УДК 004.056

В. Л. Поліщук
І. В. Богач

РЕАЛІЗАЦІЯ ГНУЧКОЇ МОДЕЛІ ІЄРАРХІЧНОГО РОЗМЕЖУВАННЯ ДОСТУПУ ДО РЕСУРСІВ У ПРОМИСЛОВИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Вінницький національний технічний університет

Анотація

Розроблено інформаційну технологію гнучкого керування доступами для систем автоматизації виробництва. Досліджено архітектуру сутностей-агрегаторів, що поєднують контроль над компонентами інтерфейсу користувача та REST-ендпойнтами. Описано ієрархічну структуру дозволів у вигляді n-арного дерева та алгоритм трансформації шаблонних прав у персоналізовані налаштування користувача.

Ключові слова: керування доступами, ієрархічна структура, ComponentAccess, Permission, REST API, автоматизація виробництва.

Abstract

An information technology for flexible access control has been developed for industrial automation systems. The architecture of aggregator entities, which combine control over user interface components and REST endpoints, has been investigated. A hierarchical structure of permissions in the form of an n-ary tree and an algorithm for transforming template permissions into personalised user settings are described.

Keywords: access management, hierarchical structure, ComponentAccess, Permission, REST API, production automation.

Вступ

У системах автоматизації виробничих процесів, які зазвичай є закритими та критично важливими, класичного розмежування на основі статичних ролей часто недостатньо. Необхідність гранульованого контролю виникає не лише на рівні серверних запитів, а й на рівні окремих елементів інтерфейсу (кнопок, компонентів, вкладок). Це вимагає розробки гнучкої архітектури, яка б дозволяла динамічно змінювати права доступу без переписування програмного коду, адаптуючись до специфіки конкретного підприємства[1].

Результати дослідження

У результаті дослідження було розроблено та реалізовано інформаційну технологію гнучкого керування доступами для систем автоматизації виробничих процесів, що базується на архітектурі сутностей-агрегаторів. Основним завданням розробленого модуля є забезпечення гранульованого контролю як на рівні серверних кінцевих точок (REST-ендпойнтів), так і на рівні компонентів користувацького інтерфейсу (UI).

Для розмежування доступу на загальному рівні впроваджено систему повноважень, що включає ролі: ADMIN (повний доступ), DATA_MANAGER (керування даними підприємства), USER (доступ до інтерфейсу клієнта) та ANONYMOUS (для автентифікації та валідації токенів).

Ключовим елементом технології є сутність ComponentAccess, яка призначена для обмеження доступу до елементів структури HTML DOM. Структура компонентів представлена у вигляді *n*-арного дерева, що дозволяє керувати доступом на різних рівнях: від глобальних модулів і вкладок до окремих функціональних кнопок. Кожен вузол дерева містить набір прапорців для CRUD-операцій (enableCreate, enableRead, enableUpdate, enableDelete), а також унікальний ключ для ідентифікації React-компонента.

Для контролю доступу до серверної частини використано сутності EndpointAccess та EndpointGroup. Вони забезпечують централізований механізм перевірки прав на виконання HTTP-запитів (GET, POST,

PUT, DELETE), дозволяючи логічно групувати ендпойнти за функціональним призначенням.

Сутність-агрегатор Permission (рис.1) відображає ролі користувачів та об’єднує ієрархії ComponentAccess з групами EndpointGroup.

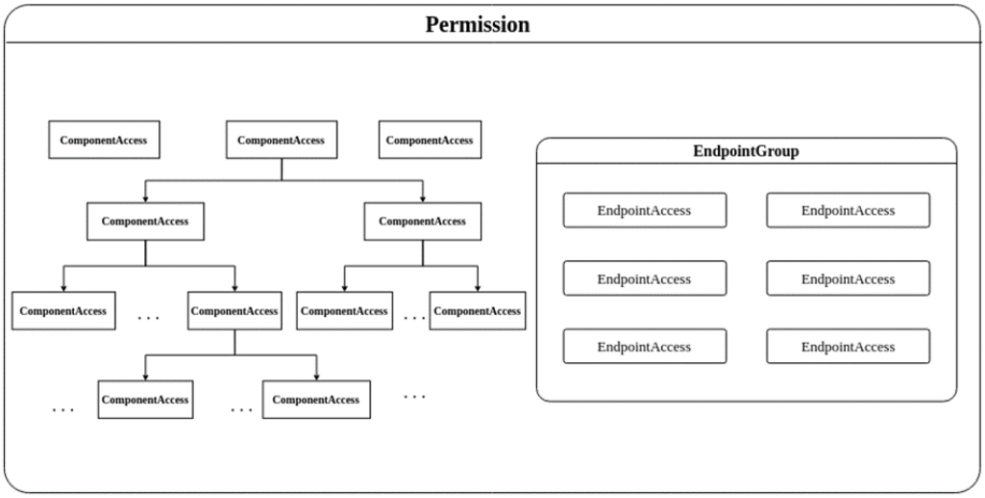

Рис.1. Композиція сутностей керування доступом

Для логічного поділу шаблонів дозволів та для зручності оперуванням сутностями було впроваджено три рівні ієрархії шаблонів дозволів:

- ADMIN level: глобальні шаблони без прив'язки до конкретних підприємств або користувачів;
- DATA_MANAGER level: шаблони, що існують у межах конкретного підприємства (enterprise);
- USER level: персоналізовані дозволи, що мають безпосередній зв'язок із конкретним користувачем.

Центральною ланкою модуля є сутність Permission, яка виконує наступні функції[2,3]:

- Агрегація: об'єднує права доступу до UI та API в єдиний профіль користувача.
- Ієрархічний поділ: підтримує три рівні доступу — ADMIN, DATA_MANAGER та USER, кожен з яких має свою логіку прив’язки до підприємства (enterprise).
- Шаблонізація: дозволяє створювати еталонні набори прав для різних категорій працівників.

Для автоматизації процесу призначення прав розроблено алгоритм перетворення шаблону на персоналізований дозвіл користувача, який включає наступні кроки:

1. Поверхневе копіювання екземпляра Permission та встановлення зв'язку з батьківським шаблоном.
2. Глибоке копіювання об'єктів EndpointGroup та всіх вкладених EndpointAccess із деактивацією ознаки шаблону (template = FALSE).
3. Рекурсивне копіювання *n*-арного дерева ComponentAccess зі збереженням ієрархії та переведенням об’єктів у статус робочих копій.
4. Агрегація та збереження: призначення створених копій до об'єкта Permission та їх фіксація в базі даних.

Такий підхід дозволяє кожному користувачу мати унікальну конфігурацію прав, яка може динамічно оновлюватися при зміні батьківського шаблону. На фронтенді отримані структури дозволів використовуються для рендерингу елементів інтерфейсу та превентивного блокування недоступних REST-запитів.

Технологічна реалізація запропонованого механізму передбачає використання спеціалізованого шару об'єктів перенесення даних (DTO), що забезпечують уніфіковану трансляцію інформації про права доступу між різними рівнями системи. Ці структури виступають проміжною ланкою для передачі стану ієрархічних моделей компонентів та ендпойнтів, дозволяючи бекенду ефективно виконувати валідацію запитів, а фронтенду – динамічно адаптувати графічний інтерфейс відповідно до актуальних дозволів користувача.

Логічний поділ сутностей керування доступом наведено на рисунку 2. UML-діаграма шару DTO сутностей керування доступом наведено на рисунку 3.

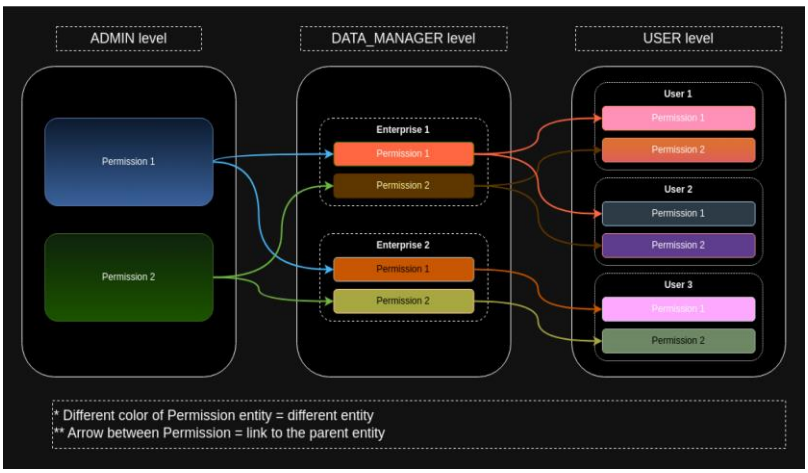

Рис. 2. Логічний поділ сутностей керування доступом

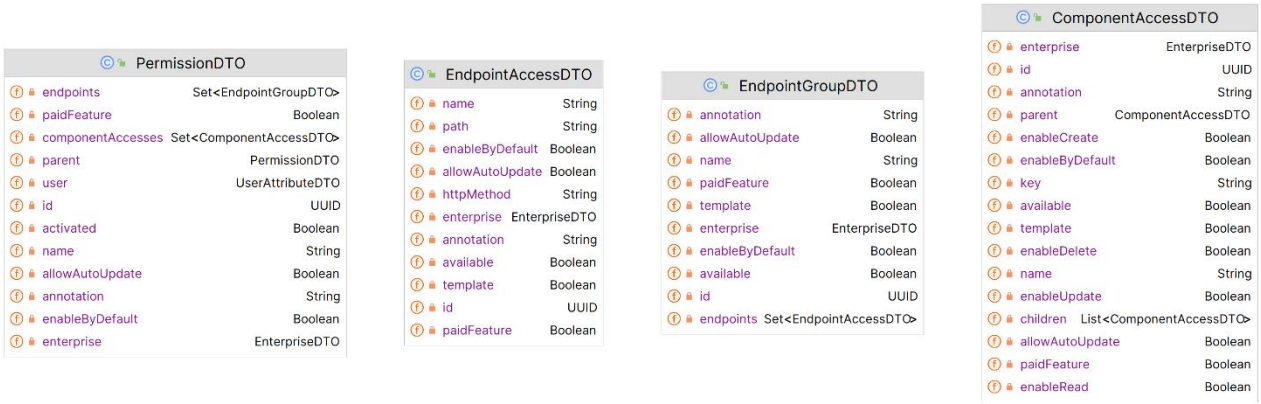

Рис. 3. UML-діаграма шару DTO сутностей керування доступом

Розроблений модуль було протестовано для перевірки коректності роботи та відповідності вимогам безпеки. Тестування виконувалося за допомогою Postman, що дозволило перевірити правильність обробки HTTP-запитів, видачі JWT-токенів та контролю доступу до ресурсів. Додатково проведено модульне тестування за допомогою JUnit5.

Висновки

Запропонована модель агрегації дозволів забезпечує високу гнучкість управління промисловими системами. Використання ієрархічних структур для UI-компонентів та механізму рекурсивного копіювання шаблонів дозволяє централізовано контролювати безпеку системи на всіх рівнях — від клієнтського інтерфейсу до серверної бізнес-логіки. Це сприяє надійному розмежуванню повноважень та спрощує аудит прав доступу в межах великих підприємств.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ragothaman, K., Wang, Y., Rimal, B., & Lawrence, M. (2023). Access Control for IoT: A Survey of Existing Research, Dynamic Policies and Future Directions. Sensors, 23(4), 1805.
2. Access Control Models – Escape.tech. URL: https://escape.tech/blog/access-control-models (дата звернення 23.05.2026).
3. Attribute-Based Access Control (ABAC). National Institute of Standards and Technology (NIST). URL: https://csrc.nist.gov/Projects/Attribute-Based-Access-Control (дата звернення: 25.05.2026).

Поліщук Володимир Леонідович – студент групи 1КН-25М, кафедра комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: volodimirpolishchuck@gmail.com

Богач Ілона Віталіївна – к.т.н., доцент, асистент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: ilona.bogach@gmail.com

Polishchuk Volodymyr Leonidovich – student of 1KN-25M group, Department of Computer Science, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: volodimirpolishchuck@gmail.com

Bogach Ilona Vitaliivna – PhD in Engineering, Associate Professor, Assistant at the Department of Computer Science, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com