**МАГІСТЕРСЬКА РОБОТА**

МР. ІПм - 08.00.00.000 ПЗ

**Група ІПм-24-1**

**Волошин Павло**

**2025**

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

**Волошин Павло Юрійович**

(прізвище, ім'я, по батькові)

УДК 004.9

(індекс)

# МАГІСТЕРСЬКА РОБОТА

## Онтологічні моделі контролю доступу в хмарних засобах соціального нетворкінгу

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Волошин П.Ю.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Бандура Вікторія Валеріївна, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

**Допущено до захисту**

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2025

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ:**
Зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ 04 ” вересня 2025 р.

# ЗАВДАННЯ

## НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

**Волошину Павлу Юрійовичу**

(прізвище, ім’я, по-батькові)

**1. Тема магістерської роботи “ Онтологічні моделі контролю доступу в хмарних засобах соціального нетворкінгу”**

керівник проекту (роботи) Бандура Вікторія Валеріївна, к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 05 ” листопада 2025 р. № 695/7

**2. Строк подання студентом проекту (роботи)** 15 грудня 2025 р.

**3. Вихідні дані до проекту (роботи)** Теоретичні концепції та формальні моделі побудови інформаційних технологій онтологічних представлень в хмарних інфраструктурах

**4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)**

1. Дослідження ПО застосування онтологій контролю доступу в соціальному нетворкінгу
2. Онтологічні моделі контролю доступу в соціальному нетворкінгу та хмарних системах
3. Імплементація онтологічних моделей контролю доступу в засобах соціального нетворкінгу
4. Розробка архітектури та робочого процесу системи контролю доступу

**5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)**

1. Принципи моделі контролю доступу на основі ролей (RDAC) (рис. 1.1)
2. Модель контролю доступу на основі атрибутів (ABAC) (рис. 1.2)
3. Архітектура акторів XACML (рис. 1.3)
4. Граф делегування XACML (рис. 1.4)
5. Онтологія контролю доступу (рис. 1.5)

**6. Консультанти розділів проекту (роботи)**

| Розділ | Консультант | Підпис, дата |
|---|---|---|
| Перевірка на плагіат | доц., к.т.н. Вовк Р.Б. | |
| | | |

**7. Дата видачі завдання** 04 вересня 2025 р.

**Керівник** _____
(підпис)

**Завдання прийняв до виконання**_____
(підпис)

# КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назви етапів магістерської роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Підбір і вивчення літератури по темі магістерської роботи | 15.09.2025 | виконано |
| 2 | Дослідження ПО застосування онтологій контролю доступу в соціальному нетворкінгу | 29.09.2025 | виконано |
| 3 | Онтологічні моделі контролю доступу в соціальному нетворкінгу та хмарних системах | 15.10.2025 | виконано |
| 4 | Імплементація онтологічних моделей контролю доступу в засобах соціального нетворкінгу | 08.11.2025 | виконано |
| 5 | Розробка архітектури та робочого процесу системи контролю доступу | 20.11.2025 | виконано |
| 6 | Реалізація функціональності запропонованої інформаційної технології | 01.12.2025 | виконано |
| 7 | Затвердження пояснювальної записки роботи завідувачем кафедри | 15.12.2025 | виконано |

**Студент – магістр** _____
(підпис)

**Керівник роботи** _____
(підпис)

# АНОТАЦІЯ

**Магістерська робота:** 82 с., 29 рис., 1 табл., 39 джерел.

**Тема:** Онтологічні моделі контролю доступу в хмарних засобах соціального нетворкінгу

**Мета магістерської роботи** - розробка та обґрунтування онтологічних моделей контролю доступу, що забезпечують гнучке, масштабоване та семантично орієнтоване управління доступом у хмарних середовищах соціального нетворкінгу.

**Об'єкт дослідження** - процеси контролю доступу до інформаційних ресурсів у хмарних середовищах соціального нетворкінгу.

**Предмет дослідження** - онтологічні моделі контролю доступу та їх застосування для управління конфіденційністю, делегуванням прав і виконанням політик доступу в соціальних мережах і хмарних обчислювальних системах.

**Результати дослідження**

В роботі розроблена політика контролю доступу демонструє здатність забезпечувати баланс між захистом конфіденційності, динамічністю взаємодії користувачів та гнучкістю управління ресурсами у хмарних соціальних платформах..

**Висновок**

У межах дослідження також було розроблено онтологічний підхід до контролю доступу на основі вмісту у соціальних мережах. Це рішення забезпечує врахування контексту повідомлень, специфіки міжкористувацьких відносин і дозволяє гнучко адаптувати політики доступу.

**ОНТОЛОГІЯ, КОНТРОЛЬ ДОСТУПУ, ХМАРНІ ОБЧИСЛЕННЯ, СОЦІАЛЬНИЙ НЕТВОРКІНГ, ДЕЛЕГУВАННЯ ПРАВ, ПОЛІТИКИ ДОСТУПУ, ІНФОРМАЦІЙНА БЕЗПЕКА, СЕМАНТИЧНІ МОДЕЛІ.**

## ABSTRACT

**Master Thesis:** 82 pp., 29 fig., 1 tab., 39 sources.

**Topic:** Ontological access control models in cloud social networking tools

**The purpose of the master's thesis** is to develop and substantiate ontological access control models that provide flexible, scalable and semantically oriented access control in cloud social networking environments.

**The object of the study** is the processes of access control to information resources in cloud social networking environments.

**The subject of the study** is ontological access control models and their application for managing confidentiality, delegation of rights and enforcement of access policies in social networks and cloud computing systems.

**Research results**

The access control policy developed in the work demonstrates the ability to provide protection between the balance of confidentiality, dynamism of user interaction and flexibility of resource management in cloud social platforms.

**Conclusion**

An ontological approach to content-based access control in social networks was also developed within the framework of the study. This solution takes into account the context of messages, the specifics of inter-user relationships and allows for flexible adaptation of the access policy.



**ONTOLOGY, ACCESS CONTROL, CLOUD COMPUTING, SOCIAL NETWORK, DELEGATION OF RIGHTS, ACCESS POLICIES, INFORMATION SECURITY, SEMANTIC MODELS**

**ЗМІСТ**

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ОСМ - онлайн-соціальна мережа

ABAC - Attribute-Based Access Control

RBAC - Role-Based Access Control

PEP - Policy Enforcement Point

PDP - Policy Decision Point

PIP - Policy Information Point

PAP - Policy Administration Point

XACML - eXtensible Access Control Markup Language

CSP - Cloud service providers

# ВСТУП

**Актуальність теми.**

Стрімкий розвиток інформаційних технологій та поширення хмарних обчислень зумовили кардинальні зміни у способах зберігання, обробки та обміну даними. Соціальні мережі, які стали невід'ємною складовою цифрового суспільства, функціонують у середовищах із високим рівнем динамічності, масштабованості та розподіленості. У таких умовах особливої актуальності набуває проблема захисту інформації та управління доступом до ресурсів.

Традиційні моделі контролю доступу, такі як DAC, MAC та RBAC, продемонстрували свою ефективність у централізованих системах, проте вони виявилися недостатньо гнучкими для динамічних середовищ соціального нетворкінгу та хмарних платформ. Виклики, пов'язані з конфіденційністю, делегуванням прав та забезпеченням сумісності політик у багатокористувацьких системах, потребують нових підходів до управління доступом.

Онтологічні моделі, що забезпечують формалізацію знань про сутності та їхні взаємозв'язки, відкривають нові можливості для розробки інтелектуальних систем контролю доступу. Використання онтологій дозволяє досягти семантичної прозорості, гнучкості та масштабованості у процесах управління доступом до інформаційних ресурсів у соціальних мережах і хмарних середовищах.

Актуальність дослідження зумовлена зростанням обсягів інформації, що циркулює у хмарних та соціальних системах, а також збільшенням кількості загроз конфіденційності. Класичні моделі управління доступом не забезпечують належного рівня гнучкості, особливо в умовах динамічної зміни користувацьких ролей, міжкористувацьких відносин і контекстів взаємодії.

Особливого значення набувають проблеми делегування прав у хмарних середовищах, де необхідно швидко та безпечно надавати доступ третім особам із можливістю подальшого контролю та відкликання повноважень. Крім того, існуючі моделі недостатньо враховують семантичний зміст даних, що призводить до конфліктів політик і надмірних обмежень у доступі.

У цьому контексті розробка онтологічних моделей контролю доступу постає як актуальний науково-практичний напрям, що поєднує у собі гнучкість атрибутивного підходу (ABAC) та переваги семантичних технологій.

**Метою магістерської роботи** є розробка та обґрунтування онтологічних моделей контролю доступу, що забезпечують гнучке, масштабоване та семантично орієнтоване управління доступом у хмарних середовищах соціального нетворкінгу.

**Об'єкт дослідження** - процеси контролю доступу до інформаційних ресурсів у хмарних середовищах соціального нетворкінгу.

**Предмет дослідження** - онтологічні моделі контролю доступу та їх застосування для управління конфіденційністю, делегуванням прав і виконанням політик доступу в соціальних мережах і хмарних обчислювальних системах.

**Завдання дослідження:**

- Провести аналіз проблем управління доступом та конфіденційністю у сучасних розподілених середовищах.

- Оцінити можливості класичних і сучасних моделей контролю доступу у контексті хмарних систем і соціальних мереж.

- Дослідити методи делегування прав доступу та механізми їх реалізації.

- Розробити онтологічну модель контролю доступу, що поєднує атрибутивний підхід і семантичні технології.

- Розробити механізми вирішення конфліктів політик та алгоритм об'єднання правил доступу.

- Реалізувати архітектуру системи контролю доступу для соціальних мереж із використанням онтологічного підходу.

**Методи дослідження**

В роботі використано наступні методи: аналіз та синтез – для вивчення проблематики управління доступом та узагальнення існуючих підходів; методи онтологічної інженерії – для формалізації знань, опису сутностей та побудови онтологій; vетоди моделювання – для побудови онтологічних моделей контролю доступу та делегування; алгоритмічні методи – для розробки механізмів верифікації, об'єднання політик та вирішення конфліктів.

**Наукова новизна отриманих результатів**

Розроблено онтологічну модель контролю доступу, яка поєднує атрибутивний підхід (ABAC) із семантичним представленням політик і запропоновано універсальне рішення для делегування прав у хмарних середовищах на основі онтологій.

**Практичне застосування результатів**

Реалізовано архітектуру системи контролю доступу для соціального нетворкінгу, яка враховує контекст повідомлень та взаємодію користувачів. Отримані результати можуть бути використані при розробці систем контролю доступу у хмарних середовищах; платформ соціального нетворкінгу для забезпечення захисту конфіденційності; інтелектуальних рішень для управління політиками доступу та делегування прав.

**Структура магістерської роботи.** Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 101 сторінку, і містить 20 рисунків, 5 таблиць, список використаних джерел із 54 найменувань.

# РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ОНТОЛОГІЧНИХ МОДЕЛЕЙ КОНТРОЛЮ ДОСТУПУ В СОЦІАЛЬНОМУ НЕТВОРКІНГУ

## 1.1. Проблеми управління доступом та конфіденційністю в сучасних розподілених середовищах

Поява Інтернету спричинила трансформацію підходів до обміну інформацією та комп'ютерними ресурсами. Вона сприяла формуванню відкритих розподілених середовищ, що забезпечують універсальний доступ до спільно використовуваних ресурсів (зокрема, апаратного та програмного забезпечення) для гетерогенних користувачів, від корпорацій до приватних осіб, з мінімальним адміністративним навантаженням. Така модель значно підвищила продуктивність як окремих користувачів, так і організацій.

Зокрема, стрімкий розвиток онлайн-соціальних мереж (ОСМ) та хмарних обчислень [2] призвів до залучення мільярдів користувачів, які активно обмінюються ресурсами та делегують обчислення. ОСМ слугують платформами для соціальної взаємодії, що ґрунтується на обміні цифровою інформацією, такою як фотографії, відео та текстові дані [3]. Хмарні обчислення, у свою чергу, здобули популярність у бізнес-середовищі та серед кінцевих користувачів завдяки низькій вартості обслуговування та універсальному доступу до ресурсів, що надаються за моделлю «як послуга» [4].

### *1.1.1. Загрози конфіденційності та роль контролю доступу*

Зростаюче використання відкритих середовищ для обміну даними значно збільшило ризики для конфіденційності користувачів. Чутливі електронні дані можуть бути агреговані, проаналізовані та повторно розповсюджені третіми сторонами, зокрема брокерами даних [4], що потенційно призводить до порушення конфіденційності осіб [6]. Таким

чином, забезпечення права на конфіденційність стало критичним завданням для контролерів даних, організацій та кінцевих користувачів [5], а захист конфіденційності у цих середовищах визнано складною науковою проблемою [6]. Один з ключових підходів до мінімізації цих ризиків полягає у впровадженні управління контролем доступу. Цей механізм регулює доступ суб'єктів до потенційно чутливих ресурсів на основі їхніх облікових даних, типу ресурсу та вимог конфіденційності власника даних [7]. У цьому контексті також важливим є делегування контролю доступу, яке дозволяє передавати права доступу іншим суб'єктам, підвищуючи гнучкість управління та знижуючи адміністративне навантаження у великомасштабних системах.

### *1.1.2. Виклики для традиційних моделей управління доступом*

Ефективне управління доступом є критично важливим для великих та динамічних середовищ. Хоча існуюча наукова література пропонує низку рішень [8 - 10], що базуються на класичних моделях (таких як DAC, MAC, RBAC або ABAC), більшість з них орієнтовані на заздалегідь визначені та керовані вручну правила. Ці підходи є адекватними для закритих, статичних середовищ з обмеженою кількістю суб'єктів і ресурсів, де ручне адміністрування є можливим.

Однак, застосування таких рішень у відкритих, динамічних сценаріях, як ОСМ чи хмарні технології, є непрактичним через низку чинників:

1. Масштабність.

Необхідність керувати величезною кількістю суб'єктів. Наприклад, Google Drive обслуговує мільярди користувачів, кожен з яких володіє різними типами ресурсів.

2. Гетерогенність.

Залучені суб'єкти мають різноманітні вимоги до конфіденційності. Наприклад, політики для корпоративного клієнта хмарного сервісу суттєво відрізнятимуться від політик для індивідуального користувача.

3. Динамічність.

Відкритість середовища призводить до частих та швидких змін у вимогах до конфіденційності, що залежать від типу послуги та користувачів.

Крім того, багато існуючих рішень виявляються неефективними для кінцевих користувачів через жорсткість механізмів [12] та відсутність у користувачів спеціалізованих технічних знань у сфері конфіденційності даних [11]. Важливо також відзначити, що продуктивність більшості механізмів управління доступом пропорційно знижується зі збільшенням кількості суб'єктів, що робить їх проблематичними для великомасштабних середовищ. Це вказує на нагальну потребу в розробці універсальних та масштабованих рішень, здатних подолати зазначені обмеження.

## 1.2. Основне завдання магістерського дослідження

Дана робота присвячена вирішенню проблем конфіденційності, що виникають при обміні даними та ресурсами у відкритих середовищах. Запропоновані механізми контролю доступу спрямовані на подолання існуючих обмежень. Відповідно до цього, основними цілями дослідження є:

1. Аналіз сучасного стану управління контролем доступу.

Провести глибокий аналіз існуючих підходів до управління контролем доступу, з особливим акцентом на їх застосовність у великомасштабних та динамічних відкритих середовищах. Дослідити механізми формального моделювання суб'єктів, залучених у процес контролю доступу, як засіб автоматизації та зниження адміністративного навантаження, пов'язаного з ручним керуванням політиками.

2. Розробка універсального механізму контролю доступу.

Запропонувати універсальний механізм, який забезпечує моделювання суб'єктів, що беруть участь у контролі доступу, та їх взаємозв'язків. Цей механізм має бути легко адаптованим до специфіки відкритих середовищ. Метою є спрощення та прискорення процесу визначення правил доступу у

складних сценаріях, а також підвищення сумісності між гетерогенними системами.

3. Проєктування механізму делегування прав.

Розробити ефективний механізм для виконання делегування, відкликання та перевірки прав доступу в розподілених відкритих середовищах. Цей механізм повинен забезпечувати автоматизацію цих процесів для підвищення гнучкості та надійності системи.

4. Створення автоматизованого механізму контролю доступу на основі вмісту.

Оскільки ризики розголошення ресурсів часто пов'язані з їхнім потенційно чутливим вмістом, ми прагнемо запропонувати автоматичний механізм контролю доступу, який враховує вміст та контекст конфіденційності. Цей механізм буде спеціально адаптований для захисту чутливих ресурсів, що публікуються у відкритих соціальних середовищах, роблячи процес захисту прозорим для користувачів та мінімізуючи необхідність ручного визначення правил доступу.

## 1.3. Засади та методи захисту конфіденційності

Конфіденційність — це право та практика, що дозволяє індивіду або групі визначати межі доступу до своєї особистої інформації та ресурсів, керуючи їх публічним розголошенням відповідно до обраних вимог. У науковій літературі поняття конфіденційності розглядається через призму декількох ключових аспектів [8]:

1) доступ до особистої інформації,
2) право на усамітнення,
3) контроль над використанням персональних даних іншими суб'єктами,
4) можливість приховування інформації від сторонніх.

Право на конфіденційність закріплено в міжнародному та національному законодавстві. Зокрема, Стаття 12 Загальної декларації прав людини визнає право на конфіденційність як основоположне. Численні сучасні законодавчі акти, такі як Регламент ЄС про захист даних (GDPR), Закони про конфіденційність медичних даних  та Закон про переносність і підзвітність медичного страхування (HIPAA), встановлюють правові рамки для захисту приватної інформації.

*1.3.1. Основні методи забезпечення конфіденційності*

Існує три основні методології для забезпечення права на конфіденційність:

- Криптографія. Цей підхід використовує математичні алгоритми для маскування цифрових даних, забезпечуючи їх конфіденційність. Доступ до відкритого тексту надається лише авторизованим суб'єктам, які володіють відповідними криптографічними ключами. Існують два основні типи криптографії:

- Симетрична криптографія, де один секретний ключ використовується для шифрування і дешифрування даних обома сторонами.

- Асиметрична криптографія, яка використовує пару ключів — публічний (для шифрування) і приватний (для дешифрування).

Головним викликом у криптографічних системах є ефективне управління ключами, оскільки їх компрометація може призвести до повного розголошення даних.

- Анонімізація даних. Ця методологія спрямована на видалення або модифікацію особистої ідентифікаційної інформації (PII) з наборів даних перед їх публічним розповсюдженням [13]. На відміну від криптографії, анонімізація є незворотним процесом, що усуває потребу в управлінні секретними ключами. Хоча анонімізовані дані зберігають певну аналітичну корисність, вони не можуть забезпечити абсолютний захист конфіденційності, оскільки існує ризик деанонімізації. Дослідження в цій

галузі зосереджені на статистичному контролі розголошення та публікації даних із гарантіями конфіденційності.

- Управління контролем доступу (УКД). Цей механізм регулює авторизацію суб'єктів на доступ до спільних чутливих ресурсів на основі їхнього рівня довіри. Укд визначає права користувачів щодо доступу до цифрових ресурсів. На відміну від криптографії та анонімізації, які модифікують самі дані, УКД зберігає дані в оригінальному вигляді, просто обмежуючи доступ до них. Це робить його особливо зручним у середовищах з центральними органами, де користувачі проходять автентифікацію. Важливо, що УКД здатний керувати доступом не лише до цифрового вмісту, а й до обчислювальних ресурсів, що робить його ключовим компонентом більшості сучасних інформаційних технологій, від бізнес-додатків до хмарних сервісів.

### 1.4. Управління контролем доступу: Моделі та архітектура

Управління контролем доступу (УКД) є ключовим механізмом, що регулює надання або відмову в доступі до певного ресурсу на основі трьох основних компонентів: ідентифікації користувача, типу ресурсу, та вимог конфіденційності власника ресурсу [16]. Системи контролю доступу функціонують на базі трьох складових:

1. Політика контролю доступу.

Формалізує правила, що визначають, як слід управляти доступом та хто має право на нього.

2. Механізм.

Здійснює перевірку запитів на доступ та, відповідно до політики, надає або відмовляє в ньому.

3. Модель.

Є формальним представленням політики безпеки, що описує процедури обробки запитів на доступ до системних ресурсів.

### *1.4.1. Класичні та сучасні моделі контролю доступу*

Традиційні моделі УКД, такі як дискреційний контроль доступу (DAC) та обов'язковий контроль доступу (MAC), становлять основу для багатьох систем. У моделі DAC рішення про доступ делегується власнику ресурсу, а доступ обмежується на основі ідентичності користувачів. На противагу цьому, в моделі MAC рішення про авторизацію приймається центральним органом на основі заздалегідь визначених правил.

Однак, у великомасштабних та динамічних відкритих онлайн-середовищах соціального нетворкінгу, де кількість гетерогенних користувачів та ресурсів значно зростає (наприклад, у соціальних мережах), класичні моделі стають неефективними. Для вирішення цих проблем були розроблені універсальніші та масштабованіші моделі: контроль доступу на основі ролей (RBAC) та контроль доступу на основі атрибутів (ABAC). Ці моделі є більш гнучкими та зручними для адміністрування, оскільки вони не покладаються на індивідуальні ідентифікатори користувачів, а використовують абстрактніші сутності.

### *1.4.2. Огляд моделей RBAC та ABAC*

У моделі RBAC права доступу управляються через ролі, які призначаються користувачам, що спрощує адміністрування у великих організаціях. Виділяють чотири основні підмоделі RBAC:

- RBAC0 (плоска): Базова модель, що встановлює зв'язки між користувачами, ролями та дозволами.

- RBAC1 (ієрархічна): Додає до RBAC0 ієрархію ролей, що відображає організаційну структуру та дозволяє успадкування привілеїв.

- RBAC2 (обмежена): Вводить обмеження та умови до ролей для реалізації принципу поділу обов'язків.

- RBAC3 (симетрична): Об'єднує ієрархію та обмеження, надаючи комплексний підхід до управління правами доступу.

Модель RBAC (Role-Based Access Control) базується на техніці відображення роль-до-об'єкта, що дозволяє користувачам отримувати доступ до ресурсів (об'єктів) на основі призначених їм ролей. Цей підхід реалізується шляхом прив'язки дозволів для певних ресурсів до конкретних ролей. Потім ці ролі призначаються користувачам, що надає їм доступ до відповідних ресурсів (рис. 1.1).

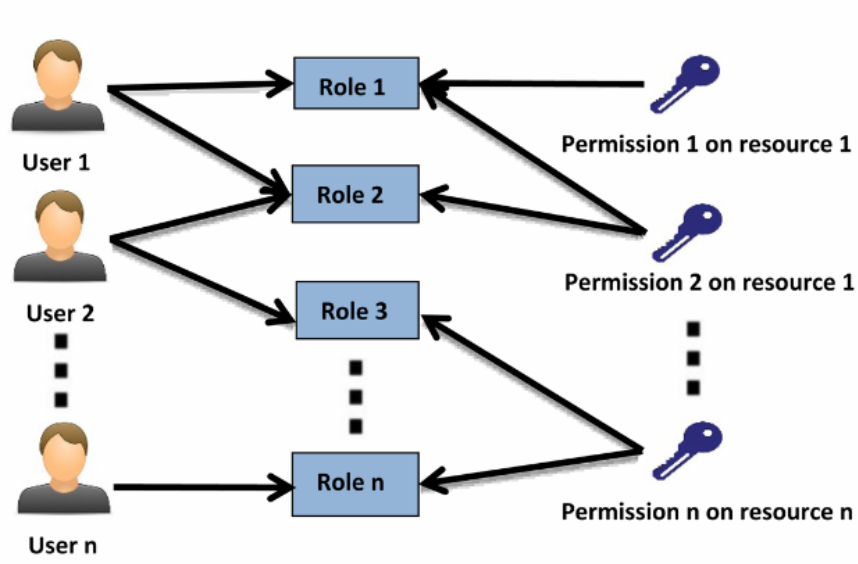


Рис. 1.1. Принципи моделі контролю доступу на основі ролей (RDAC)

Як показано на рис. 1.1, модель RBAC складається з трьох основних сутностей:

- Користувач - sндивід, що має доступ до ресурсів організації. Кожен користувач має свій унікальний ідентифікатор.

- Роль - функція або посада в межах організації. Зазвичай, кожна роль має опис повноважень і відповідальності, які вона надає.

- Дозвіл (Permission) - схвалення певного типу доступу до об'єкта. Також використовуються еквівалентні терміни, такі як привілей або право доступу.

Модель ABAC базується на атрибутах суб'єктів, об'єктів та оточення, що робить її надзвичайно гнучкою. Рішення про доступ приймаються динамічно, шляхом перевірки атрибутів користувача та ресурсу.

Архітектура ABAC зазвичай включає два ключові компоненти:

- Точка виконання політики (PEP) - приймає запити на доступ, звертається до PDP та реалізує отримане рішення.

- Точка прийняття рішень щодо політики (PDP) - аналізує атрибути суб'єкта та об'єкта, та на їх основі приймає рішення про авторизацію.

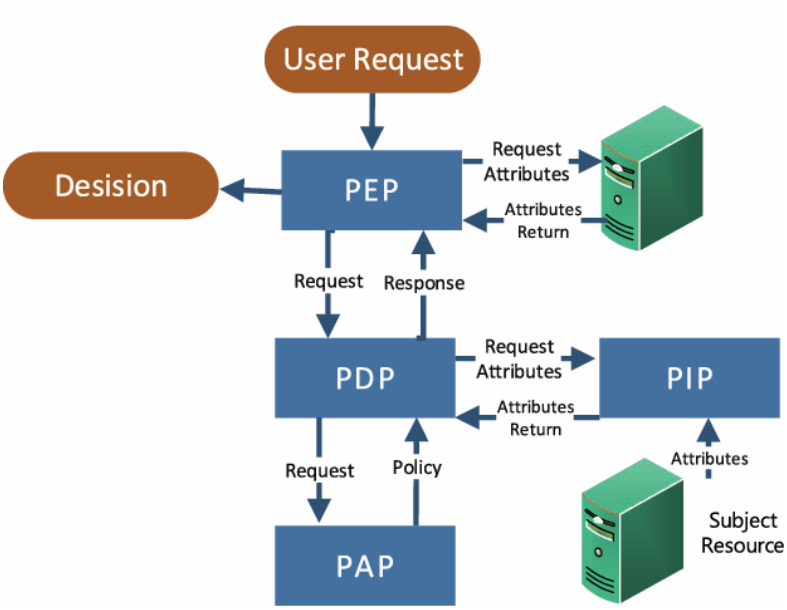


Рис. 1.2. Модель контролю доступу на основі атрибутів (ABAC)

На рисунку 1.2 показано модель контролю доступу на основі атрибутів (ABAC), яка визначає права доступу користувачів на основі їхніх атрибутів, а також атрибутів ресурсів та середовища. Ця модель складається з чотирьох основних компонентів, що взаємодіють для прийняття рішення про доступ.

Розглянемо компоненти архітектури.

PEP (Policy Enforcement Point) - точка виконання політики. Це перша ланка в процесі. PEP отримує "Запит користувача" на доступ до ресурсу. Він не приймає рішення самостійно, а перенаправляє запит до PDP для аналізу. Також PEP може запитувати атрибути користувача (суб’єкта) та ресурсу безпосередньо у відповідних систем (зображено як "сервер" у верхній частині схеми), щоб потім передати їх до PDP. Після отримання рішення, PEP виконує його, надаючи або відхиляючи доступ.

PDP (Policy Decision Point) - точка прийняття рішень щодо політики. Це "мозок" системи, що відповідає за прийняття рішення про авторизацію. PDP отримує запит від PEP і звертається до PIP для отримання всіх необхідних атрибутів (про користувача, ресурс, контекст). На основі цих атрибутів, а також правил, отриманих від PAP, PDP аналізує запит і формує відповідь ("Рішення"), яку повертає до PEP.

PIP (Policy Information Point) - точка отримання інформації про політику. PIP служить джерелом усіх необхідних даних (атрибутів). Коли PDP запитує інформацію, PIP звертається до відповідних джерел (бази даних, каталоги користувачів, системи управління ресурсами), щоб зібрати атрибути суб'єктів і ресурсів.

PAP (Policy Administration Point) - точка адміністрування політики. PAP — це місце, де адміністратори визначають і зберігають правила та політики доступу. PDP звертається до PAP, щоб отримати відповідні правила, які будуть використані для оцінки запиту на доступ.

Процес авторизації в цій архітектурі відбувається наступним чином:

1. Користувач ініціює "Запит користувача" на доступ до ресурсу.
2. Запит надходить до PEP, який перенаправляє його до PDP.
3. PDP запитує необхідні атрибути у PIP.
4. PIP збирає атрибути з відповідних систем і повертає їх до PDP.
5. Одночасно, PDP отримує правила політики від PAP.

Маючи правила та атрибути, PDP приймає "Рішення" (дозволити/заборонити) та повертає його до PEP. PEP виконує отримане рішення, надаючи або блокуючи доступ користувачеві.

Загалом, ABAC перевершує RBAC за критеріями масштабованості, гнучкості та зручності управління. Ця перевага пояснюється відсутністю необхідності ручного керування численними ролями, оскільки одна політика може бути застосована до безлічі користувачів через спільні атрибути.

Незважаючи на значні зусилля дослідників, спрямовані на вдосконалення моделей (наприклад, класифікація ресурсів або використання

списків користувачів), вони залишаються недостатньо ефективними у великих та складних середовищах. Це обумовлено зростанням вимог до конфіденційності, які існуючі рішення не можуть задовольнити та значним адміністративним навантаженням, пов'язаним з ручним визначенням та управлінням правилами [15].

## 1.5. Делегування прав доступу в розподілених системах соціального нетворкінгу

Крім стандартних завдань контролю доступу, делегування прав доступу є важливим аспектом управління. Це механізм, що дозволяє користувачеві передавати свої привілеї доступу іншим суб'єктам. Такий підхід має подвійну цінність: він підвищує ефективність управління у розподілених середовищах (наприклад, у хмарі або розподілених базах даних) та суттєво знижує адміністративне навантаження на власників ресурсів.

### *1.5.1. Концептуальні засади делегування*

У процесі делегування розрізняють дві основні сторони:

- Делегуючий - користувач, який володіє правами доступу до певного ресурсу та має законне право передати ці привілеї.

- Делегований - одержувач прав доступу.

Під час делегування делегуючий визначає обмеження та обсяг переданих привілеїв, що обмежує делегованого у їх використанні. Ці умови можуть бути формалізовані у вигляді ролей або політик, залежно від моделі контролю доступу. Делегування є проксі-процесом, який дозволяє делегованому виконувати певні дії від імені делегуючого [20].

Для ефективної реалізації цього процесу необхідний надійний механізм, здатний відповідати на такі запити:

- Чи володіє користувач правами, які він намагається використати?

- Чи мав делегуючий право надавати ці права?

- Чи мав делегуючий привілеї, які він делегував?

*1.5.2. XACML як стандарт для контролю доступу та делегування*

Для підтримки таких механізмів було запропоновано низку стандартів, серед яких XACML (eXtensible Access Control Markup Language) є найпоширенішим, особливо в частині профілю делегування XACML.

Переваги XACML включають:

- Можливість опису та аналізу політик, незалежно від конкретного середовища.

- Зниження адміністративного навантаження, оскільки політики описуються лише один раз.

- Здатність адаптуватися до динамічних змін у вимогах політики.

- Можливість застосування однієї політики до кількох суб'єктів одночасно.

XACML визначає мову політики контролю доступу на основі XML, що дозволяє інтуїтивно оцінювати запити на доступ. Він забезпечує сумісність між різними реалізаціями. XACML є стандартом, що базується на моделі ABAC, але може також реалізовувати RBAC як її спеціалізацію. Його структура складається з трьох ключових елементів: набору політик, політики та правила. Власник ресурсу управляє доступом, визначаючи політики, які об'єднуються в набори. Правила, що керують доступом, визначаються в межах кожної політики. Як і в моделі ABAC, архітектура XACML складається з чотирьох основних акторів (таблиця 1.1), що взаємодіють для обробки запиту на доступ до ресурсу (рис. 1.3).

Таблиця 1.1.

Актори в XACML

| **Актор** | **Опис** |
|---|---|
| Точка адміністрування політики (PAP) | Сховище для політик та обслуговування політик PDP |

| Актор | Опис |
|---|---|
| Точка прийняття рішень щодо політики (PDP) | Приймає рішення про доступ на основі запиту на доступ та також збирає пов'язані дані від інших акторів. |
| Точка виконання політики (PEP) | Це інтерфейс до запитувача та внутрішній для системи. Він обробляє запит та відповідь. |
| Точка інформації про політику (PIP) | Отримує та оцінює атрибути суб'єктів. |

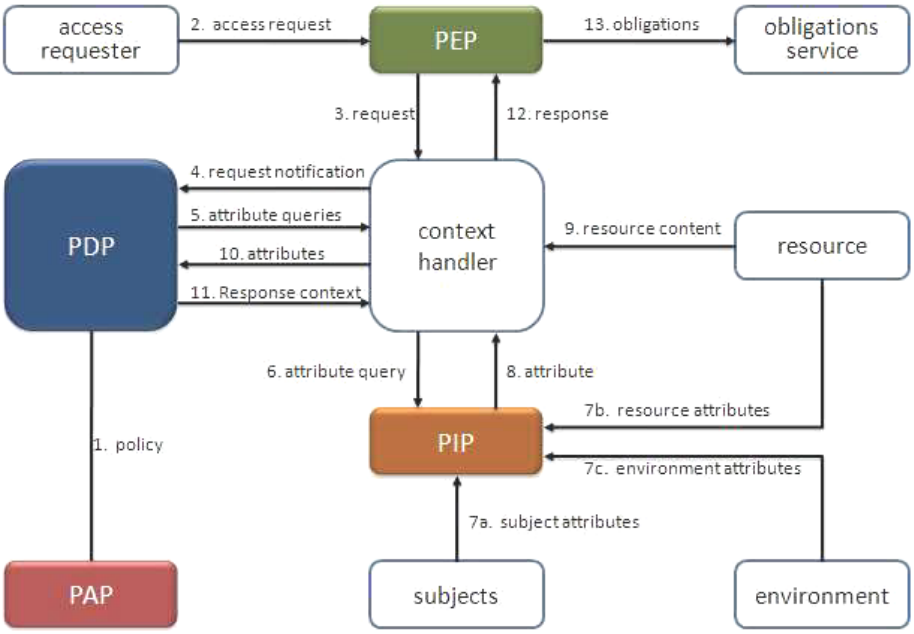


Рис. 1.3. Архітектура акторів XACML

*1.5.3. Делегування в профілі XACML*

Профіль делегування XACML дозволяє передавати права доступу у вигляді політик. Для перевірки справжності делегування використовується процес, відомий як редукція. Цей процес включає побудову графа політик, який ілюструє ієрархію делегованих прав. Граф генерується для кожного запиту на доступ:

- Пошук атрибутів. Атрибути запитувача та ресурсу шукаються у межах політики, що йому делегована.

- Побудова зв'язків. Створюються зв'язки між цією політикою та її делегованими вузлами шляхом зіставлення атрибутів суб'єктів в ієрархії делегування.

- Перевірка шляху. Аналізується шлях у графі, що з'єднує власника ресурсу із запитувачем через усіх проміжних делегуючих, для підтвердження справжності політики.

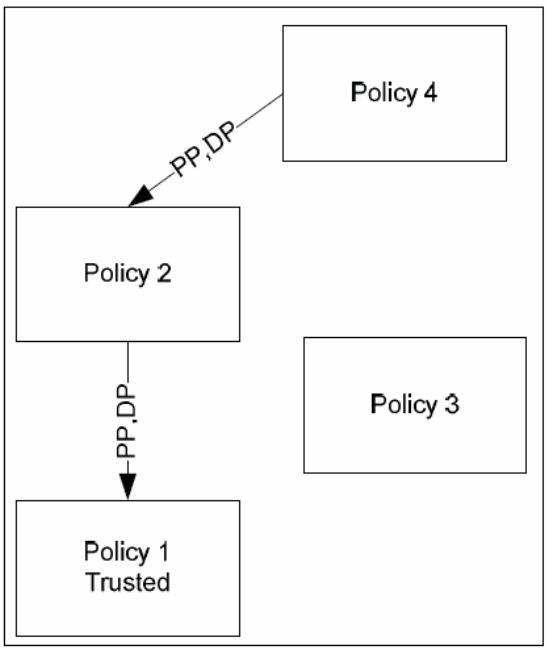


Рис. 1.4. Граф делегування XACML

Як показано на рис. 1.4, з'єднання між делегованими політиками демонструють потік делегування. Ребра графа відображають рішення про делегування: "дозволити" (PP) або "відмовити" (DP). Однак, такий підхід є обчислювально витратним, оскільки він вимагає генерації графа та пошуку атрибутів для кожного запиту, що знижує продуктивність системи.

## 1.6. Онтології як інструмент управління знаннями та контролю доступу

Онтології набули значної уваги як потужні інструменти для організації інформації та зниження складності управління знаннями. За визначенням,

онтологія є "формальною специфікацією, яка визначає базові терміни та відносини, що формують словник предметної області, а також правила для комбінування цих термінів та відносин для розширення словника."

Онтології забезпечують передачу знань та сумісність між гетерогенними суб'єктами завдяки своїм ключовим властивостям:

- Вони спрощують обмін знаннями між різними компонентами системи.

- Дозволяють повторно використовувати знання предметної області.

- Надають структурований підхід до управління та маніпулювання сутностями домену та їхніми взаємозв'язками.

Онтології особливо корисні для формалізації концептуалізації та взаємозв'язків у домені знань. Конкретні об'єкти домену, такі як користувачі та ресурси, моделюються як екземпляри цієї концептуалізації.

### *1.6.1. Компоненти онтологій та їх застосування*

Типова онтологія складається з чотирьох основних компонентів:

1. Класи.

Концептуалізують компоненти домену. Вони зазвичай організовані в таксономії, пов'язані між собою відношеннями. Відношення можуть бути таксономічними (ієрархія спадкування, наприклад, "є-частиною") або нетаксономічними (що описують інші типи зв'язків, як-от "частина-ціле", "причина-наслідок").

2. Об'єкти.

Конкретні сутності домену, що є екземплярами класів. Вони можуть мати специфічні характеристики, представлені атрибутами.

4. Відносини.

Визначають зв'язки між класами та/або об'єктами.

5. Атрибути.

Властивості, що описують об'єкти.

Онтології сприяють зниженню потреби у ручному введенні даних при змінах у моделі знань. Їх широко використовують у таких галузях, як штучний інтелект, програмна інженерія, інженерія знань та обробка природної мови. У сфері обробки природної мови, онтології забезпечують семантичне розуміння лінгвістичних термінів, що є критично важливим для аналізу текстових документів, пошукових запитів та локальних баз даних.

### *1.6.2. Онтологічна інженерія та застосування в контролі доступу*

Процес побудови онтологій (онтологічна інженерія) включає:

- Визначення класів (концепцій).
- Створення таксономічної ієрархії класів.
- Визначення атрибутів, їхніх значень і відношень між класами.
- Створення екземплярів концепцій та виконання виведення.

Найпоширенішою мовою для онтологічного моделювання є OWL (Web Ontology Language) у поєднанні з RDF (Resource Description Framework).

У контексті управління контролем доступу онтології можуть моделювати суб'єкти домену (наприклад, користувачі, ресурси) та їх взаємозв'язки. Це дозволяє відстежувати власників ресурсів, їхні відносини з користувачами та самими ресурсами. За допомогою онтологій права доступу можуть бути ефективно керовані та реалізовані, оскільки вони базуються на взаємозв'язках онтологічних сутностей. Наприклад, правило "користувач А може отримати доступ до ресурсу R1" можна представити, де А та R1 є екземплярами класів "Користувач" та "Ресурс", а "може_отримати_доступ" — це відношення. Атрибути, такі як "відділ", можуть уточнювати відносини: "тільки користувачі з 'бухгалтерського відділу' можуть отримати доступ до 'фінансових записів'".

Крім того, онтології дозволяють аналізувати семантику текстових даних, що особливо актуально для управління доступом до контенту в соціальних мережах, наприклад, на онлайн-форумах. Лексичний аналіз може ідентифікувати чутливу інформацію (наприклад, медичні терміни), що

дозволяє автоматично регулювати контроль доступу відповідно до вимог конфіденційності.

Як буде детальніше обговорено далі, дослідники вже використовували онтології для вирішення проблем RBAC та ABAC, зокрема для спрощення визначення та управління правилами та політиками. Моделювання політик за допомогою онтологій може значно підвищити продуктивність системи, дозволяючи швидко знаходити цільову політику шляхом слідування за взаємозв'язками, а не шляхом пошуку в базі даних.

## Висновки до розділу

У першому розділі було проведено ґрунтовний аналіз проблематики управління доступом та захисту конфіденційності у сучасних розподілених середовищах, що функціонують у сфері соціального нетворкінгу. Показано, що традиційні моделі контролю доступу (DAC, MAC, RBAC) демонструють низку обмежень у динамічних і гетерогенних середовищах, де користувачі постійно взаємодіють, обмінюючись контентом та делегуючи права. Особлива увага приділена питанням конфіденційності, які ускладнюються через масштабованість соціальних платформ та різнорідність інформаційних потоків. Детально досліджено принципи і методи забезпечення конфіденційності, зокрема шифрування, псевдонімізацію, політико-орієнтовані методи контролю та сучасні протоколи автентифікації.

# РОЗДІЛ 2. ОНТОЛОГІЧНІ МОДЕЛІ КОНТРОЛЮ ДОСТУПУ В СОЦІАЛЬНОМУ НЕТВОРКІНГУ ТА ХМАРНИХ СИСТЕМАХ

## 2.1. Удосконалення управління контролем доступу у хмарних середовищах за допомогою онтологічної моделі

Управління контролем доступу до ресурсів регулюється на основі вимог конфіденційності їхніх власників, що значною мірою залежить від відкритості та ефективності застосовуваного механізму. Хоча для досягнення цієї мети було запропоновано кілька рішень на основі моделей RBAC (Role-Based Access Control) та ABAC (Attribute-Based Access Control), вони мають суттєві обмеження. Ці рішення, що включають класифікацію ресурсів за категоріями, сегментування даних профілю або використання списків користувачів, демонструють низьку масштабованість у великомасштабних та складних середовищах. Ця неефективність є наслідком двох ключових проблем:

1) нездатності існуючих рішень задовольнити зростаючі вимоги до конфіденційності,

2) значного адміністративного навантаження, пов'язаного з ручним визначенням і управлінням правилами [21].

### *2.1.1. Онтології як інструмент вирішення проблем контролю доступу*

Для подолання цих недоліків було запропоновано рішення для управління контролем доступу, що моделюють суб'єкти у вигляді графів [21] або онтологій [22]. Онтології є особливо цінними, оскільки вони забезпечують формалізоване визначення концептуалізації та взаємозв'язків предметної області [23], що дозволяє представляти конкретні сутності (наприклад, користувачів і ресурси) як екземпляри цих концепцій.

Застосування онтологій дозволяє ефективно управляти контролем доступу, спираючись на семантичні взаємозв'язки між залученими

суб'єктами. Однак, існуючі підходи на основі онтологій часто визначають спеціалізовані онтології для конкретних сценаріїв. Це обмежує їх універсальність і ускладнює сумісність між гетерогенними середовищами, оскільки кожна система базується на унікальній онтологічній основі [23].

### *2.1.2. Запропоноване універсальне рішення на основі онтологій*

Для подолання цих обмежень ми представляємо універсальне онтологічне рішення, натхненне парадигмою контролю доступу на основі атрибутів (ABAC). Ця система моделює суб'єкти та їхні політики доступу, пропонуючи такі переваги:

- Її можна легко розширити для адаптації до конкретних середовищ, що дозволяє визначати контроль доступу з різним рівнем деталізації.

- Система автоматично виконує онтологічне виведення правил, що значно спрощує визначення та реалізацію політик.

Для демонстрації застосовності та переваг цього рішення ми апробували його на двох великомасштабних відкритих сценаріях: онлайн-соціальних мережах (OSN) та хмарних технологіях.

## 2.2. Онтологічна модель контролю доступу на основі атрибутів

У цьому розділі представлена розроблена онтологія, яка, як показано на рисунку 2.1, ґрунтується на моделі керування доступом на основі атрибутів (ABAC). Ця модель охоплює три ключові компоненти, необхідні для реалізації контролю доступу: суб’єкт, об’єкт і політика.

### *2.2.1. Онтологічні компоненти та їх взаємозв'язки*

Суб’єкт може виступати як власник ресурсу, що визначає права доступу для інших користувачів, або як цільовий користувач, для якого здійснюється контроль доступу. Об’єкти є ресурсами (наприклад, послуги, файли, повідомлення), що потребують захисту від несанкціонованого

доступу. Захист реалізується через визначення політик, що містять правила доступу. Кожне правило доступу є кортежем:

$$RULE \equiv <s_i, o_j, a>$$

де $s_i$ — це цільовий суб'єкт, $o_j$ — об'єкт-ресурс, а $a$ — дія, що містить рішення щодо доступу (наприклад, дозволити, заборонити).

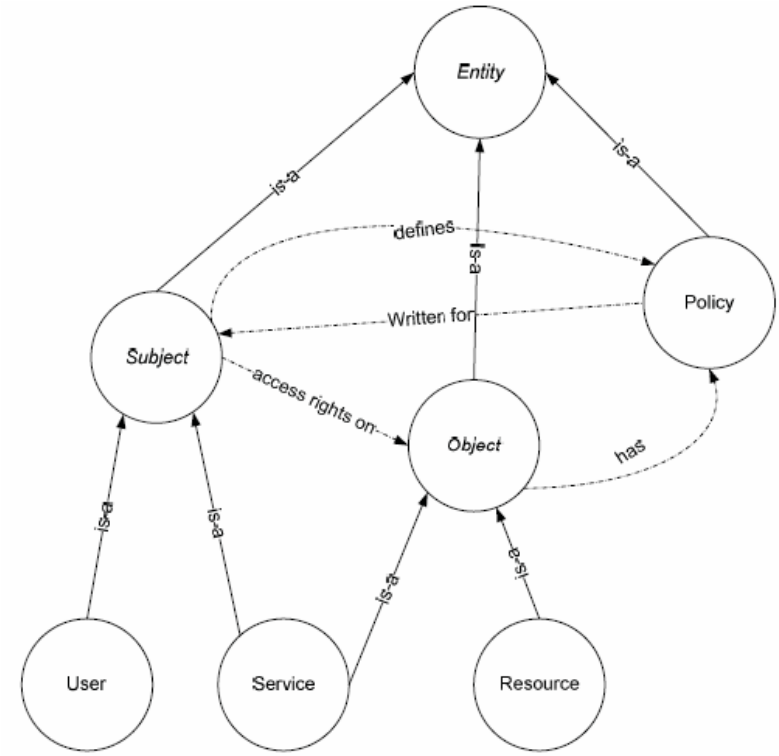


Рис. 2.1. Онтологія контролю доступу

Онтологічна властивість accesses (доступ до) між суб'єктом та об'єктом визначає роль користувача щодо ресурсу, тобто чи є він власником, чи запитувачем доступу. Властивість defines (визначає) між суб'єктом і політикою вказує на роль творця політики. Властивість appliesTo (застосовується до) демонструє зв'язок між цільовим користувачем і політикою. Нарешті, кожен ресурс пов'язаний з політикою за допомогою властивості hasPolicy (має політику).

*2.2.2. Універсальність та розширюваність онтології*

Універсальний дизайн онтології дає змогу визначати загальні правила, що посилаються на абстрактні класи (суб'єкт, об'єкт, політика), а не на конкретні сутності. Суб'єкти, що беруть участь у певному сценарії (конкретні користувачі та ресурси), можуть бути представлені як екземпляри цих онтологічних класів. Таким чином, контроль доступу до них може здійснюватися на основі загальних правил, що спираються на онтологічну структуру. Це дозволяє автоматично виводити конкретні правила на рівні суб'єкта з загальних правил, визначених на рівні класу. Крім того, універсальна онтологія може бути спеціалізована шляхом введення більш конкретних класів, що відповідають певному сценарію, і адаптації до них більш конкретних правил.

Для прийняття рішень щодо авторизації механізм контролю доступу оцінює взаємозв'язки та атрибути суб'єкта, об'єкта та політики, як визначено в моделі ABAC. Система витягує з онтології таку інформацію:

- Суб'єкт-запитувач.
- Власник ресурсу.
- Сам ресурс.
- Політика, визначена власником ресурсу.

У наступних підрозділах буде продемонстровано, як ця універсальна онтологія може бути розширена для моделювання сутностей, залучених у два поширені сценарії: онлайн-соціальні мережі (OSN) та хмарні обчислення.

## 2.3. Застосування онтологічних моделей для соціального нетворкінгу

Соціальний нетворкінг або онлайн-соціальні мережі (OSN), такі як Twitter, Facebook та інші, є платформами, що полегшують соціальні взаємодії та обмін контентом. Користувачі в OSN можуть створювати соціальні кола, групи та спільноти, що дозволяє їм встановлювати зв'язки на основі спільних

інтересів або діяльності. Значна частина активності в соціальному нетворкінгу пов’язана з публікацією та доступом до інформації, яка часто містить чутливі дані, такі як дата народження, політичні чи релігійні погляди, медична інформація тощо.

Публічне розповсюдження такого контенту через повідомлення, профілі або додатки становить серйозний ризик для конфіденційності, оскільки воно може розкривати особисте життя користувачів. Незважаючи на те, що OSN-платформи часто розглядають усіх користувачів як "друзів", рівень довіри між ними не завжди може бути виміряний [21]. Це призводить до потенційного розкриття чутливої інформації ненадійним користувачам. Дослідження [22] показує, що користувачі більше стурбовані внутрішніми загрозами конфіденційності (тобто з боку друзів), ніж зовнішніми. Таким чином, більшість "друзів" в OSN можуть вважатися ненадійними для обміну чутливими даними і така інформація потребує захисту від несанкціонованого використання третіми сторонами.

### *2.3.1. Розширення онтології для соціального нетворкінгу*

З метою забезпечення конфіденційності користувачів соціального нетворкінгу, ми пропонуємо механізм для управління правами доступу до спільних ресурсів. Рисунок 2.2 ілюструє розширення нашої загальної онтології, яке моделює сутності та їх взаємозв'язки в OSN.

У цьому сценарії суб’єкти (власники ресурсів) керують доступом до об’єктів (фотографій, повідомлень, відео), визначаючи політики контролю доступу для інших суб’єктів (інших користувачів). Ці політики містять правила доступу, що визначають права (дозволити або заборонити) до ресурсів, завантажених власником.

Оскільки соціальні мережі дозволяють класифікувати контакти за категоріями (наприклад, близькі друзі, члени сім'ї, незнайомці), клас Суб'єкт було спеціалізовано підкласом Контакт, який охоплює типи контактів користувача. Ця спеціалізація дозволяє користувачам визначати різні правила

доступу залежно від категорії контактів. Користувач моделюється як підклас Суб'єкт, а його належність до певного типу контакту власника ресурсу представлена властивістю hasContactType.

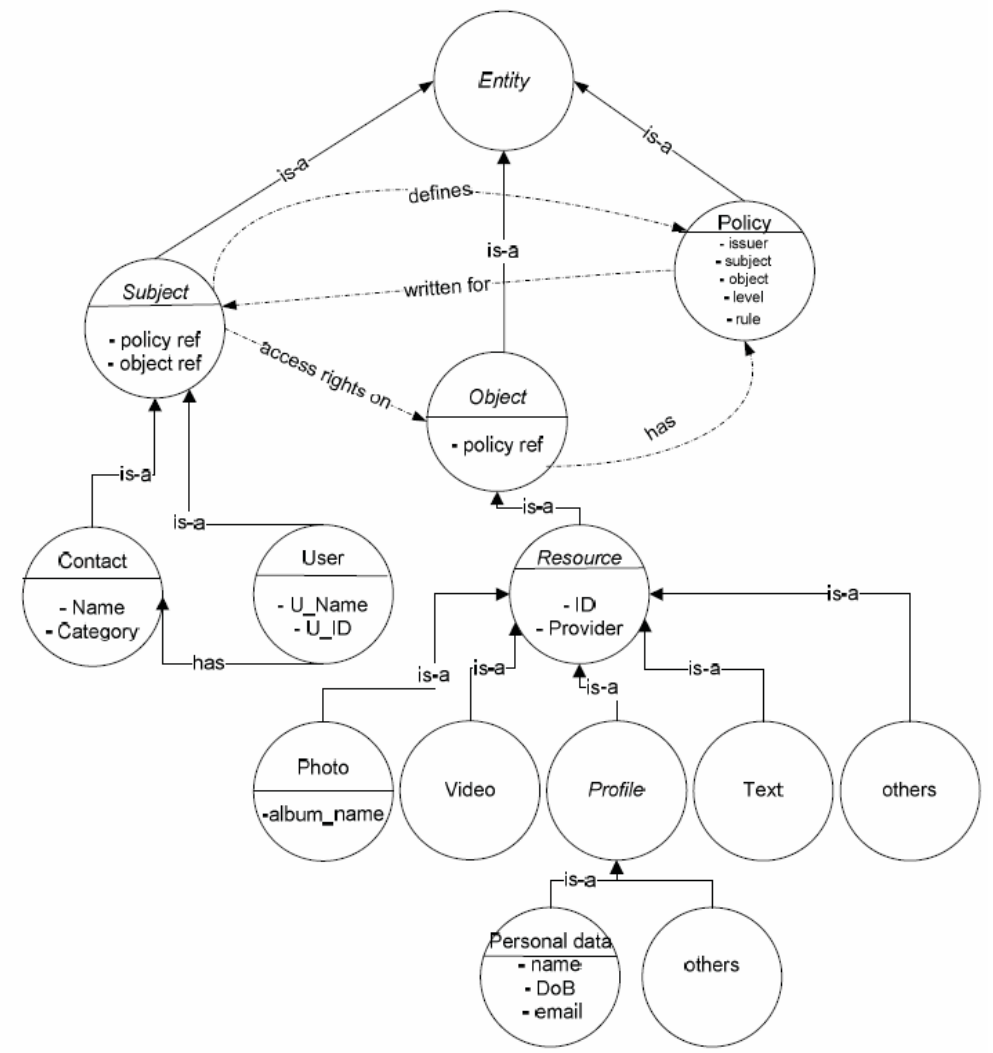


Рис. 2.2. Розширена онтологія контролю доступу для соціального нетаоркінгу

Клас Об'єкт складається з ресурсів, що потребують захисту. У контексті OSN, об'єкти спеціалізуються на конкретних типах ресурсів (Фото, Відео, Профіль, Текст), що дозволяє застосовувати більш деталізований контроль доступу замість застосування одного правила до всіх ресурсів. Клас Профіль далі класифікується на підкласи Дані профілю (для захисту

особистої інформації) та Інша інформація (для даних, що стосуються чутливої інформації, наприклад, інтересів).

Хоча ця онтологія моделює основні сутності OSN, вона може бути розширена для врахування специфіки конкретних провайдерів (наприклад, Facebook) або більш конкретних типів ресурсів.

### *2.3.2. Управління та виконання контролю доступу*

Як було описано в попередньому підрозділі, користувач може обмежити доступ до своїх ресурсів, визначаючи правила для цільових користувачів. Наш онтологічний підхід дозволяє визначення правил для онтологічних класів на будь-якому рівні абстракції. Це забезпечує автоматичне успадкування та виконання правил для відповідних підкласів та їх екземплярів (сутностей). У сценарії OSN правило за замовчуванням для всіх ресурсів – це заборонити, що дозволяє користувачеві визначати лише дозволи.

Наступний приклад демонструє спеціалізацію та інстанціацію онтології OSN для конкретного сценарію, а також автоматичне виведення правил та їх виконання.

На рисунку 2.3 показано онтологічну спеціалізацію та інстанціацію сутностей соціальної мережі, пов'язаних з обліковим записом Аліси (наприклад, Facebook).

Аліса, з метою захисту конфіденційності, визначає правило: правило_Аліси = <родичі, ресурс, 'дозволити'>. Це правило, представлене екземпляром політики, пов'язане з екземпляром ресурсу та цільовим типом контакту (родичі). Оскільки правило визначено на рівні класу (родичі та ресурс), онтологічне виведення автоматично застосовує його до всіх відповідних підкласів та їх екземплярів. Оскільки Боб є родичем Аліси, система надає йому повний доступ до всіх екземплярів фотографій та відео Аліси. Конкретно, для екземплярів користувачів, які є родичами Аліси (в даному випадку лише Боб), автоматично генеруються наступні правила:

$RULE_{Alice}$ = <*Боб, "college.jpg", 'дозволити'*>

$RULE_{Alice}$ = <*Боб, "family.jpg", 'дозволити'*>

$RULE_{Alice}$ = <*Боб, "party.avi", 'дозволити'*>

$RULE_{Alice}$ = <*Боб, "festival.avi", 'дозволити'*>

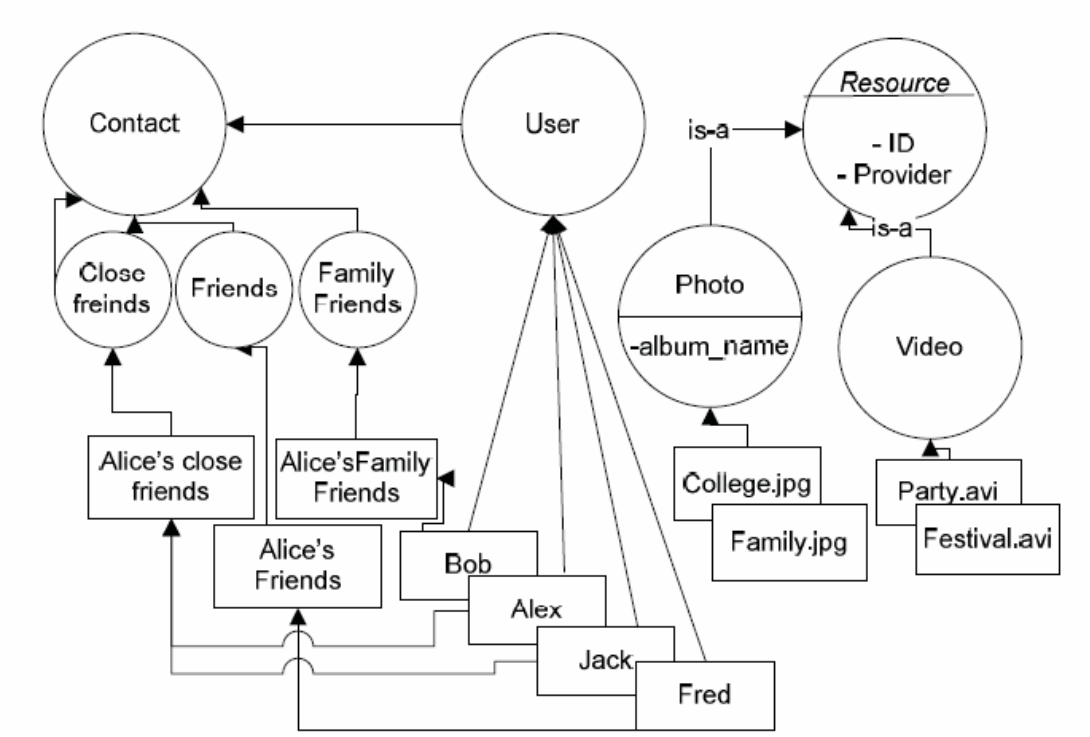


Рис. 2.3. Інстанціація онтології OSN для певного користувача

Крім того, Аліса може визначати правила для конкретних екземплярів користувачів. Наприклад, вона може дозволити лише Алексу з групи близьких друзів доступ до всіх її фотографій: правило_Аліси = <Алекс, фото, 'дозволити'>. З цього загального правила виводяться наступні правила:

$RULE_{Alice}$ = <*Алекс, "college.jpg", 'дозволити'*>

$RULE_{Alice}$ = <*Алекс, "family.jpg", 'дозволити'*>

## 2.4. Приклад застосування онтологічних моделей для платформи хмарних обчислень

Хмарні обчислення є універсальною платформою, що забезпечує спільне використання ресурсів та надання послуг користувачам. Через свою відкриту архітектуру вона вимагає масштабованого механізму для керування доступом до спільних ресурсів. Хмара зазвичай функціонує на основі багаторівневих моделей обслуговування:

- Інфраструктура як послуга (IaaS) - спільне використання фізичних обчислювальних ресурсів.

- Платформа як послуга (PaaS) - надання доступу до баз даних та операційних систем.

- Програмне забезпечення як послуга (SaaS) - надання програмних додатків.

Для управління правами доступу в цьому середовищі ми розширили нашу загальну онтологію, включивши хмарні сутності та атрибути, релевантні для керування доступом у хмарі.

### *2.4.1. Розширення онтології для хмарних обчислень*

На рисунку 2.4 представлено розширену онтологію, що містить хмарні сутності (орендар, хмарна послуга та хмарний ресурс) та їх взаємозв'язки.

На цьому рисунку, клас Орендар є підкласом Суб'єкта і охоплює двох акторів: (користувача — споживача хмарних послуг) та (постачальника хмарних послуг (CSP) — надавача послуг). Аналогічно, Послуга є підкласом Суб'єкта, що представляє послуги, надані CSP, які можуть потребувати доступу до спільних ресурсів. Водночас, Послуга є підкласом Об'єкта, оскільки орендарі можуть отримувати до них доступ. Нарешті, Хмарні ресурси можуть бути Апаратними (сервери, дисковий простір) або Програмними (веб-додатки, веб-сервіси). CSP може керувати доступом до цих ресурсів і послуг, визначаючи правила, які містяться в політиках.

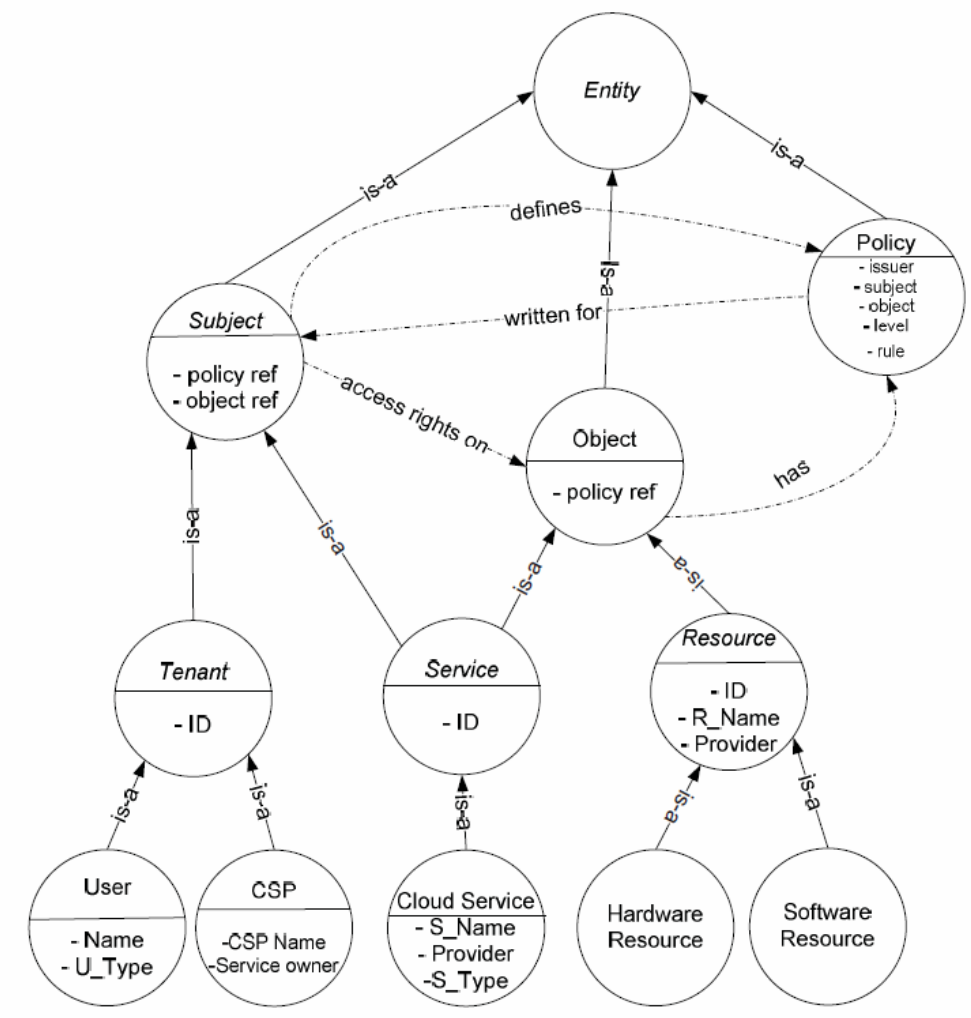


Рис. 2.4. Розширена онтологія контролю доступу для хмари

*2.4.2. Управління та виконання контролю доступу*

На рисунку 2.5 зображено розширення та інстанціацію хмарної онтології для CSP Google, який надає послуги та ресурси користувачам.

Google пропонує послуги на різних рівнях (SaaS, PaaS, IaaS) для стандартних користувачів та освітніх установ (наприклад, освітні установи отримують більший простір на Google Drive та професійний домен). Google налаштовує доступ за допомогою двох правил:

1) дозволяє послугам SaaS доступ до всіх ресурсів;

2) надає користувачам з освітніх установ спеціальний доступ до послуг, призначених для освітніх цілей.

Відповідно до моделі ABAC, ми можемо використовувати атрибути для визначення загальних правил:

*RULE_Google-R1 ≡ <SaaS, ресурс, 'дозволити'>*

*RULE_Google-R2 ≡ <Користувачі <U_Type="Освіта">, Хмарні послуги <S_Type="Освіта">, 'дозволити'>*

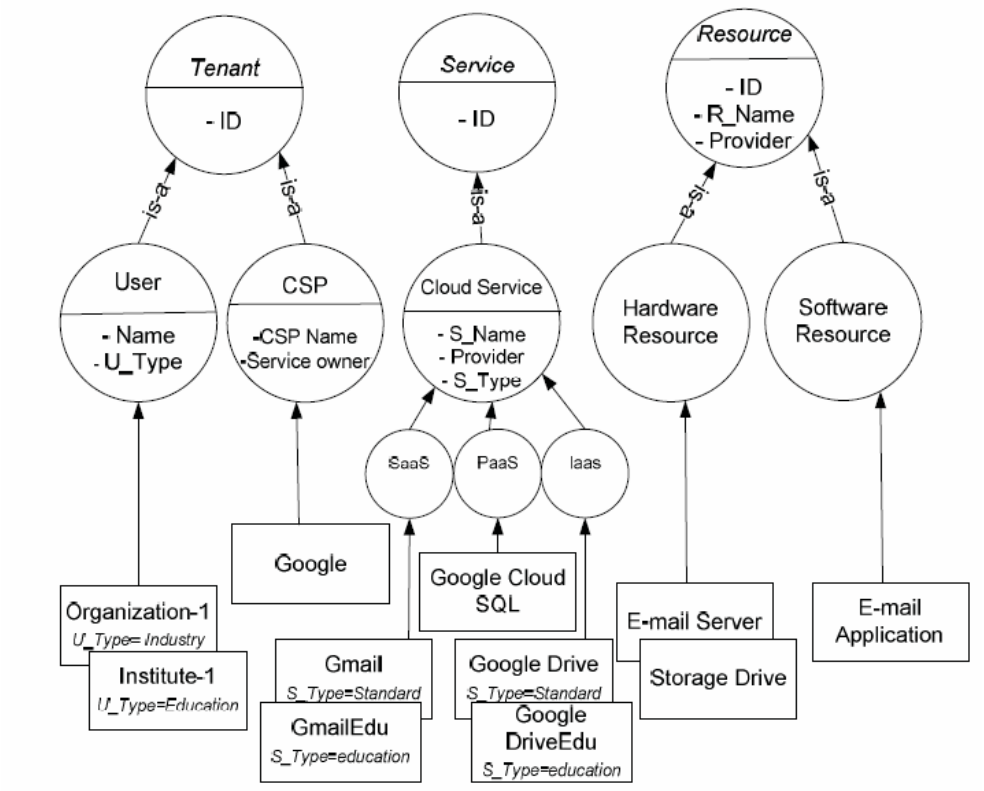


Рис. 2.5. Створення екземпляра хмарної онтології для Google

RULE_Google-R1 визначено на концептуальному рівні класів Ресурсу та SaaS, що забезпечує його застосування до всіх суб'єктів у підкласах Апаратного ресурсу та Програмного ресурсу. Шляхом виведення конкретних правил на рівні екземплярів отримуємо:

*RULE _Google-R1 = <Gmail, поштовий сервер, 'дозволити'>*

*RULE _Google-R1 = <Gmail, дисковий простір, 'дозволити'>*

*RULE _Google-R1 = <Gmail, поштові додатки, 'дозволити'>*

*RULE _Google-R1 = <GmailEdu, поштовий сервер, 'дозволити'>*

*RULE _Google-R1 = <GmailEdu, дисковий простір, 'дозволити'>*

*RULE _Google-R1 = <GmailEdu, поштові додатки, 'дозволити'>*

З іншого боку, RULE_Google-R2 надає доступ до хмарних послуг, призначених для освітніх установ. Тип користувача визначається значенням атрибута U_Type, а освітні послуги — значенням S_Type. Завдяки цьому виведенню, екземпляри користувачів з освітнім типом отримують доступ до всіх хмарних послуг, призначених для них:

*RULE _Google-R2 = <Інститут-1, GmailEdu, 'дозволити'>*

*RULE _Google-R2 = <Інститут-1, Google DriveEdu, 'дозволити'>*

У цьому розділі представлена універсальна онтологія, що моделює сутності, їх взаємозв'язки та політики контролю доступу. Вона легко розширюється для конкретних середовищ. Для демонстрації її застосовності, ми розширили її для двох масштабних відкритих сценаріїв: OSN та хмарних обчислень. Ми проілюстрували, як визначення правил та управління доступом значно спрощуються для системних адміністраторів, оскільки можуть здійснюватися на рівні концептуальних класів. Динамічні правила для конкретних сутностей можуть автоматично виводитися, що є критично важливим у динамічних відкритих середовищах.

## Висновки до розділу

Другий розділ присвячено розробці онтологічних моделей контролю доступу у хмарних середовищах і соціальних мережах. Показано, що

застосування онтологій є ефективним способом подолання недоліків традиційних підходів, оскільки вони забезпечують семантичну інтерпретацію політик та зв’язків між сутностями. Запропоновано універсальне рішення на основі онтологічної моделі контролю доступу, що інтегрує атрибутивний підхід (ABAC) та дозволяє формалізувати правила на рівні концептів і відношень. Досліджено онтологічні компоненти та їх взаємозв’язки, що дає змогу створювати розширювані та адаптивні моделі для динамічних середовищ. Особливо підкреслено універсальність онтологічних моделей, які можуть бути адаптовані як для соціальних мереж, так і для хмарних платформ.

# РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ ОНТОЛОГІЧНИХ МОДЕЛЕЙ КОНТРОЛЮ ДОСТУПУ В ХМАРНИХ ЗАСОБАХ СОЦІАЛЬНОГО НЕТВОРКІНГУ

## 3.1. Механізм делегування контролю доступу в динамічних середовищах

Делегування прав доступу є механізмом, що дозволяє користувачам передавати свої повноваження доступу до ресурсів іншим суб'єктам, що сприяє зниженню адміністративного навантаження. Традиційні моделі делегування часто ґрунтуються на контролі доступу на основі ролей (RBAC), де делегування здійснюється через передачу ролей. Інші підходи використовують контроль доступу на основі атрибутів (ABAC), де делегування керується політиками.

Виконання делегування в динамічних, розподілених системах, таких як хмарні обчислення, є особливо складним через велику кількість гетерогенних суб'єктів, що вимагає значного адміністративного навантаження. Розглянемо приклад хмарного середовища, де постачальники хмарних послуг (CSP - Cloud service providers) надають послуги багатьом орендарям. У мультиорендному середовищі управління доступом до спільних послуг становить серйозну проблему для провайдера, оскільки доступ до них повинні мати лише авторизовані орендарі. Провайдери хмарних послуг стикаються з проблемою ефективного управління доступом через різноманітність послуг та вимог безпеки орендарів.

Одним з рішень є делегування контролю доступу, що забезпечує децентралізацію управління, масштабованість для великих організацій та ефективне керування змінами ролей. За допомогою делегування, CSP (як делегуючий) може передавати адміністративні привілеї щодо певної послуги іншим орендарям (делеговані). Ці орендарі, у свою чергу, можуть керувати доступом для своїх користувачів.

### *3.1.1. Проблематика делегування в хмарному середовищі*

У хмарному середовищі делегування може відбуватися на різних рівнях обслуговування (IaaS, PaaS, SaaS) і бути ієрархічним. Однак гетерогенність хмарних федерацій та недостатній рівень довіри ускладнюють авторизацію та верифікацію делегуючих. Авторизація — це процес передачі прав доступу, тоді як верифікація підтверджує повноваження делегуючого. Особистість делегуючого може бути сфальсифікована, що призводить до атак спуфінгу та несанкціонованого делегування [23].

Існуючі хмарні рішення, такі як Microsoft Azure, часто використовують прості механізми, засновані на облікових даних (ім'я користувача, пароль). Цей підхід вимагає централізованого управління, що може перевантажувати адміністраторів і бути неефективним для організацій, що вимагають подальшого делегування з обмеженими правами.

### *3.1.2. Онтологічна модель делегування*

У цьому дослідженні ми пропонуємо онтологічний фреймворк для делегування, який є розширенням профілю делегування XACML. Даний підхід, на відміну від існуючих методів не потребує ручного визначення обмежень чи правил; управління делегуванням здійснюється за допомогою автоматичних алгоритмів. Для цього ми використовуємо нашу раніше запропоновану онтологію контролю доступу, яку інстанціюємо хмарними сутностями для моделювання робочого процесу делегування.

Ключові особливості:

- Моделювання делегування за допомогою онтології. Ми використовуємо розширену онтологію для моделювання основних сутностей делегування (делегуючі, делеговані, ресурси, політики) та їх взаємозв'язків. Це дозволяє відстежувати, хто, які привілеї та до якого ресурсу делегує, а також забезпечує інтуїтивне рішення для верифікації атрибутів суб'єктів.

- Розподілена модель делегування. Ми представляємо модель, яка класифікує основних акторів хмари (CSP, організації, користувачі) на різні

рівні делегування. На відміну від інших рішень, делегування відбувається розподіленим способом. Автентичність політики делегування верифікується за допомогою довіреної політики, написаної та підписаної CSP.

- Автоматична верифікація повноважень. Наша система автоматично верифікує повноваження делегуючого через атрибути суб'єктів та політику, дотримуючись їх онтологічних взаємозв'язків, що призводить до довіреної політики. Це відрізняє наш підхід від методів, що покладаються на верифікацію через ролі.

- Автоматизоване виконання та відкликання делегування. Пропонована система не вимагає визначення додаткових правил. Вона автоматично виконує делегування, верифікує повноваження та відкликає привілеї за допомогою простих алгоритмів. Крім того, делегована політика автоматично інтегрується з існуючою, використовуючи алгоритм об'єднання політик, який також вирішує можливі конфлікти.

## 3.2. Делегування контролю доступу з використанням онтологічної моделі

Традиційні підходи до делегування прав доступу, зокрема профіль делегування XACML, перевіряють повноваження видавця політики через граф делегування. Цей граф генерується для кожного запиту на доступ до ресурсу. Атрибути запиту (користувач, ресурс) зіставляються з делегованими політиками, що зберігаються в базі даних. Це дозволяє ідентифікувати відповідні політики, які формують ієрархічний граф. Верифікація повноважень відбувається шляхом перевірки ієрархічного ланцюжка делегуючих у цьому графі. Однак такий підхід є неоптимальним для масштабованих хмарних середовищ через:

1) значні накладні витрати на пошук політик та генерацію графа для кожного запиту;

2) велику кількість гетерогенних суб'єктів;

3) високу частоту запитів, що обробляються постачальниками хмарних послуг (CSP).

*3.2.1. Онтологічний підхід до делегування*

Наш підхід є розширенням профілю делегування XACML, що включає онтологію для моделювання семантики контролю доступу та його делегування. Використання онтологічної парадигми має переваги:

1) спрощення реалізації процесу делегування,

2) автоматичне визначення та інтерпретація взаємозв'язків між суб'єктами.

Наша онтологія моделює суб'єктів (делегуючий, делегований, ресурси), їх типи та взаємозв'язки. У цій моделі суб'єкти представлені як екземпляри класів, а робочий процес делегування відображається через їх взаємозв'язки. Для обробки запиту на доступ, повноваження делегуючого (наприклад, CSP) оцінюється та перевіряється через взаємозв'язки екземплярів онтології, а не шляхом пошуку в політиках. Атрибути запиту зіставляються з онтологічними екземплярами, і відповідна політика витягується з бази даних.

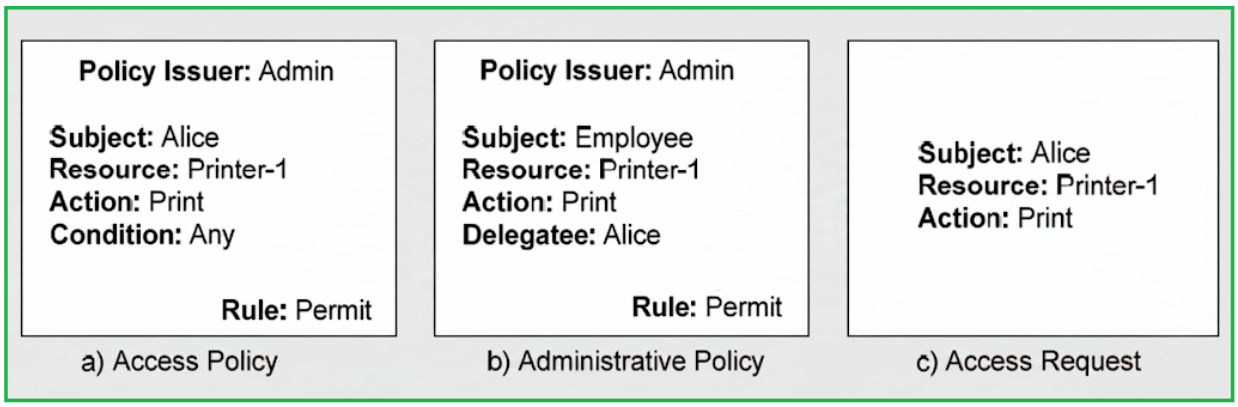


Рис. 3.1. Структура політик та запиту на доступ

На відміну від профілю XACML, де політики всіх делегуючих, що розділяють ресурс, зберігаються в єдиному наборі, у нашій системі кожен

делегуючий підтримує власний набір політик для кожного ресурсу. Це значно зменшує накладні витрати на:

1) пошук політики у великомасштабних середовищах,

2) генерацію графа делегування.

У нашій моделі користувач може керувати двома типами політик (рис. 3.1):

- Політика доступу - дозволяє або забороняє доступ до ресурсу.

- Адміністративна політика - надає орендарю привілеї видавати політики іншим орендарям.

Наприклад, на рис. 3.1 a) CSP дозволяє Алісі доступ до принтера, а на рис. 3.1 b) він делегує їй повноваження видавати інші політики.

*3.2.2. Онтологічне представлення ABAC та хмарних сутностей*

Сутності, що беруть участь у моделі ABAC, — це суб'єкти, об'єкти та політики. Суб'єкти володіють або делегують привілеї на об'єкти (послуги, ресурси). Це досягається шляхом визначення правил політики доступу для об'єктів та цільових суб'єктів.

XACML-модуль (PIP) оцінює атрибути суб'єкта та об'єкта для прийняття рішень щодо авторизації. Аналогічно, права доступу можуть бути делеговані через політики делегування, які містять атрибути та правила для делегованих ресурсів.

Для автоматизації робочого процесу делегування хмарні сутності моделюються як класи в онтології. Атрибути політик використовуються для моделювання взаємозв'язків суб'єктів як онтологічних властивостей. Спрямовані ребра в онтології представляють таксономічні залежності та асоціативні властивості між класами.

На рисунку 3.2 представлено онтологію, що відображає хмарні сутності та логіку процесу делегування повноважень суб'єктів для платформ хмарних обчислень.

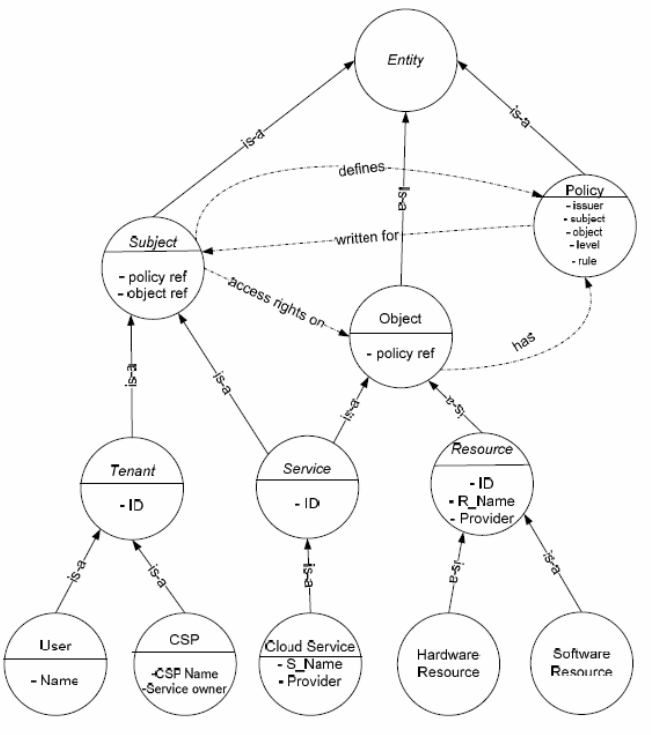


Рис. 3.2. Онтологічне моделювання процесу делегування для хмарних суб'єктів

### *3.2.3. Представлення робочого процесу делегування*

Запропонована онтологія інстанціюється для демонстрації робочого процесу делегування в хмарному середовищі.

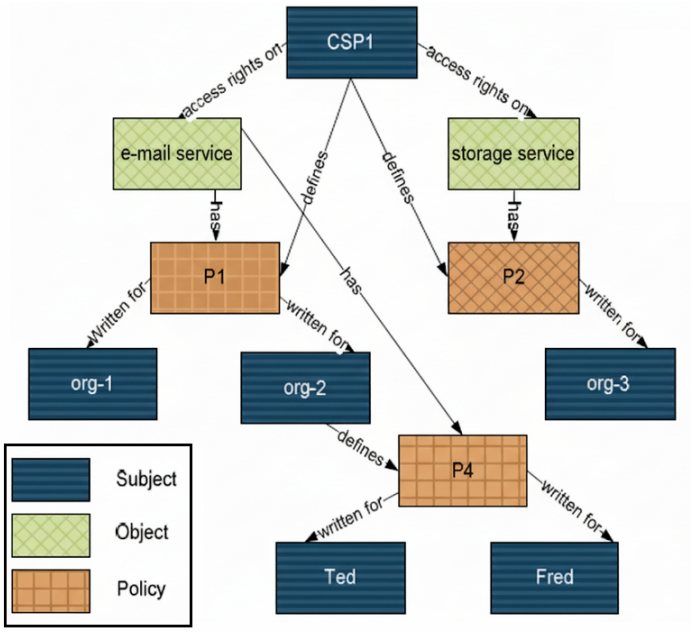


Рис. 3.3. Представлення робочого процесу делегування

На рис. 3.3 представлено інстанційований робочий процес, де суб'єкти, об'єкти та політики взаємопов'язані. Кожен делегуючий підтримує свій набір політик у локальному сховищі.

У цьому прикладі, CSP1 надає послуги електронної пошти та зберігання. Він делегує привілеї організаціям, зберігаючи окремі набори політик (P1 для org-1 та org-2, P2 для org-3). Організації org-1 та org-3 отримують лише політику доступу, тоді як org-2 отримує адміністративну політику, що дозволяє подальше делегування. org-2, у свою чергу, делегує права доступу до електронної пошти користувачам Алексу та Теду, підтримуючи власний набір політик P4.

### *3.2.4. Робочий процес системи*

Коли користувач ініціює запит на делегування прав доступу, система автоматично створює та оновлює онтологічну модель взаємозв'язків між суб'єктами (делегуючий, делегований, ресурс) та наборами політик. Цей процес, деталізований в Алгоритмі 1, робить верифікацію делегування ефективнішою, ніж у профілі XACML, який генерує граф делегування для кожного запиту. Наш підхід оновлює робочий процес лише під час делегування, а не для кожного запиту на доступ, що значно знижує накладні витрати.

Для обробки запиту на доступ система виконує такі кроки:

1. Початкова верифікація. Перевіряє права доступу користувача згідно з політикою, пов'язаною з ним в онтологічній моделі.

2. Верифікація повноважень. Визначає видавця політики та перевіряє його автентичність. Це досягається шляхом зіставлення атрибутів запиту на доступ (рис. 3.1 c) з атрибутами екземплярів політики в онтологічному робочому процесі.

3. Перевірка ланцюга делегування. Якщо політика знайдена, система генерує адміністративний запит для верифікації автентичності видавця. Вона перевіряє атрибути делегуючого, пов'язаного з цільовою політикою, і

послідовно верифікує повноваження всіх попередніх делегуючих до початкового власника ресурсу.

4. Рішення про доступ. Доступ надається лише за умови, що:

1) ланцюг делегування є легітимним (тобто походить від довіреної політики, підписаної власником ресурсу);

2) користувач має відповідні права доступу. В іншому випадку, доступ відхиляється.

Аналогічно, запит на відкликання прав доступу ініціюється делегуючим. Система верифікує цільову політику та її ланцюг, після чого видаляє відповідні екземпляри з онтологічної моделі, скасовуючи подальші делегування.

### *3.2.5. Конфлікти політик*

У розподіленому середовищі, такому як хмара, можуть виникати конфлікти між правилами політик. Наші алгоритми обробляють ці ситуації без необхідності визначення додаткових правил або обмежень, що є відмінністю від підходів.

Два основних типи конфліктів:

- Конфлікт дозволу/заборони - один користувач має дві політики від різних суб'єктів на спільний ресурс, де одна дозволяє доступ, а інша забороняє.

- Конфлікт політики доступу/адміністративної політики - користувач має дві політики на спільний ресурс, де одна надає лише доступ, а інша — права на подальше делегування.

Система вирішує ці конфлікти, надаючи пріоритет делегуючим вищого рівня. Якщо делегуючі знаходяться на одному рівні, застосовується найсуворіше правило (як правило, "заборонити").

На рис. 3.4 проілюстровано два сценарії конфліктів:

- Конфлікт 1.

Відділ dept1 має політики від org-1 (забороняє) та CSP1 (дозволяє) на спільний ресурс електронної пошти. Політика від CSP1 має пріоритет, оскільки він є делегуючим вищого рівня. Отже, доступ надається.

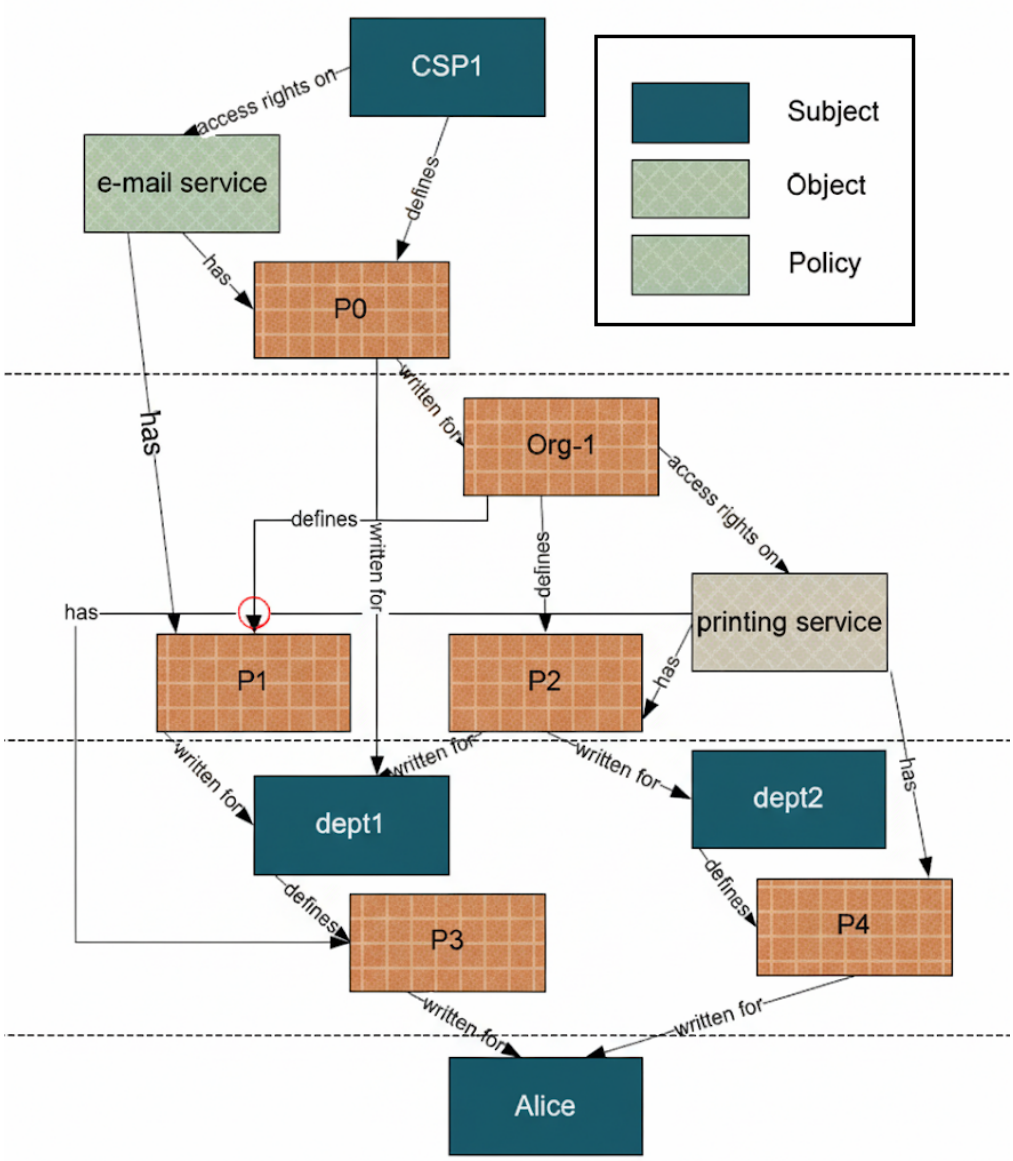


Рис. 3.4. Приклад конфлікту політик

- Конфлікт 2.

Відділ dept1 надає Алісі доступ до принтера, а dept2 забороняє. Оскільки dept1 і dept2 знаходяться на одному рівні делегування, застосовується найсуворіше правило. В цьому випадку, це заборона (надана dept2).

### 3.3. Виконання делегування доступу в хмарному середовищі

Для підтримки делегування контролю доступу в хмарному середовищі необхідно класифікувати хмарні сутності відповідно до їх ролей та типів послуг. Ця класифікація дозволяє керувати делегуванням в межах рівнів обслуговування (IaaS, PaaS, SaaS) та забезпечує основу для управління політиками. Існує три основні категорії суб'єктів, що керують хмарними ресурсами:

- Постачальник хмарної інфраструктури (CP) - надає базову інфраструктуру (сервери, сховища).

- Постачальник хмарних послуг (CSP) - пропонує платформи, послуги або додатки.

- Орендарі - фактичні споживачі послуг, які можуть надалі керувати доступом для інших користувачів у межах своїх організацій.

Для децентралізації делегування ці суб'єкти повинні мати розподілені локальні сховища для зберігання своїх політик. Це зменшує навантаження на одного суб'єкта. Робочий процес делегування та управління політиками в межах хмарних сутностей представлений на рисунку 3.5.

Рівні делегування:

- Перший рівень: CP делегує ресурси CSP, зберігаючи політики у своєму локальному сховищі.

- Другий рівень: CSP керують доступом до ресурсів (своїх або делегованих від CP) шляхом підтримки політик у власних сховищах. Вони можуть делегувати ресурси іншим CSP або орендарям.

- Третій рівень: Орендарі керують делегованими ресурсами та можуть передавати права доступу іншим орендарям або користувачам, якщо це дозволено адміністративною політикою.

Незважаючи на розподілене зберігання політик, система об'єднує всі сутності через єдиний робочий процес делегування, що дозволяє легко верифікувати повноваження делегуючого на будь-якому рівні.

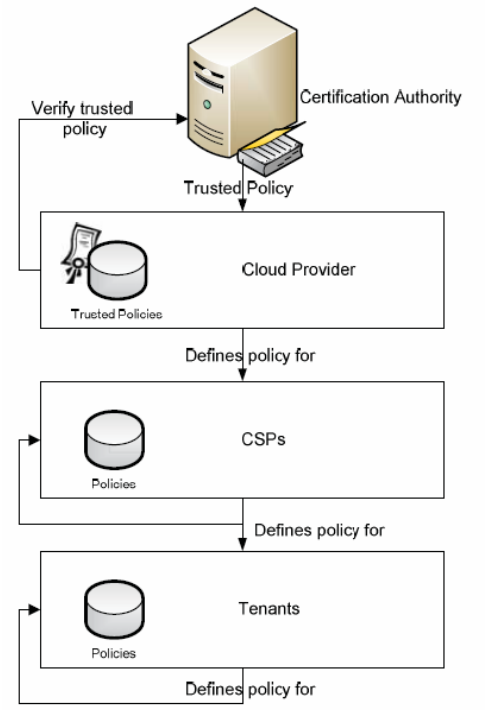


Рис. 3.5. Потік делегування та управління політиками в хмарному середовищі

Для забезпечення надійності, робочий процес делегування має бути довіреним. Довіра встановлюється через довірену політику, яка ініціює робочий процес і цифровим підписом засвідчується власником ресурсу. У хмарному середовищі CP та CSP можуть бути власниками ресурсів, тому їхні політики вважаються довіреними. Ми припускаємо, що ресурси та їхні власники зареєстровані в надійному центрі сертифікації.

Завдяки онтологічним зв'язкам між користувачами, ресурсами та політиками, наша система є більш стійкою до підробки ідентичності (спуфінгу). На відміну від поточного профілю XACML, де політика вважається довіреною, якщо вона не містить елемента видавця, наш підхід вимагає цифрового підпису, що значно підвищує безпеку.

### *3.3.1. Делегування, верифікація та відкликання повноважень*

Алгоритм 1 описує процес делегування. Система отримує запит на делегування, що містить інформацію про делегуючого, делегованого та ресурс. На першому етапі верифікується авторитет делегуючого. Якщо

делегуючий має відповідні адміністративні привілеї, починається процес делегування, ініціюється зв'язок з локальним сховищем політик.

**Algorithm 1:** *DelegationOfAuthority (subject_delegator, subject_delegatee, object_resource, action)*

```
1:  authority ← VerifyDelegatorAuthority(subject_delegator,object_resource, action);
2:  if authority is valid then
3:      establish connection with local database of subject_delegator;
4:      if subject_delegator is the owner then
5:          set delegation_level; // e.g. default value is '5';
6:          prepare trusted policy by signing an attribute of the owner;
7:      else  prepare delegated policy;
8:          delegation_level = Delegation_level -1;
9:      end if
10:     if delegation_level is greater than zero then
11:         create new policy_rule for the subject_delegatee;
12:         define new policy by adding policy_rule and delegation_level;
13:         if subject_delegator has policySet for the object_resource then
14:             add policy to the existing policySet;
15:         else  create new policySet;
16:             add policy to the policySet; end if
17:         store policySet in the local database of the subject delegator;
18:         CreateInstanceRelations (subject_delegator, subject_delegatee, object_resource,
            policySet);
19:     else object_resource reached maximum limit of delegation; end if
20: else  not a valid delegator;  end if
```

Алгоритм 1 описує кроки, необхідні для делегування прав доступу від одного суб'єкта до іншого.

1. Перевірка повноважень. Спочатку система перевіряє, чи має делегуючий суб'єкт (той, хто надає права) достатні повноваження для виконання цієї дії. Якщо ні, алгоритм зупиняється.

2. Визначення ролі делегуючого. Якщо повноваження дійсні, алгоритм встановлює з'єднання з базою даних делегуючого. Система перевіряє, чи є делегуючий власником ресурсу.

- Якщо так, це початкове делегування. Система встановлює максимальний рівень делегування і створює довірену політику, яка підписується власником.

- Якщо ні, це повторне делегування. Система готує делеговану політику і знижує рівень делегування на одиницю. Це гарантує, що права не можуть делегуватися безкінечно.

3. Створення та збереження політики. Алгоритм перевіряє, чи не вичерпано ліміт делегування. Якщо рівень делегування дозволяє, він створює нове правило для делегованого суб'єкта (того, хто отримує права) і додає його до політики. Потім система перевіряє, чи існує вже набір політик для цього ресурсу в базі даних делегуючого.

- Якщо набір існує, нова політика додається до нього.

- Якщо ні, створюється новий набір.

4. Створення онтологічних зв'язків. Після збереження політики, алгоритм створює онтологічні зв'язки між суб'єктами, ресурсом та набором політик. Це оновлює модель системи, відображаючи нові відносини делегування. Цей процес забезпечує, що делегування є ієрархічним, верифікованим та керованим, дозволяючи системі ефективно відстежувати та перевіряти повноваження без потреби щоразу генерувати граф делегування.

**Algorithm 2:** *AccessRequest (subject_requester, object_resource, action)*

```
1:  policyFound = false;
2:  subjectInstances ← get all subject instances connected with the policy instance of the
    subject_requester from ontology-based delegation workflow;
3:  while there are subjectInstances to parse and policyFound is false do
4:      policyInstance ← get policy instance related to object_resource that connects
        subjectInstance and subject_requester through the ontology-based workflow;
5:      policySet ← get policy set related to object_resource through policyInstance from the
        policy repository;
6:      policy ← get policy issued to subject_requester;
7:      if policy is valid then
8:          rule ← read rule of the policy;
9:          if rule matches with the action then
10:             authority = VerifyDelegationAuthority (subjectInstance, object_resource, action);
11:             if authority is valid then
12:                 policyFound = true;
13:             end if
14:         end if
15:     end if
16: end while
17: if policyFound is true and authority is valid then
18:     grant Access;
19: else deny access; end if
```

Алгоритм 2 використовується для верифікації авторитету. Цей алгоритм перевіряє, чи є делегуючий власником ресурсу. Якщо так, його авторитет вважається достовірним. Якщо ні, алгоритм перевіряє, чи є делегуючий делегованим, і рекурсивно перевіряє авторитет вищестоящих делегуючих у ланцюгу, аж до власника ресурсу.

Алгоритм 3 описує процес відкликання. Коли делегуючий запитує відкликання прав доступу, система перевіряє його повноваження. Якщо делегуючий є власником, він може видалити всі пов'язані з ресурсом політики. Якщо він є делегованим, система перевіряє його повноваження і, якщо запит справжній, видаляє політики, пов'язані з делегованим.

**Algorithm 3:** *VerifyDelegatorAuthority (subject, object_resource, action)*

```
subInstances ← get all subject instances connected to the policy instances of the
object_resource that are connected to the subject from the ontology-based workflow;
while there are subjectInstance to parse do
    policyInstance ← get subject's policy instance connected to the owner's instance from
    the ontology;
    policySet ← get policy set through policyInstance from the database;
    policy ← get policy issued to subjectInstance from the policySet;
    rule ← get rule from the policy;
    if subjectInstance is NOT the owner then
        if rule matches with the action then
            authority ← VerifyDelegatorAuthority (subjectInstance,object_resource,action);
        else authority = 'invalid';
            continue with other subjectInstances; end if
    else
      if rule matches with the action then
            signature ← verify identity of the owner using her public key;
            if signature is valid then
              authority = 'valid';
              end while;
              return authority;
            else authority = 'invalid';
            end if
        else authority = 'invalid';
            continue with other subjectInstances; end if
    end if
end while
return authority;
```

### *3.3.2. Алгоритм об'єднання політик*

Алгоритм 4 вирішує конфлікти між політиками, що можуть виникнути, коли суб'єкт має політики від різних делегуючих.

**Algorithm 4:** *RevokeDelegatorAuthority (subject_delegator, subject_delegatee, object_resource, action)*

```
1:  authority ← VerifyDelegatorAuthority (subject_delegator,object_resource,action);
2:  policyInstance ← get subject_delegator's policy instance from ontological workflow;
3:  policySet ← get policy set through policyInstance from database;
4:  policy ← get policy issued to subjectInstance from the policySet;
5:  rule ← get rule from the policy;
6:  if authority is valid & rule matches with the action then
7:      delete policy issued to subject_delegatee from policy set;
8:      if policy set is empty then
9:          delete policyInstance;
10:     else update policy set in database; end if
11: else invalid request; end if
```

1. Система ідентифікує всі політики, пов'язані з ресурсом.

2. Визначає правила в кожній політиці.

3. Порівнює правила для виявлення конфліктів.

4. Вирішує конфлікти на основі пріоритету делегуючих (вищестоящі мають пріоритет).

5. Якщо делегуючі знаходяться на одному рівні, застосовується найсуворіше правило (наприклад, "заборонити" має пріоритет над "дозволити").

6. Якщо конфліктів немає, всі політики виконуються.

Алгоритм 5 називається "Алгоритм об'єднання політик" (Policy CombiningAlgorithm), призначений для вирішення конфліктів між двома політиками доступу, які стосуються одного й того ж ресурсу.

Алгоритм приймає на вхід два екземпляри політик, цільовий суб'єкт-делегата ($subject_{delegatee}$) та цільовий ресурс ($objectr_{esource}$).

Алгоритм отримує рівень делегування для кожної з конфліктних політик (DL1 і DL2). Рівень делегування вказує на позицію делегуючого в ієрархії: менше число означає вищий рівень делегування (ближче до власника ресурсу).

Далі він порівнює рівні делегування (DL1 і DL2). Якщо DL1 більший за DL2, це означає, що політика policyInstance1 була видана суб'єктом, який

знаходиться на нижчому рівні в ієрархії. Таким чином, пріоритет має політика, видана суб'єктом, який знаходиться вище в ієрархії.

В цьому випадку, якщо DL1 > DL2, то повертається policyInstance1. Інакше, повертається policyInstance2. Це означає, що перевага надається політиці, що була видана суб'єктом з вищим рівнем делегування (наприклад, ближче до власника ресурсу).

**Algorithm 5:** *PolicyCombiningAlgorithm (policyInstance1, policyInstance2, subject_delegatee, object_resource)*

```
1:  policySet1 ← get policy set from policy database using policyInstance1 instance for
    object_resource;
2:  policySet2 ← get policy set from policy database using policyInstance2 instance for
    object_resource;
3:  if policySet1 and policySet2 are valid then
4:      policy1 ← get policy from policySet1 issued to subject_delegatee;
5:      policy2 ← get policy from policySet2 issued to subject_delegatee;
6:      DL1 ← get delegation level of policy1;
7:      DL2 ← get delegation level of policy2;
8:      if DL1 is greater than DL2 then
9:          return policyInstance1;
10:     else return policyInstance2; end if
11:  else invalid policies; end if
```

Цей алгоритм є ключовим для вирішення конфліктів у розподіленому середовищі, де один суб'єкт може отримувати суперечливі права доступу до одного й того ж ресурсу від різних делегуючих. Замість використання складних правил об'єднання або обмежень, він спирається на ієрархію делегування, що робить його ефективним і інтуїтивно зрозумілим.

### 3.4. Керування доступом на основі вмісту в соціальному нетворкінгу

Соціальні медіа та, зокрема, онлайн-соціальні мережі (OSN) набули значної популярності, що призвело до публікації величезних обсягів персональних даних. Як було зазначено у попередньому розділі, ці дані часто містять чутливу інформацію, що вимагає впровадження ефективних

механізмів контролю доступу для забезпечення конфіденційності користувачів. У цьому розділі ми розглянемо існуючі підходи до захисту доступу до чутливих ресурсів у соціальних медіа та запропонуємо інноваційне автоматизоване рішення, що враховує вміст та рівень конфіденційності.

### *3.4.1. Існуючі рішення та їхні обмеження*

Соціальні мережі, такі як Facebook, впровадили базові механізми контролю доступу, що дозволяють користувачам класифікувати свій контент як "публічний", "приватний", "для друзів" або "для друзів друзів. Однак, ці функції виявилися ненадійними та недостатньо зрозумілими для більшості користувачів. Крім того, налаштування цих параметрів вимагає значних зусиль, оскільки користувачам доводиться вручну визначати політики для кожного користувача та типу ресурсу.

Тому було запропоновано рішення, що враховують тип ресурсів (наприклад, фотографії, відео), застосовуючи до них правила доступу. Ці підходи, що базуються на онтологіях, дозволяють моделювати ресурси, але все ще мають суттєві обмеження:

- Класифікація ресурсів є жорсткою та фіксованою, не враховуючи їхній фактичний вміст.

- Політики застосовуються до об'єкта в цілому, що призводить до повного дозволу або повної заборони доступу, незалежно від чутливості конкретних частин контенту.

- Користувачам, які не мають досвіду в налаштуванні політик, важко керувати такими механізмами.

### *3.4.2. Пропоноване рішення*

Для подолання цих недоліків ми представляємо орієнтовану на конфіденційність схему, яка забезпечує контроль доступу на основі вмісту. Наш підхід використовує онтології для семантичного аналізу та

автоматичного виявлення чутливого контенту. Основні переваги пропонованої технології:

1. Прозорий та динамічний механізм.

Ми пропонуємо механізм, який автоматично захищає вміст повідомлень, ґрунтуючись на вимогах конфіденційності автора. Ці вимоги визначаються один раз і узагальнені, що дозволяє інтуїтивно задавати, який тип інформації та рівень деталізації є дозволеним для кожного типу контактів.

2. Деталізована оцінка конфіденційності.

На відміну від існуючих рішень, наш підхід оцінює ризик конфіденційності для кожної частини ресурсу (наприклад, кожного текстового терміну в повідомленні), а не для ресурсу в цілому. Це досягається через автоматичний процес семантичного анотування, що спирається на онтологічні бази знань (наприклад, DBPedia) та лінгвістичні інструменти.

3. Небінарний контроль доступу.

Замість повного дозволу або заборони, наша схема надає кожному типу читачів очищену версію оригінальної публікації, яка узгоджується з вимогами конфіденційності. Ці версії автоматично створюються на основі результатів семантичного аналізу та оцінки ризику.

Пропонований підхід значно спрощує управління конфіденційністю для користувачів та забезпечує більш гнучкий та точний контроль над поширенням чутливої інформації в соціальних мережах.

### 3.5. Архітектура та робочий процес системи контролю доступу

Як показано на рисунку 3.6, пропонована система контролю доступу включає трьох основних суб'єктів: автора, читача та соціальну мережу. Автор відповідає за встановлення вимог до конфіденційності та публікацію контенту. Читач ініціює запит на доступ, у відповідь на який отримує

очищену версію публікації, що відповідає вимогам конфіденційності автора. Соціальна мережа забезпечує функціонування системи, включаючи семантичне анотування та захист конфіденційності. Для цього в соціальну мережу інтегровано два ключових компоненти: анотатор та монітор.

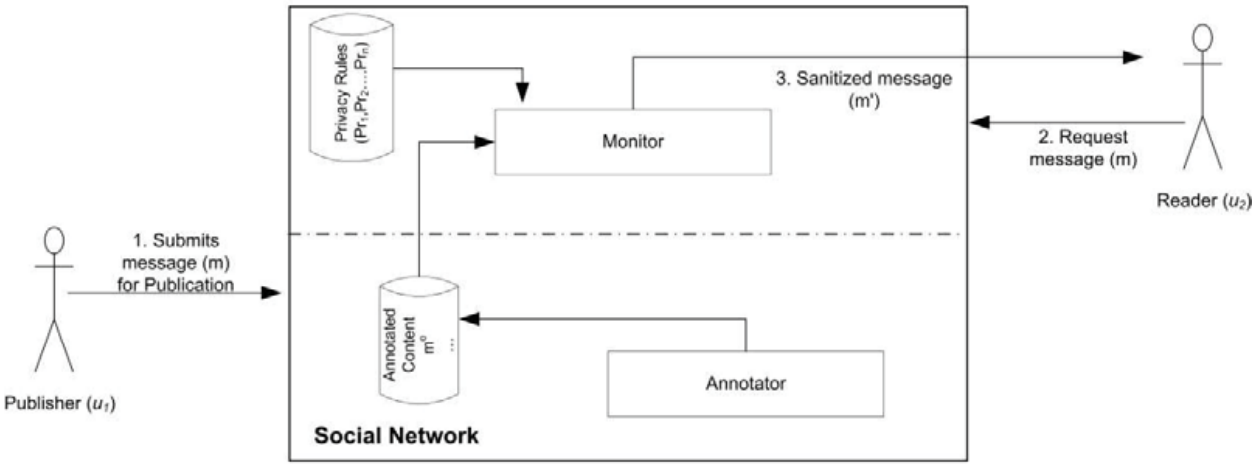


Рис. 3.6. Архітектура системи

*3.5.1. Робочий процес системи*

На етапі ініціалізації автор визначає вимоги до конфіденційності, вказуючи рівень розкриття інформації для кожного типу контактів (наприклад, "тільки родинні контакти можуть знати про мою сексуальну орієнтацію"). Ці вимоги зберігаються у базі даних правил конфіденційності, якою керує соціальна мережа. Цей процес виконується одноразово для кожного автора.

Після визначення вимог, подальший робочий процес виглядає так:

1. Обробка публікації.

Коли автор (u1) надсилає повідомлення (m) для публікації, модуль анотатора виконує синтаксичний та семантичний аналіз тексту. Анотоване повідомлення (m") зберігається у базі даних анотованого контенту.

2. Обробка запиту на доступ.

Коли читач (u2) запитує повідомлення m автора u1, модуль монітора обробляє запит. Монітор оцінює чутливість вмісту m" на основі вимог

конфіденційності u1 і очищає чутливі дані відповідно до дозволеного рівня доступу для типу контакту u2.

3. Надання доступу.

Отримане очищене повідомлення (m') надається читачеві u2.

### *3.5.2. Надсилання повідомлення для публікації*

Кожен раз, коли автор надсилає повідомлення, активується модуль анотатора. Цей модуль аналізує вміст, оскільки оцінка чутливості ґрунтується саме на ньому. Оскільки іменники зазвичай містять найбагатшу семантику та чутливі дані, анотатор зосереджується на їхньому виявленні та анотуванні. Для вирішення проблеми багатозначності слів (word-sense disambiguation), анотатор обирає найбільш релевантне значення, яке відповідає контексту повідомлення.

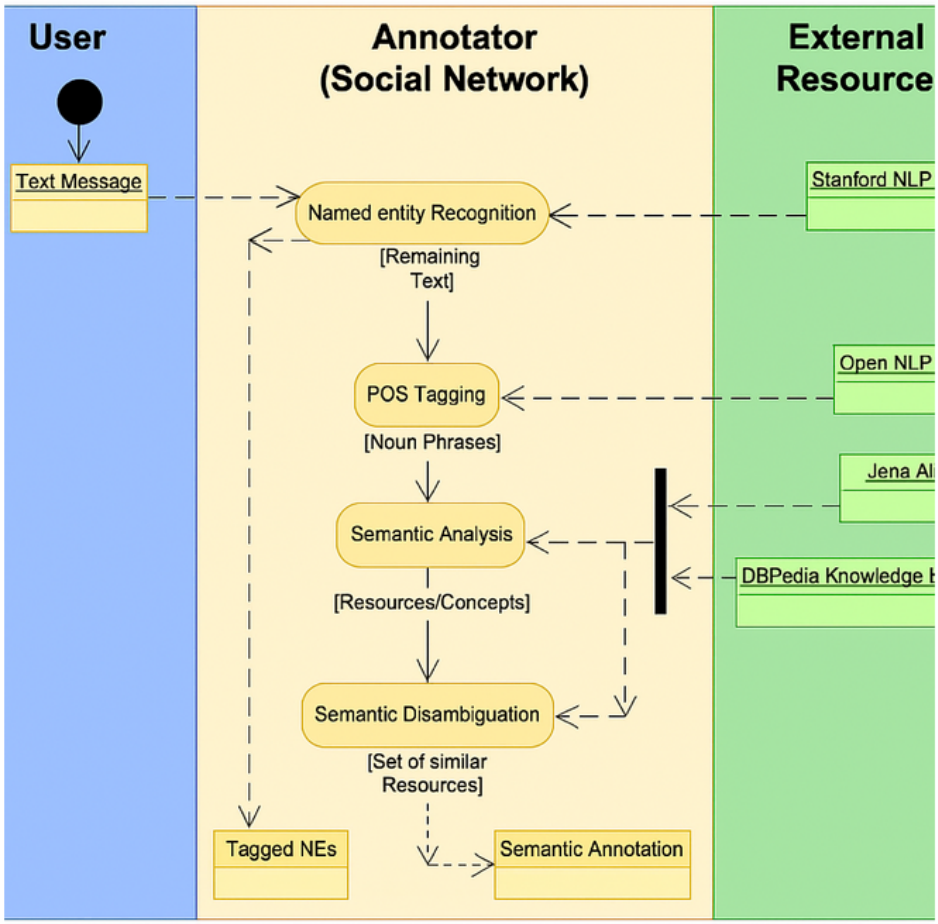


Рис. 3.7. Діаграма активності семантичної анотації

Робочий процес анотатора (рисунок 3.7) складається з наступних етапів:

1. Виявлення іменованих сутностей (NE).

Модуль розпізнавання іменованих сутностей ідентифікує власні іменники (наприклад, імена осіб, місць) та класифікує їх. Ці сутності є критичними для захисту конфіденційності, оскільки вони часто однозначно ідентифікують осіб.

2. Розмітка частин мови (POS).

Використовуючи бібліотеки обробки природної мови (OpenNLP), система виявляє загальні іменники, які можуть позначати чутливі теми (наприклад, хвороби, сексуальна орієнтація).

3. Семантичний аналіз.

Виявлені іменні фрази асоціюються з концепціями, використовуючи онтологічну базу знань DBPedia. Система виконує три кроки:

- Пошук ресурсів DBPedia за ключовими словами.
- Розширення списку ресурсів на основі їх семантичних зв'язків.
- Отримання таксономічних категорій Вікіпедії для всіх пов'язаних ресурсів.

4. Семантичне вирішення неоднозначності.

Якщо іменна фраза має кілька можливих концептуалізацій, система обирає найбільш відповідну, обчислюючи семантичну подібність між усіма можливими значеннями. Завдяки гіпотезі, що слова в реченні мають спільну тему, система вибирає комбінацію значень, яка має найменшу сукупну семантичну відстань. Семантична відстань обчислюється за формулою:

$$dist(a,b) = \log_2\left(1 + \frac{|T(a) \cup T(b) - T(a) \cap T(b)|}{|T(a) \cup T(b)|}\right)$$

де T(a) та T(b) — набори таксономічних предків концепцій.

Результат цього аналізу — семантично анотоване повідомлення — зберігається, створюючи основу для подальшої оцінки чутливості та забезпечення контролю доступу.

## 3.6. Політика контролю доступу до повідомлень в соціальному нетворкінгу

У цьому розділі ми представляємо систему контролю доступу до повідомлень у соціальних мережах (OSN), де політики визначаються власниками ресурсів. Ця система класифікує друзів та визначає дозволений рівень доступу до чутливої інформації. Її можна вважати дискреційним варіантом моделі контролю доступу на основі ролей (RBAC), оскільки класифікація користувачів аналогічна використанню ролей.

Система розроблена таким чином, щоб її можна було інтегрувати в будь-яку соціальну сереже, яка підтримує публікації. Вона включає компонент, який називається Монітор, відповідальний за авторизацію кожного запиту на доступ. Монітор обробляє запити, ґрунтуючись на трьох вхідних даних:

- Анотоване повідомлення. Текст, до якого читач хоче отримати доступ, анотований автором, співавторами та семантичними тегами.

- Класифікація читача. Тип контакту читача щодо автора.

- Вимоги конфіденційності автора. Правила, визначені автором для керування доступом до публікацій.

Монітор обробляє запит, анотуючи читача відповідно до його типу контакту (наприклад, "близькі друзі", "родина"), який визначається автором. Ця класифікація зменшує адміністративне навантаження, оскільки автор визначає правила лише один раз для групи друзів. На основі цієї анотації монітор застосовує відповідне правило конфіденційності, щоб оцінити та керувати доступом.

*3.6.1. Визначення правил доступу*

Для мінімізації адміністративних зусиль користувача система допомагає йому налаштувати вимоги конфіденційності на етапі створення облікового запису. Ці вимоги визначають правила, які містять рівні доступу до чутливих даних для різних категорій контактів. Правила представлені у вигляді кортежу:

$$rule_i \equiv <st_i, cc_i, al_i>$$

де:

sti - чутливі теми (ST) - теми, які вважаються чутливими, наприклад, згідно з законодавством про конфіденційність (наприклад, здоров'я, раса, політика).

$cc_i$ - категорії контактів (CC) - класифікації друзів, визначені користувачем (наприклад, "близькі друзі", "родина"), що ґрунтуються на рівні довіри.

$al_i$ - рівень доступу (AL) - рівень розкриття інформації, дозволений для певної категорії контакту.

У цьому прикладі ми демонструємо, як користувач може гнучко налаштовувати свої політики конфіденційності.

Приклад № 1. Користувач на ім'я Боб конфігурує свої налаштування, пов'язані з медичним здоров'ям.

Він створює Рівні доступу (AL), що містять терміни "Хвороба" та "Гепатит", і класифікує своїх друзів у категорії контактів (CC): "близькі друзі" та "родинні друзі".

- Для близьких друзів Боб встановлює рівень доступу "хвороба".

- Для родинних друзів він встановлює рівень доступу "гепатит".

У результаті, якщо Боб опублікує повідомлення, що містить інформацію про гепатит:

- Близькі друзі не отримають повний текст. Натомість, система надасть їм очищену версію, де термін "гепатит" буде замінено на більш загальний термін "хвороба".

- Родинні друзі отримають інформацію про гепатит, але без більш конкретних деталей (наприклад, тип "гепатит В" чи "гепатит С").

Ці налаштування автоматично генерують наступні правила:

*RULE1: <медичне здоров'я, близькі друзі, "хвороби">*

*RULE2: <медичне здоров'я, родинні друзі, "гепатит">*

Приклади правил, що ілюструють їхнє застосування

1. Релігія:

*RULE3: <релігія, друзі, "релігія">* — Обмежує доступ друзів до конкретних деталей про релігію автора.

*RULE4: <релігія, родинні друзі, "Мусульманин">* — Дозволяє родині знати, що автор є мусульманином, але без додаткових подробиць.

2. Сексуальність:

*RULE5: <сексуальність, друзі, null$\rangle$>* — Друзі не можуть отримати жодної інформації про сексуальність автора.

Правила визначаються на концептуальному рівні, що дозволяє системі захищати як конфіденційні дані (наприклад, хвороби), так і ідентифікаційні дані (наприклад, імена). Оскільки процес анотування може виявляти та класифікувати іменовані сутності (NE), можна створювати правила для їх захисту.

Приклади правил для іменованих сутностей:

*RULE7: <NE_особа, незнайомці, null$\rangle$>* — Забороняє незнайомцям бачити імена людей.

*RULE8: <NE_особа, родинні друзі, ім'я_особи$\rangle$>* — Дозволяє лише родині бачити конкретне ім'я.

*RULE9: <NE_місце, родинні друзі, назва_місця$\rangle$>* — Дозволяє родині бачити назви місць.

*RULE10: <NE_організація, родинні друзі, назва_організації$\rangle$>* — Дозволяє родині бачити назви організацій.

Таким чином, захист фокусується або на розкритті атрибутів (конфіденційна інформація), або на розкритті ідентичності (ідентифікаційні дані).

### *3.6.2. Забезпечення гнучкого контролю доступу*

Для забезпечення доступу до чутливих даних, система оцінює чутливість кожного терміна у повідомленні, порівнюючи його семантичні анотації з дозволеним рівнем доступу читача. Використовуючи таксономічну структуру, отриману з DBPedia, система визначає, чи є термін більш специфічним (розташований нижче в таксономічному дереві), ніж дозволений рівень. Якщо так, термін замінюється на більш загальний термін з рівня доступу (AL).

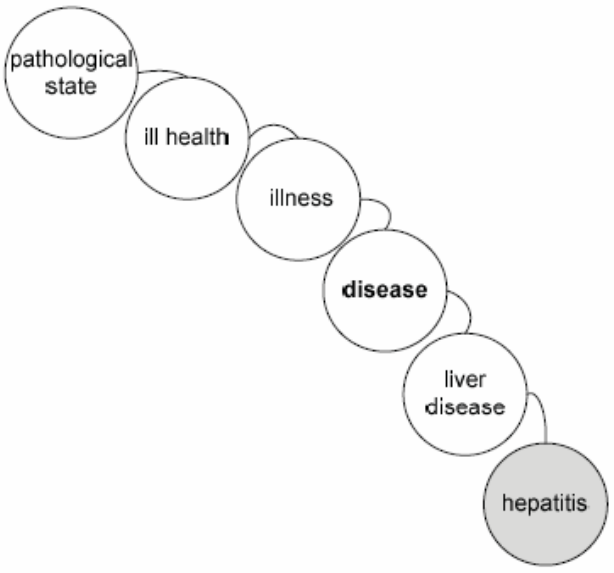


Рис. 3.8. Таксономічні узагальнення гепатиту

Приклад № 2. Якщо Боб публікує повідомлення про "гепатит", а його друг Аліса, класифікована як "близький друг", запитує доступ, монітор

бачить, що для Аліси дозволений рівень — "хвороба". Оскільки "гепатит" є більш специфічним терміном, ніж "хвороба" (як показано на рис. 6.3), система замінює "гепатит" на "хвороба", щоб захистити чутливу інформацію.

*3.6.3. Вирішення конфліктів політик у соціальних мережах*

Окрім очищення вмісту, система також обробляє потенційні конфлікти політик, які можуть виникнути, коли до публікації залучені кілька користувачів. Конфлікт виникає, коли автор публікує повідомлення на сторінці іншого користувача або позначає його, оскільки вміст може стосуватися обох. У таких випадках користувач, який був позначений, стає співавтором, і його правила конфіденційності також враховуються.

Оскільки автор та співавтор можуть мати різні правила для одних і тих же типів контактів (наприклад, "близькі друзі"), виникає ситуація конфлікту. Для вирішення цього, система застосовує найсуворіше правило з-поміж усіх, що стосуються даного читача. На практиці це означає, що буде обрано той рівень доступу, який знаходиться вище в таксономічному дереві, оскільки він є більш загальним і, відповідно, накладає найжорсткіші обмеження на розкриття інформації.

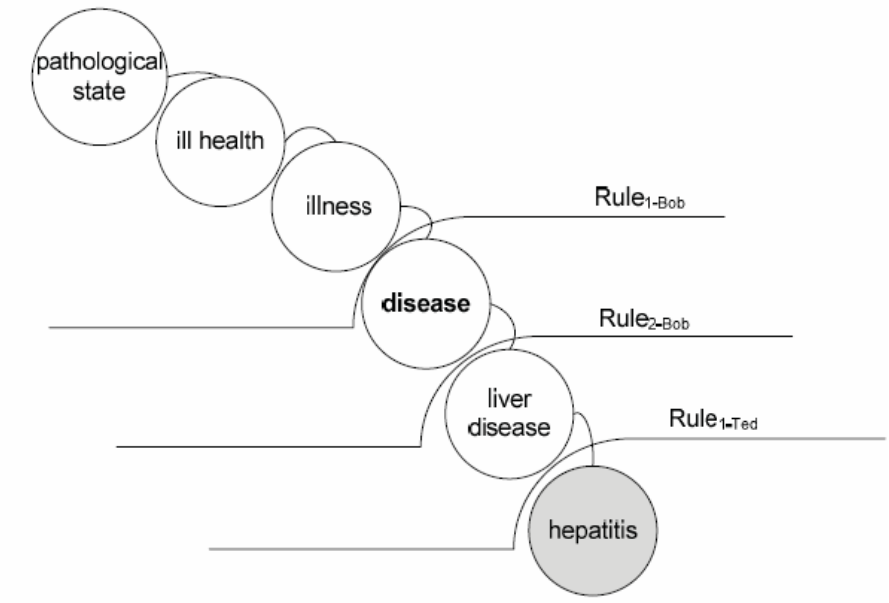


Рис. 3.9. Рівні доступу, визначені користувачами Боб і Тед

Приклад № 3. Розв'язання конфлікту

Розглянемо ситуацію, де Боб і Тед є співавторами публікації, а Аліса є їхнім спільним контактом. Рівні доступу, визначені Бобом та Тедом, показано на рисунку 3.9.

Відповідно до їхніх налаштувань, генеруються такі правила:

*RULE_1-Боб: <медичне здоров'я, незнайомці, "хвороба">*

*RULE_2-Боб: <медичне здоров'я, близькі друзі, "хвороба">*

*RULE_1-Тед: <медичне здоров'я, близькі друзі, "хвороба печінки">*

Сценарій 1: Аліса — близький друг Боба і Теда

Коли Аліса запитує доступ до публікації, виникає конфлікт між RULE_2-Боб та RULE_1-Тед. Щоб задовольнити вимоги обох користувачів, монітор порівнює їхні рівні доступу: "хвороба" (Боб) та "хвороба печінки" (Тед). Оскільки "хвороба" є більш загальним терміном і знаходиться вище в таксономічному дереві, він вважається суворішим. Отже, система очистить вміст публікації для Аліси, замінивши специфічні терміни на "хвороба".

Сценарій 2: Аліса — незнайомець для Боба, але близький друг для Теда

У цьому випадку для Аліси діють два правила: RULE_1-Боб ("хвороба") та RULE_1-Тед ("хвороба печінки"). Застосовуючи ту ж стратегію, система знову обирає більш суворе правило, тобто "хвороба". Таким чином, вміст повідомлення буде очищено, і Аліса побачить лише загальний термін "хвороба".

## 3.7. Практична імплементація та масштабованість системи

У цьому розділі ми представляємо практичну імплементацію нашої системи, аналізуємо масштабованість її модулів та демонструємо її

функціонування на прикладі спеціалізованої соціальної мережі в галузі охорони здоров'я.

Важливо наголосити, що єдиною взаємодією, яку система вимагає від користувача, є визначення вимог конфіденційності. Цей процес виконується одноразово під час ініціалізації облікового запису. На основі цих вимог генеруються правила, а автоматична оцінка чутливої інформації за допомогою семантичного анотування забезпечує прозорий та автоматизований контроль доступу до всіх наступних публікацій. Вимоги можуть ґрунтуватися на чутливих темах (ST), визначених у поточному законодавстві. Кількість рівнів доступу (AL) для кожної теми залежить від кількості категорій контактів (CC) у соціальній мережі, яка в середньому становить три. Таким чином, зусилля користувача мінімізуються.

Для ілюстрації порівняємо наш підхід зі стандартними методами. У стандартному підході користувач має:

1) вручну оцінити чутливість кожного нового повідомлення;

2) створити кілька очищених версій для різних типів контактів;

3) визначити правила доступу для кожної версії.

Враховуючи, що середньостатистичний користувач Facebook публікує 90 одиниць контенту на місяць, з яких 58% потребують налаштувань конфіденційності, йому довелося б вручну захищати близько 52 публікацій. За наявності трьох типів контактів це призвело б до створення 156 очищених версій і 156 правил доступу. Натомість наш підхід вимагає лише одноразового визначення 18 рівнів доступу (6 чутливих тем x 3 типи контактів).

Для демонстрації ми використовуємо приклад медичної соціальної мережі, де захист конфіденційності зосереджений на темі здоров'я. Контакти користувачів класифіковані на три групи (CC): "Клініцисти/Дослідники", "Підписники" та "Зареєстровані користувачі". Рівні доступу (AL) для цих груп пов'язані з різними рівнями розкриття медичного стану.

Набори AL та CC для цієї мережі:

*AL={(ВІЛ/СНІД/Гепатит/ЗПСШ),Інфекції,погане здоров'я,Стан/Умова}*

*CC={(Клініцисти/Дослідники),Підписники,Зареєстровані користувачі}*

Користувач може інтуїтивно налаштувати свої вимоги, призначивши AL кожному елементу CC.

У цьому прикладі:

- Для клініцистів/дослідників дозволено розкриття специфічної інформації про хвороби (ВІЛ/СНІД).

- Для підписників — лише загальне поняття "інфекції".

- Для зареєстрованих користувачів — лише "погане здоров'я".

Ці вимоги формалізуються в наступні правила:

*rule1=<Медичне здоров'я,Клініцисти/Дослідники,ВІЛ>*

*rule2=<Медичне здоров'я,Клініцисти/Дослідники,СНІД>*

*rule3=<Медичне здоров'я,Клініцисти/Дослідники,Гепатит>*

*rule4=<Медичне здоров'я,Клініцисти/Дослідники,ЗПСШ>*

*rule5=<Медичне здоров'я,Підписники,Інфекції>*

*rule6=<Медичне здоров'я, Зареєстровані користувачі, Погане здоров'я>*

Коли автор публікує повідомлення, модуль анотатора виконує семантичний аналіз. Спочатку повідомлення обробляється для розмітки частин мови.

Далі відбувається семантичне анотування. Масштабованість процесу залежить від кількості іменних фраз у повідомленні. Час, необхідний для отримання можливих значень (концептуалізацій) для кожної фрази з DBPedia за допомогою SPARQL-запитів, є критичним показником.

Як видно, час виконання є лінійним щодо кількості іменних фраз (середня вартість на фразу — 0.25 мс), що свідчить про високу масштабованість процесу анотування.

Отримані значення проходять процес семантичного розв'язання неоднозначностей, який не вимагає додаткових запитів, а лише попарної оцінки вже наявних таксономій.

Після анотування монітор обробляє запити на доступ. Він оцінює чутливість кожного терміна, порівнюючи його семантичну анотацію з рівнем доступу (AL), дозволеним для читача. Якщо термін знаходиться нижче в таксономічному дереві, ніж AL, він вважається чутливим і підлягає очищенню.

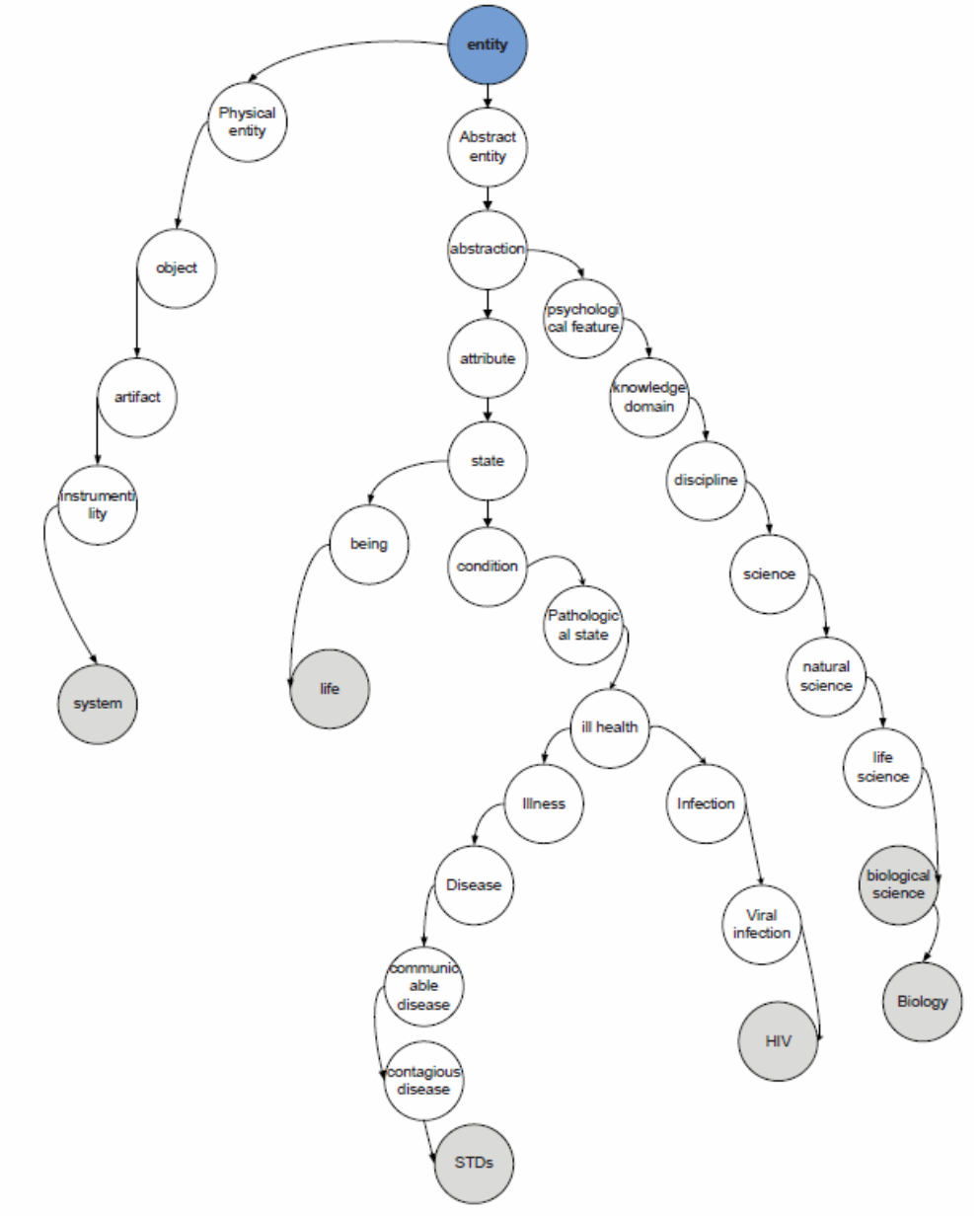


Рис. 3.10. Таксономічне дерево значень для екземпляру повідомлення

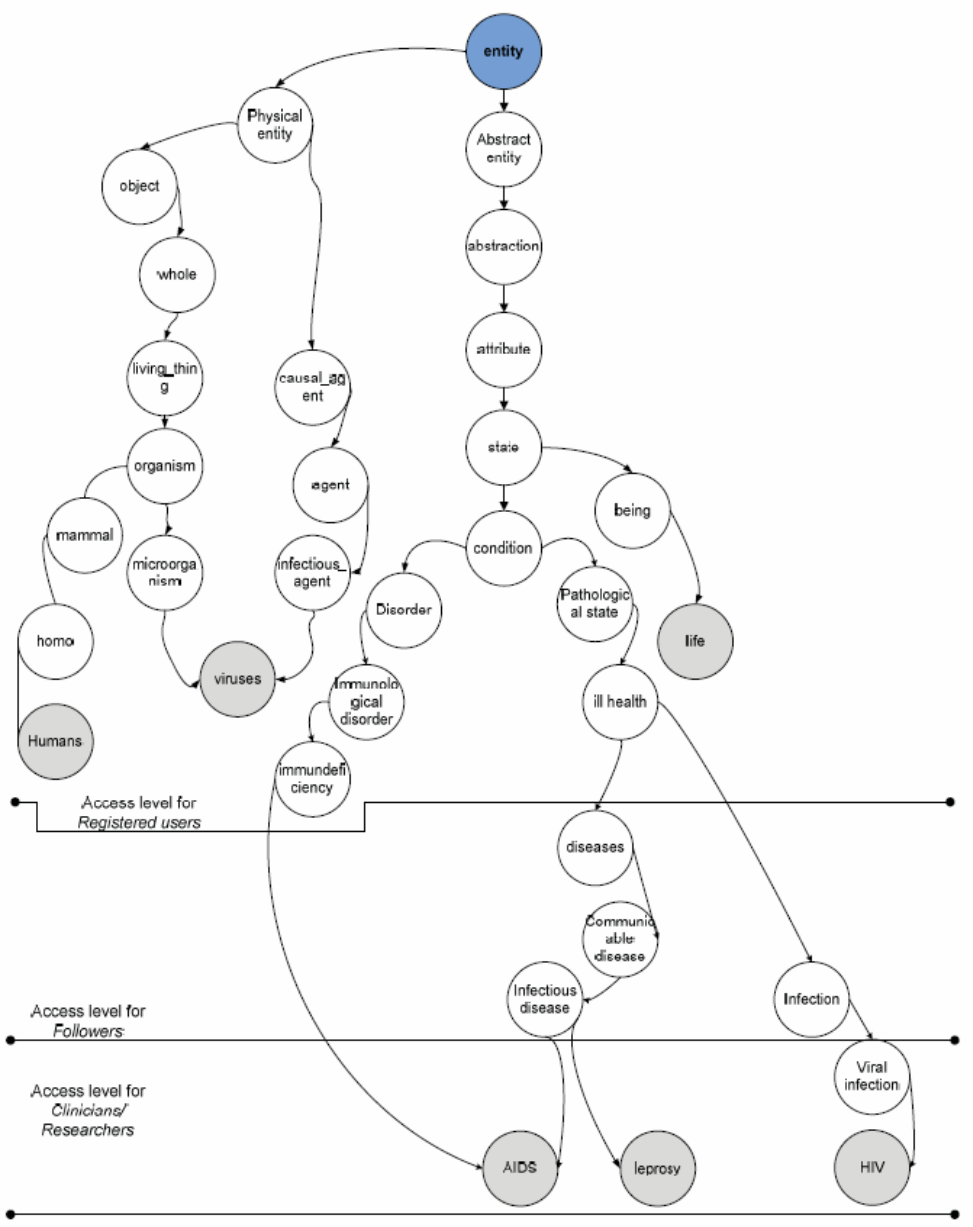


Рис. 3.11. Таксономії DBPedia та рівні доступу фраз у екземплярі повідомлення

Описаний процес семантичного розв'язання неоднозначностей є ключовим етапом у системі анотування, оскільки він дозволяє встановити справжню семантику повідомлення. Це досягається шляхом обчислення семантичної відстані між різними значеннями іменних фраз, що містяться в тексті.

Процес полягає в наступному.

- Ідентифікація значень. Для кожної іменної фрази, виявленої в повідомленні, система отримує набір потенційних значень (senses) з онтологічної бази знань DBPedia.

- Обчислення відстані. Використовуючи таксономічну структуру DBPedia (як показано на рис. 3.10), система обчислює семантичну відстань між кожним значенням однієї іменної фрази та значеннями всіх інших іменних фраз у повідомленні.

Наприклад, для зразкового повідомлення про ВІЛ система згенерує різні версії для різних категорій контактів:

- lля підписників (з rule5): Дозволений рівень "Інфекції". Оскільки терміни "ВІЛ" та "СНІД" знаходяться нижче в таксономії (рисунок 3.11), ніж "Інфекції", вони будуть замінені на "інфекції" та "інфекційна хвороба".

Для зареєстрованих користувачів (з rule6): Дозволений рівень "Погане здоров'я". Усі специфічні терміни будуть замінені на "погане здоров'я".

Для клініцистів/дослідників (з rule1−4): Дозволено доступ до всіх деталей, тому очищення не відбувається.

Процес очищення також є високоефективним і масштабованим, оскільки залежить від обмеженої кількості анотацій та типів контактів. Очищені версії можуть бути кешовані для подальших запитів, що додатково підвищує ефективність системи.

## Висновки до розділу

У третьому розділі здійснено імплементацію онтологічних моделей у хмарних засобах соціального нетворкінгу з акцентом на механізмах делегування прав доступу. Показано, що делегування у динамічних середовищах супроводжується низкою проблем, пов’язаних із конфліктами політик, верифікацією прав та відкликанням делегованих повноважень.

Розроблено онтологічну модель делегування, яка забезпечує інтеграцію атрибутивного підходу (ABAC) із сутностями хмарного середовища.

Запропоновано формалізацію робочих процесів делегування, включаючи їх онтологічне представлення, а також механізм вирішення конфліктів політик. Особлива увага приділена алгоритму об’єднання політик, що дозволяє знизити ймовірність суперечностей між правилами доступу у складних багатокористувацьких системах.

# ВИСНОВКИ

У ході виконання магістерської роботи було проведено комплексне дослідження проблеми побудови та застосування онтологічних моделей контролю доступу в умовах хмарних середовищ соціального нетворкінгу. На основі проведеного аналізу, теоретичних узагальнень і практичної розробки сформульовано такі основні результати й висновки:

Дослідження предметної області показало, що традиційні моделі контролю доступу (DAC, MAC, RBAC) не відповідають вимогам сучасних розподілених і динамічних середовищ. Вони не забезпечують належної гнучкості у делегуванні прав, масштабованості в управлінні ресурсами та повноцінного захисту конфіденційності у контексті соціальних мереж і хмарних систем.

Встановлено ключові виклики для сучасних систем управління доступом, серед яких – високий рівень загроз конфіденційності, складність забезпечення сумісності між різними політиками, а також потреба у гнучких і семантично зрозумілих моделях управління. Доведено, що використання онтологій здатне подолати ці виклики завдяки їх можливості формалізувати знання про сутності та відношення у системах.

Розроблено онтологічну модель контролю доступу на основі атрибутів (ABAC), яка забезпечує універсальність і розширюваність, дозволяючи інтегрувати нові сутності та політики без суттєвих змін у системі. Модель є адаптивною до специфіки як соціальних мереж, так і хмарних платформ, що свідчить про її універсальний характер.

Запропоновано онтологічний підхід до делегування доступу у хмарних середовищах, що враховує проблеми конфліктів політик, верифікації та відкликання прав. Розроблено алгоритм об’єднання політик, який підвищує узгодженість правил і знижує ризик суперечностей у багатокористувацьких системах.

Імплементовано приклад онтологічної моделі для управління доступом у соціальних мережах, зокрема на основі контенту повідомлень. Це рішення враховує контекст публікацій і міжкористувацькі відносини, забезпечуючи більшу точність у визначенні політик доступу.

Розроблено архітектуру системи контролю доступу для хмарних засобів соціального нетворкінгу, що інтегрує онтологічні підходи, механізми делегування та управління на основі вмісту. Така архітектура забезпечує комплексність, гнучкість і масштабованість процесів управління доступом.

Практичне значення роботи полягає у можливості використання запропонованих онтологічних моделей та методів у реальних хмарних платформах і соціальних мережах. Вони здатні підвищити рівень захисту інформації, гнучкість управління ресурсами та довіру користувачів до систем соціального нетворкінгу.

Таким чином, у роботі теоретично обґрунтовано та практично реалізовано онтологічні моделі контролю доступу, які довели свою ефективність для вирішення задач захисту конфіденційності, гнучкого делегування прав та управління доступом у складних динамічних середовищах. Отримані результати мають як наукову, так і прикладну цінність та можуть слугувати основою для подальших досліджень у напрямі розвитку інтелектуальних систем управління доступом.

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1. Aïmeur, E., & Schonfeld, R. (2010). "Towards a Privacy-Enhanced Social Networking Site." У збірнику праць: Proceedings of the 2010 International Conference on Computer Systems and Applications (AICCSA). Стор. 1-8. IEEE.
2. Batet, M., Sánchez, D., & Moreno, M. L. (2014). "Semantic Similarity in DBPedia: An Experimental Study." Journal of Universal Computer Science, 20(3), стор. 306-328.
3. Brain Research Institute (BRI). (2015). "Social Media Usage and User Behavior Trends." Technical Report.
4. Carminati, B., Ferrari, E., & Pernici, B. (2009). "A Privacy-Aware Access Control Model for Social Network Services." У збірнику праць: Proceedings of the 2009 International Conference on Advances in Social Networks Analysis and Mining. Стор. 182-189. IEEE.
5. Carminati, B., & Ferrari, E. (2011). "User-Centric Access Control for Social Networks." IEEE Transactions on Knowledge and Data Engineering, 23(1), стор. 1-13. IEEE.
6. Cheng, R., Zhang, J., & Chen, G. (2012). "Privacy Preserving Data Publishing." У книзі: Data Privacy and Security. Springer.
7. DoHNY (New York State Department of Health). (2013). "Confidentiality and Disclosure of HIV-Related Information." Policy and Guidance Document.
8. Eecke, P. V. (2010). "The Right to be Forgotten." У збірнику праць: Proceedings of the Privacy and Identity Management Workshop.
9. European Union. (1995). "Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data." Official Journal of the European Communities.
10. Finkel, J. R., Grenager, T., & Manning, C. (2005). "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling."

Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics. Стор. 363-370.

11. Harispe, S., T. S., & Le-Cun, Y. (2014). "Semantic Similarity from Lexical Taxonomies." Artificial Intelligence Review, 42(3), стор. 235-251.
12. Health Insurance Portability and Accountability Act (HIPAA). (1996). Public Law 104-191. U.S. Government Printing Office.
13. Imran-Daud, M., Sánchez, D., & Viejo, L. A. (2016). "Ontology-based Access Control Management: Two Use Cases." Proceedings of the 11th International Conference on Evaluation of Novel Approaches to Software Engineering. Стор. 195-206. SciTePress.
14. Jena, M. K., & Dash, M. K. (2014). "Role-Based Access Control in Cloud Computing." International Journal of Computer Science and Technology, 5(2), стор. 15-20.
15. Johnson, T. (2012). "Privacy in Social Networks: A Survey of Recent Research." ACM Computing Surveys, 44(4), стор. 1-32.
16. Kilgarriff, A. (2000). "Using corpora to investigate privacy." International Journal of Corpus Linguistics, 5(1), стор. 45-63.
17. Kissmetrics. (2015). "Social Media Demographics and Statistics." Market Report.
18. Lehmann, J., I. S., & Völkel, M. (2014). "DBPedia: A Multilingual Cross-Domain Knowledge Graph." Semantic Web, 5(5), стор. 351-365.
19. Liu, K., & Terzi, E. (2011). "Towards a Privacy-Aware Data Publication Model." ACM Transactions on Database Systems, 36(3), стор. 1-44.
20. Masoumzadeh, A., & Ghandeharizadeh, S. (2010a). "A Framework for Policy-Based Access Control in Online Social Networks." У збірнику праць: Proceedings of the 2010 International Conference on Social Computing. Стор. 121-128. IEEE.
21. Masoumzadeh, A., & Ghandeharizadeh, S. (2010b). "Enforcing Privacy Policies on User-Generated Content in Social Networks." IEEE Transactions on Knowledge and Data Engineering, 22(10), стор. 1424-1437.

22. OpenNLP. (2010). "Apache OpenNLP: A Machine Learning-based Toolkit for the Processing of Natural Language Text." Apache Software Foundation.
23. Pew Research Center. (2010). "The Rise of Social Networking." Pew Internet & American Life Project Report.
24. Sánchez, D., Batet, M., & Masip, D. (2012). "Semantic similarity for user profile matching in social networks." Expert Systems with Applications, 39(12), стор. 10972-10981.
25. Sánchez, D., Batet, M., & Masip, D. (2013a). "A Semantic-based Access Control Model for Social Networks." У збірнику праць: Proceedings of the 2013 International Conference on Social Computing.
26. Sandhu, R. S., B. N., & M. S. (1996). "Role-Based Access Control Models." IEEE Computer, 29(2), стор. 38-47.
27. Adkins, A. (2015). "Privacy and Personalization: A Dual-Sided Coin in Digital Marketing." Journal of Marketing Research, 52(6), стор. 817-832.
28. Baker, J. (2014). "Semantic Web Technologies for Access Control." International Journal of Computer Science Issues, 11(2), стор. 15-24.
29. Collins, R. (2016). "Discretionary Access Control in Modern Computing." ACM Transactions on Information Systems Security, 19(4), стор. 1-25.
30. D'Amico, L. (2013). "Privacy Preserving Social Media Data Analytics." Journal of Privacy and Confidentiality, 5(1), стор. 11-28.
31. Epstein, H. (2015). "User-Generated Content and Privacy Risks." IEEE Security & Privacy, 13(4), стор. 45-53.
32. Green, T. (2017). "The Role of Ontologies in Data Protection." International Journal of Semantic Computing, 11(1), стор. 1-15.
33. Harper, L. (2012). "Social Network User Behavior and Privacy Concerns." Journal of Computer-Mediated Communication, 17(3), стор. 297-313.
34. Jackson, P. (2011). "Scalability of Privacy-Preserving Systems." ACM Transactions on the Web, 5(1), стор. 1-19.
35. Patel, K. (2014). "Access Control Policies for Complex Data." Journal of Information Security, 5(4), стор. 121-135.

36. Quinn, S. (2013). "Privacy in Online Health Communities." Journal of Medical Internet Research, 15(7), e132.
37. Smith, J. (2015). "Automated Content Analysis in Social Media." Communications of the ACM, 58(8), стор. 55-63.
38. Taylor, K. (2016). "Semantic Disambiguation in Textual Analysis." ACM Transactions on Speech and Language Processing, 12(2), стор. 1-20.
39. Wilson, R. (2014). "The Use of DBPedia for Semantic Web Applications." International Journal of Web Engineering and Technology, 9(1), стор. 1-17.