політики у різних форматах, наприклад, текст, графік тощо) та методи перевірки даних користувачів та передачі їх до класу EnformentDataCollection, також клас EnformentDataCollection, який має методи для застосування політику збору даних, методи фільтрації даних перед передачею в хмару, визначені політикою збору даних та методи боротьби зі зберіганням та видаленням даних.

Клас ComunicationControlUnit має методи управління зв’язком між віртуальними об'єктами відповідно до протоколів. Клас Sensor містить інформацію про датчики. У ньому є методи створення та видалення віртуальних об’єктів.

Клас Category містить інформацію про категорії і аналогічно клас Action визначає дії над даними (наприклад, шифрування, дешифрування).

Клас DataItem містить інформацію про елементи даних. Category, Action та DataItem мають конструктори, а також методи отримання та налаштування. Усі класи цього шару працюють у локальному концентраторі.

Шар датчиків (фізичний об'єкт): пристрої взаємодіють з віртуальними об'єктами (клас Sensor) через драйвери (пристрої IoT можуть підключатися та спілкуватися за допомогою декількох стандартних протоколів, таких як Wi-Fi, Bluetooth, Z-хвиля, ZigBee).

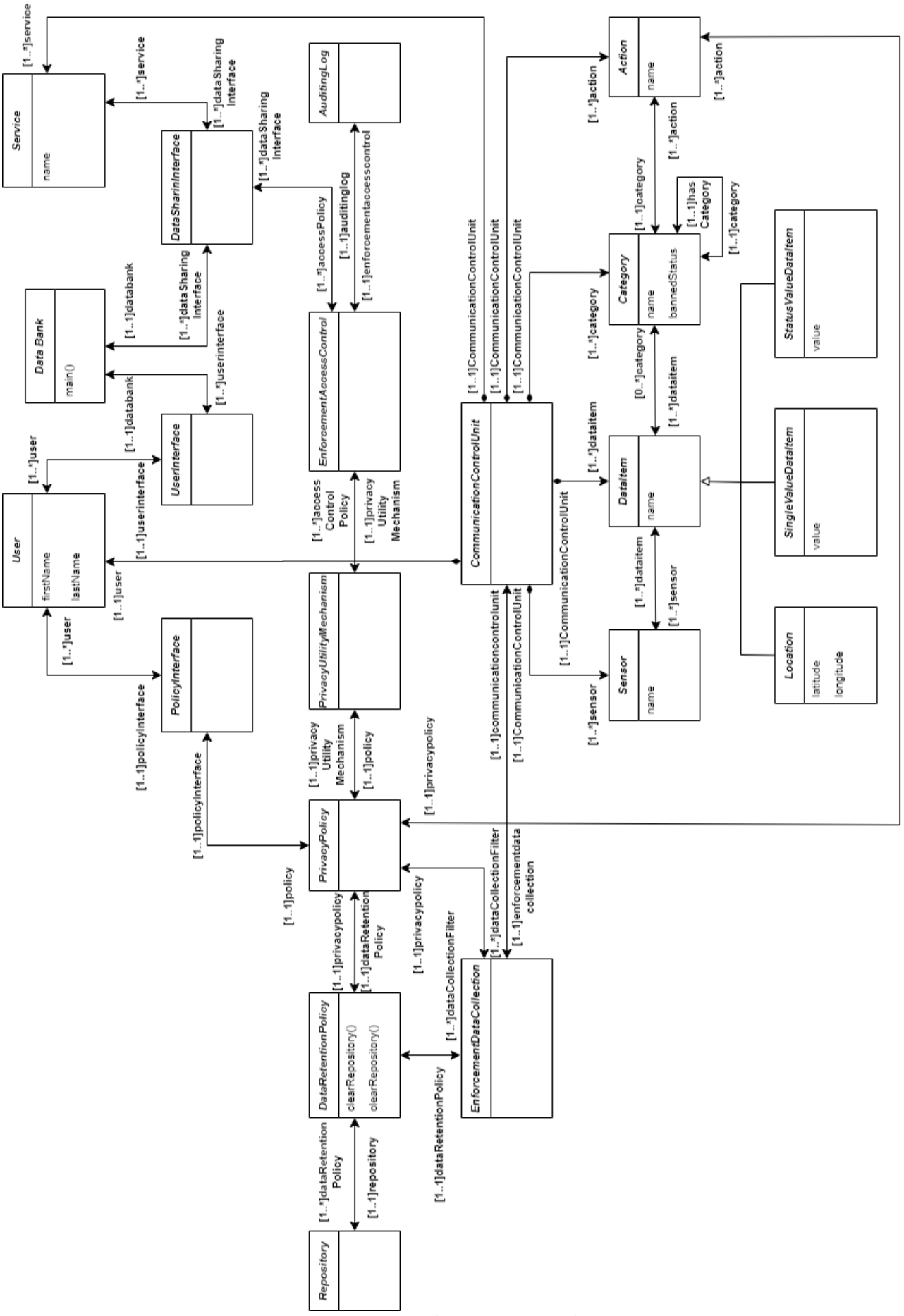

Рис. 3.5 Діаграма класів

Висновки:

1) Проведено модифікацію існуючої архітектури Інтернету речей при обміні даними, за рахунок концепції Data Bank. Показано заміну основних компонентів та специфікацію їх використання.
2) Запропонований новий спосіб управління інформацією за рахунок методу узгодження політик конфіденційності користувачів, що дає користувачам детальний контрольна їх даними. Описано ключові характеристики та сценарії використання.
3) Продемонстровано загальну структуру взаємодії компонентів в кожному рівні та зв’язок компонентів між рівнями, а також шляхи реалізації компонентів.

РОЗДІЛ 4

РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОГО МЕТОДУ ТА АНАЛІТИЧНА ОЦІНКА

4.1 Натурне моделювання запропонованого методу

Для реалізації запропонованого методу управління інформацією та підвищення рівня конфіденційнсті було використано хмарні послуги від компанії Amazon під назвою AWS.

AWS - найпоширеніша в світі комерційна платформа хмарних обчислень, яка підтримується і розвивається компанією Amazon. Надає більше 175 повнофункціональних сервісів для центрів обробки даних по всій планеті. Технології AWS базуються на серверних кластерах(фермах), що розташовані по всьому світу. Мільйони клієнтів, в тому числі стартапи, які стали лідерами за швидкістю зростання, найбільші корпорації і передові урядові установи, використовують AWS для зниження витрат, підвищення гнучкості і прискореного впровадження інновацій[20].

Для серверної частини було вибрано EC2Instance, що надається засобами AWS. На ньому було розгорнуто сервер для прийнятття запитів та їх обробки. Мовою програмування стала Python. Python – інтерпретована об’єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією.

Клієнтська сторона(у даному випадку постачальник послуг) реалізована у вигляді програми для надіслання запитів на доступ до даних та їх отримання. Програма розташована на персональному комп’ютері та реалізована також на мові Python.

Для оцінки ефективності даного методу потрібно визначити, що означає ефективність в даному контексті.

Оскільки в запропонованому методі виконується порівняння політик користувачів, що відповідають їх уподобанням конфіденційності в процесі

обміну даними з постачальниками послуг, то важливо, щоб час затрачений на обробку цих запитів не виходив за рамки оптимальних показників.

Тобто, під ефективністю вважається час обробки узгодження політик конфіденційності.

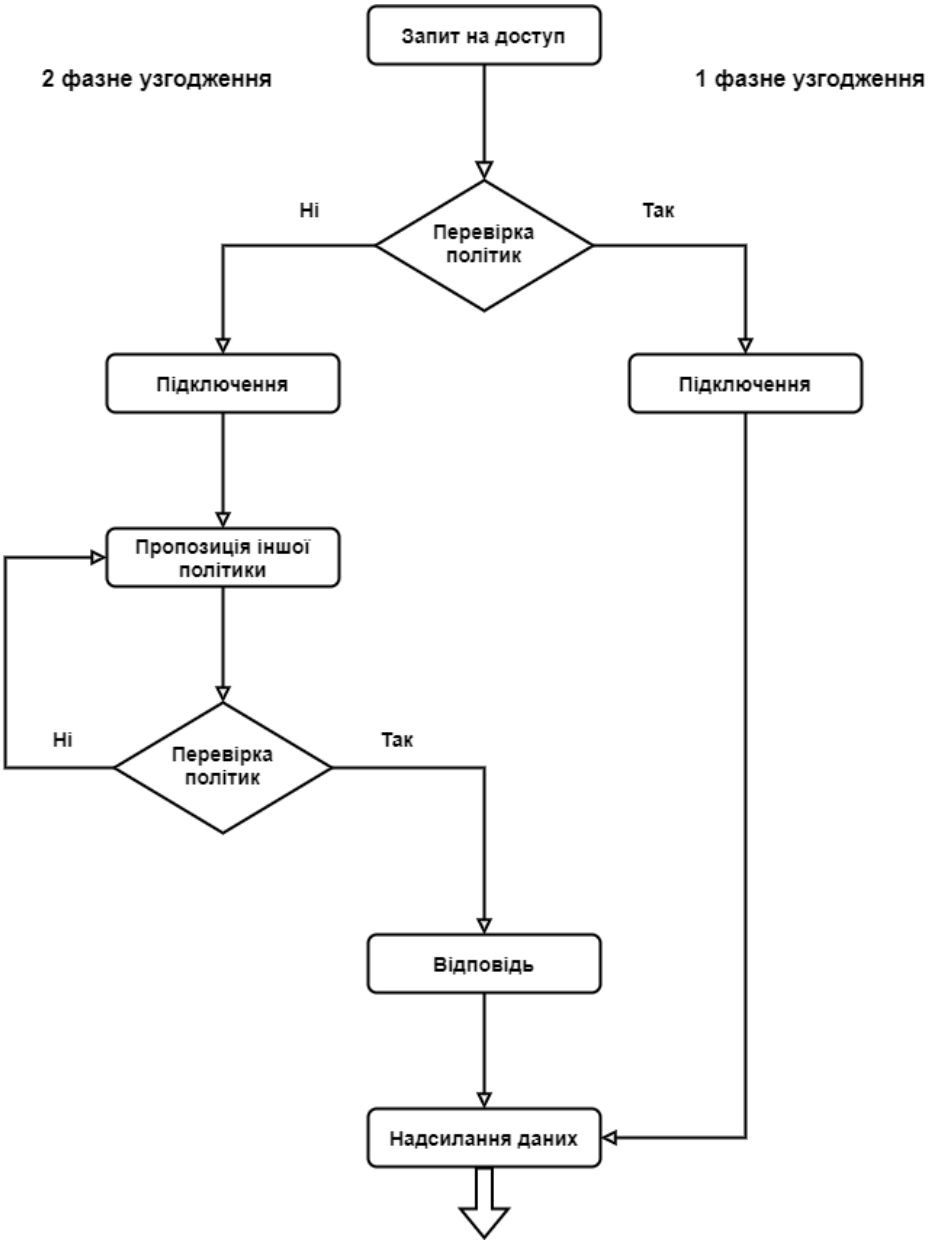

Рис. 4.1 Блок-схема алгоритму роботи узгодження

Однофазне узгодження. Постачальник послуг робит запит на доступ до даних які він хоче отримати. Запит відправляється на сервер в хмару, де містяться політики конфіденційності. В залежності від даних, що потребуються, обирається політика, що встановив власник даних і

проводиться її аналіз на відповідність. Якщо політика задовільна для обох сторін, відбувається підключення до користувача і передача даних.

Двофазне узгодження відбувається аналогічним чином до етапу перевірки політики. Якщо політику відхилено, то відбувається підключення до коритсувача і надсилається політика, що встановлена постачальником послуг. Після цього, відбувається перевірка цієї політики. Якщо політика відповідає умовам конфіденційності, що допускаються користувачем, то відправляється позитивна відповідь і починається передача даних, в іншому випадку, дія пропозиції політики - повторюється.

Рис. 4.2 Час затрачений при узгодженні політик

4.2 Аналітична оцінка методу

Можливості використання методу були продемонстровані в трьох варіантах, широко прийнятих сценаріїв. Відповідно до результатів натурного моделювання, метод узгодження політик при однофазному узгодженні працює на декілька мікросекунд повільніше за рахунок порівняння політики, від загального часу отримання даних з датчика без вимог до конфіденційності. Двофазне узгодження працює повільніше в середньму на

30 мікросекунд за рахунок відправлення іншої політики для проведення процесу визначення переваг. Для більш простішого сприйняття результатів, було проведено апроксимацію методом найменших квадратів.

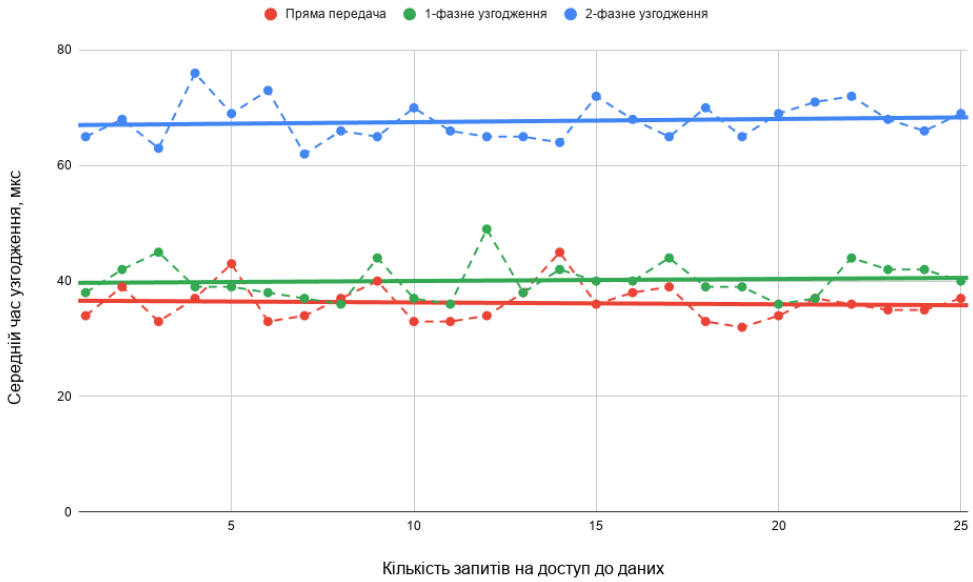

Рис. 4.3 Апроксимація часу затраченого на узгодження політик

Обидва випадки узгодження пропонують користувачам послуги з врахуванням вказаних переваг / налаштувань конфіденційності та не виходять за межі зручності користування послугами. Запропонований підхід є практичним, оскільки він узгоджує політику конфіденційності користувача з власником IoT без втручання користувача і підтримує вибір з безлічі попередньо визначених політик конфіденційності.

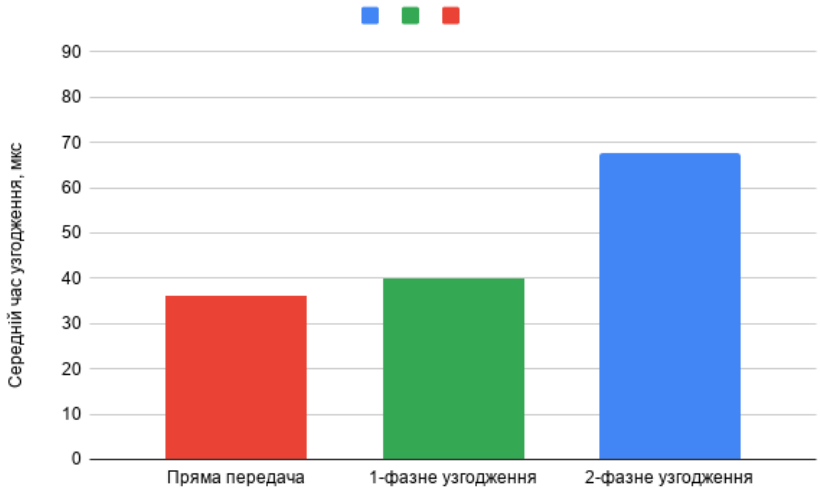

Рис. 4.4 Середній час узгодження

Нижче в таблиці 4.1 показано порівняння інструментів, представлених в загальній та модифікованій Data Bank архітектурі, з урахуванням характеристик, що впливають на процес прийняття рішень користувачів[21].

Таблиця 4.1

Характеристики, що впливають на процес прийняття рішень користувачів.

	Basic IoT Architecture	Iot Architecture with Data Bank
Чи можете користувач відстежувати, які компанії можуть збирати та використовувати особисті дані?	✓	✓
Чи може користувач контролювати, які дані будуть збиратися компаніями?	—	✓
Чи може користувач знати причину запиту даних?	✓	✓
Чи може користувач анонімізувати інформацію, яка стане доступною?	—	✓
Чи може користувач узгодити запитувані дані?	—	✓
Чи може користувач переглянути, які дані вже були оприлюднені?	✓	✓

Виходячи з результатів проведеного моделювання та даних таблиці 4.1, бачимо, що метод прямого пересилання даних працює швидше, але він задовільняє лише 50% відсотків ситуацій, що впливають на процес прийняття рішень користувача. Тому запропонований метод є кращим рішенням для обміну даними зі збереженням конфіденційності, за рахунок запропонованого методу узгодження, контролю політик збору та обміну

даними, що застосовуються в Data Bank і можливості оновлювати політики в будь-який час[22].

Висновки:

1) За рахунок проведення натурного моделювання, продемонстровано роботу запропонованого методу узгодження політик конфіденційності користувачів, описано основні кроки роботи методу.
2) Проведено аналітичну оцінку запропонованого методу узгодження в процесі обміну даними в мережі Інтернету речей, що дало змогу визначити підвищення ефективності управління персональними даними.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

1) Проведено аналіз існуючих моделей хмарних інфраструктур для постачання послуг Інтернету речей та збереженості даних в мережах, на основі чого визначено їх основні проблеми та недоліки.
2) Проаналізовано існуючі методи, що забезпечують збереження конфіденційності та управління даними в мережі Інтернету речей, що дало змогу вибрати прототип.
3) Удосконалено архітектуру мережі Інтернету речей за рахунок концепції Data Bank, що дозволяє провести процес узгодження політик конфіденційності користувачів при обміні даними.
4) Проведено натурне моделювання запропонованого рішення, що підтвердило його працездатність, а аналітична оцінка модифікованої архітектури показала підвищення ефективності управління персональною інформацією та забезпечення збереженості даних в 2 рази.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Varadharajan, V., & Bansal, S. (2016). Data Security and Privacy in the Internet of Things (IoT) Environment. Connectivity Frameworks for Smart Devices, 261–281.
2. Enabling privacy and security in Cloud of Things: Architecture, applications, security & privacy challenges [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S2210832719302819
3. Flynn D. IoT considerations — cloud services — IaaS, PaaS, SaaS, build your own [Електронний ресурс] / Des Flynn. – 2015. – Режим доступу до ресурсу: https://medium.com/lattice-research/iot-considerations-server-side-iaas-paas-saas-1f55afc03185.
4. What Is Platform-as-a-Service (PaaS)? [Електронний ресурс] – Режим доступу до ресурсу: https://www.cloudflare.com/learning/serverless/glossary/platform-as-a-service-paas/.
5. Watts S. SaaS vs PaaS vs IaaS: What's The Difference and How To Choose [Електронний ресурс] / S. Watts, R. Muhammad. – 2019. – Режим доступу до ресурсу: https://medium.com/lattice-research/iot-considerations-server-side-iaas-paas-saas-1f55afc03185.
6. Li, S., Tryfonas, T., & Li, H. (2016). The Internet of Things: a security point of view. Internet Research, 26(2), 337–359.
7. The security and privacy issues that come with the Internet of Things [Електронний ресурс] – Режим доступу до ресурсу: https://www.businessinsider.com/iot-security-privacy.
8. Jun B (2014) Make way for the internet of things. RSA conference '14, San Francisco, 24–28 Feb 2014. http://www.rsaconference.com/writable/presentations/file_upload/tech-r02-inter net-of-things-v2.pdf

9. Suo H et al (2012) Security in the internet of things: a review. In: International conference on computer science and electronics engineering (ICCSEE ’12), vol. 3, pp 648–651 IEEE, 23 Mar 2012
10. The 4 stages of an IoT architecture [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://techbeacon.com/enterprise-it/4-stages-iot-architecture.
11. H. Haddadi, H. Howard, A. Chaudhry, J. Crowcroft, A. Madhavapeddy, and R. Mortier, “Personal Data: Thinking Inside the Box,” CoRR, ArXiv e-prints, 2015.
12. M. Mun, S. Hao, N. Mishra, K. Shilton, J. Burke, D. Estrin, M. Hansen, and R. Govindan, “Personal Data Vaults: A Locus of Control for Personal Data Streams,” in Proceedings of the 6th International COnference, ser. Co-NEXT ’10. New York, NY, USA: ACM, 2010, pp. 17:1–17:12.
13. Jaimunk, J. (2019). Privacy-Preserving Cloud-IoT Architecture (Abstract). 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft).
14. K. Alanezi and S. Mishra, “A privacy negotiation mechanism for the internet of things,” in IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2018, pp. 512–519.
15. JA Stankovic. 2014. Research directions for the internet of things. IEEE Internet of Things Journal 1, 1 (2014), 3–9.
16. L. Cranor. 2002. Web privacy with P3P. " O’Reilly Media, Inc."
17. A. Das, M. Degeling, X. Wang, J. Wang, N. Sadeh, and M. Satyanarayanan. 2017. Assisting Users in a World Full of Cameras: A Privacy-Aware Infrastructure for Computer Vision Applications. In CVPRW. IEEE, 1387–1396.

18. D. Wyatt, T. Choudhury, and J. Bilmes. 2007. Conversation detection and speaker segmentation in privacy-sensitive situated speech data. In Interspeech.
19. PE Naeini, S. Bhagavatula, H. Habib, M. Degeling, L. Bauer, L. Cranor, and N. Sadeh. 2017. Privacy Expectations and Preferences in an IoT World. In SOUPS.
20. What is AWS [Електронний ресурс] – Режим доступу до ресурсу: https://aws.amazon.com/what-is-aws/.
21. Пороло Є. Удосконалена архітектура мережі для хмарного Інтернету речей / Є. Пороло, В. Курдеча // ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ / Є. Пороло, В. Курдеча. – м. Київ, Україна: ISSN (print) 2663-502X, ISSN (online) 2664-3057, 2020. – С. 219–221.
22. Пороло Є. Застосування концепції Data Bank в мережі хмарного IoT / Євгеній Пороло // ПЕРСПЕКТИВИ РОЗВИТКУ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ ТА СИСТЕМ / Євгеній Пороло. – м. Київ, Україна: ISSN (print) 2663-502X, ISSN (online) 2664-3057, 2020. – С. 368.