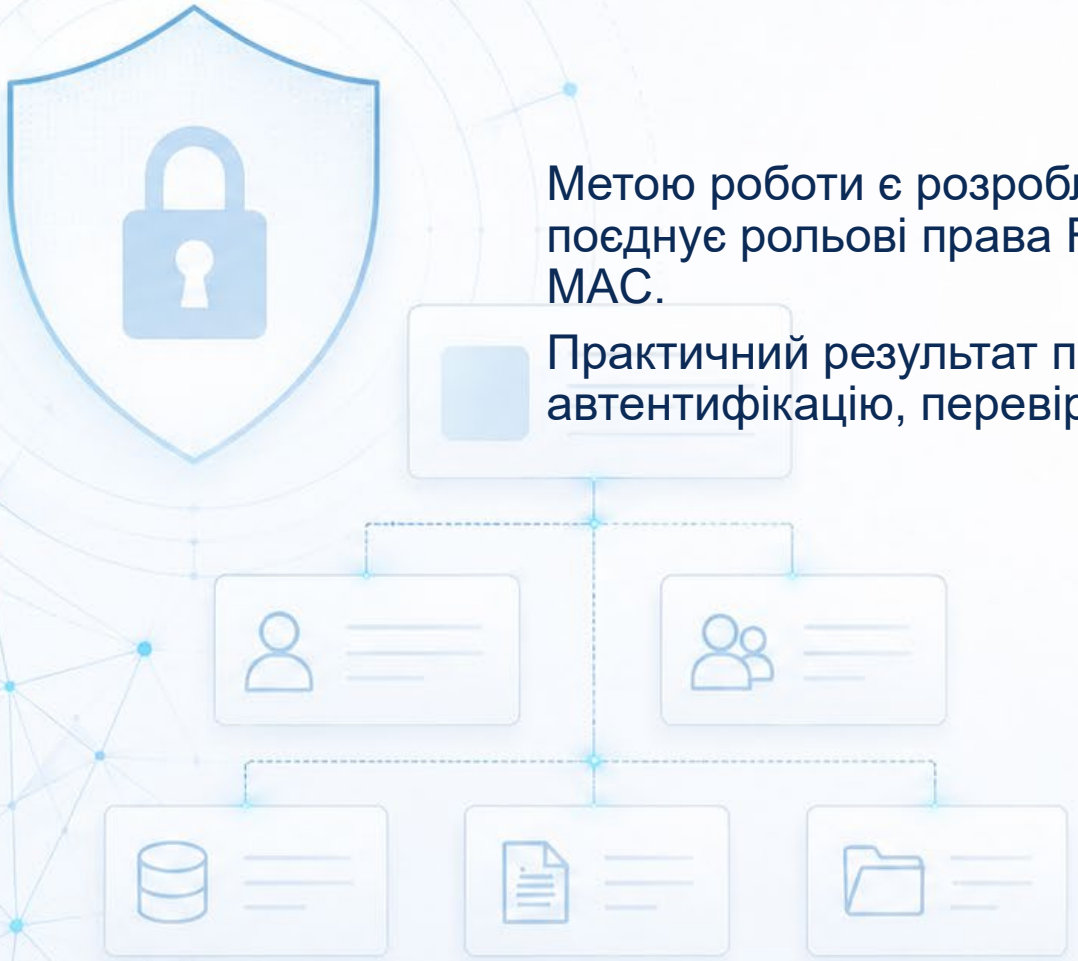
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ

Інтеграція та модернізація рольової моделі доступу в інформаційно-комунікаційній системі підприємства

Виконав: Дмитрієв Артем · 123 «Комп’ютерна інженерія»
Керівник: Ізмайлова О.В.

Мета та практичний результат

Робота орієнтована на створення не лише моделі, а й працюючого прототипу

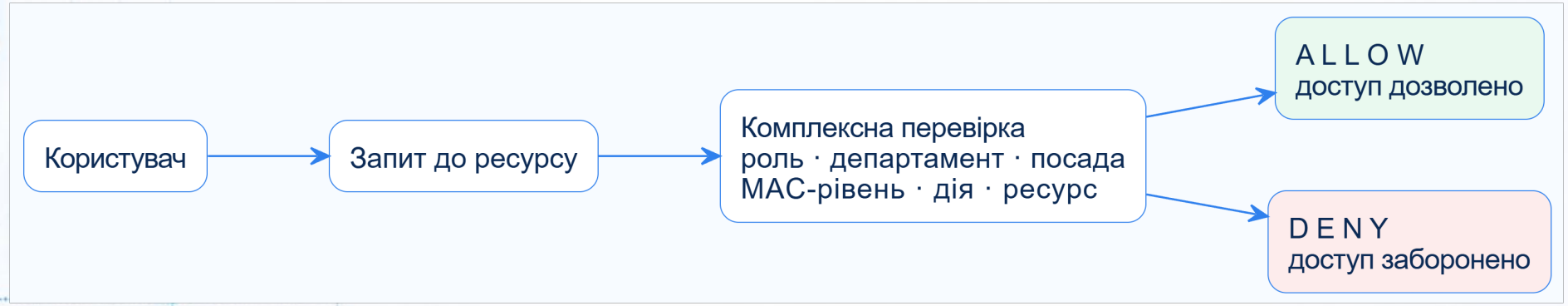


Метою роботи є розроблення гібридної системи керування доступом, яка поєднує рольові права RBAC з обов’язковим контролем рівня допуску MAC.

Практичний результат полягає у створенні web-прототипу, де реалізовано автентифікацію, перевірку доступу, різні ролі користувачів і журнал аудиту.

Проблема керування доступом у бізнес-системі

Для прийняття рішення недостатньо лише логіна і пароля



Контекст користувача
роль · департамент · посада
MAC-рівень

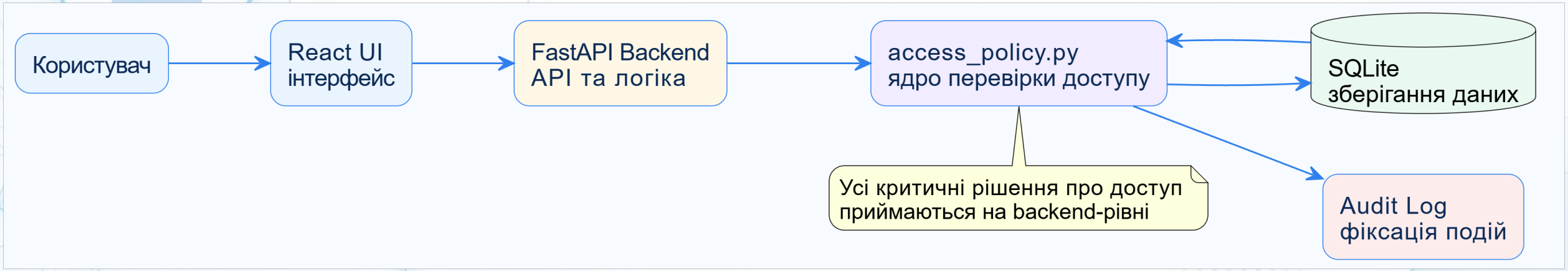
Контекст ресурсу
департамент-власник
рівень класифікації

Фінальне рішення
ALLOW лише після проходження
усіх обов’язкових перевірок

Ключова ідея: авторизація — це перевірка конкретної дії в конкретному контексті.

Загальна архітектура системи

Система реалізована як web-застосунок із чітким поділом ролей компонентів



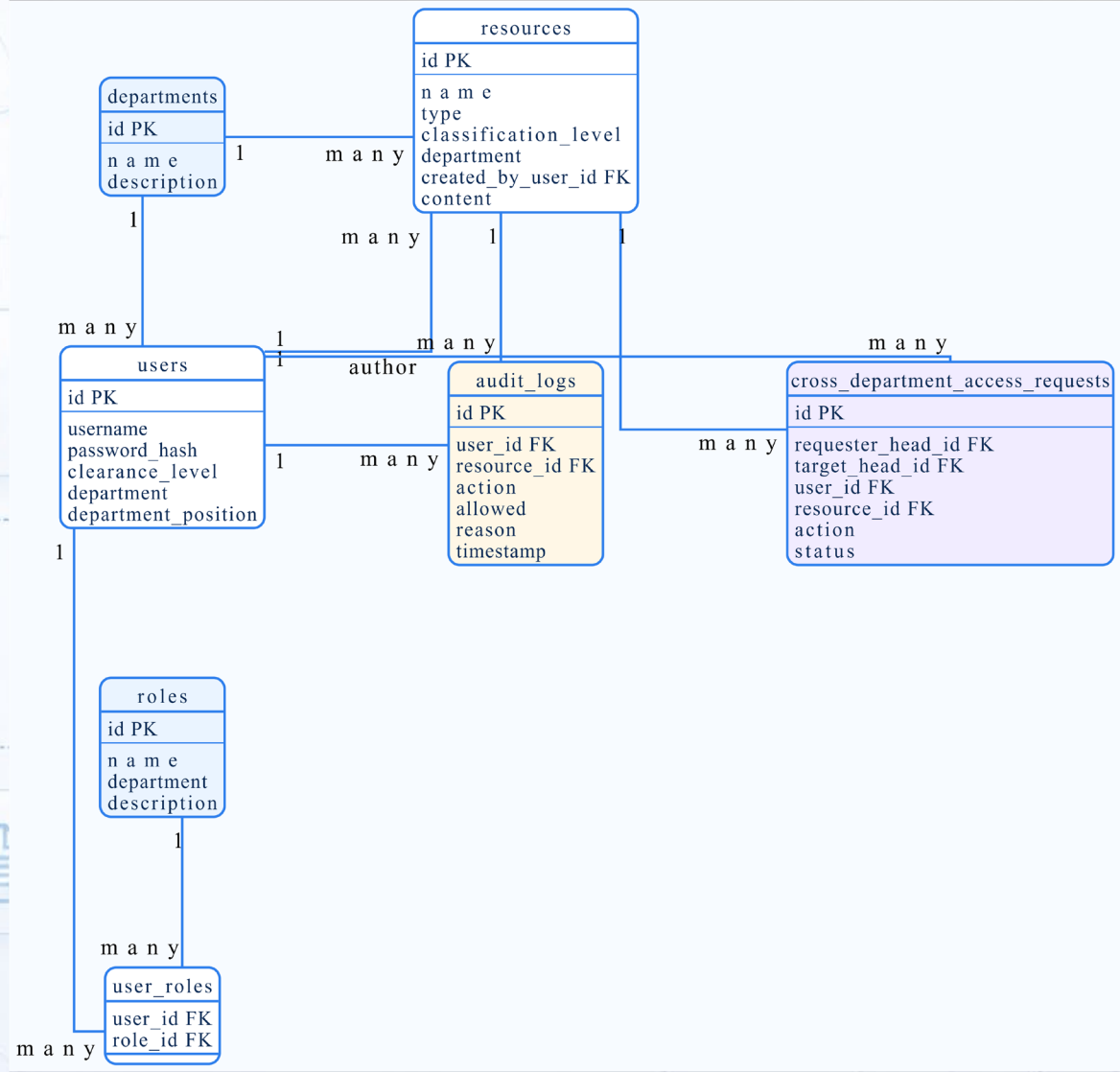
Frontend
показує панелі та кнопки,
але не приймає рішення доступу

Backend / access_policy.py
виконує RBAC, MAC,
департаментні та посадові перевірки

SQLite + Audit
зберігає політики, ресурси
та фіксує результат кожної дії

Структура бази даних

Основні сутності системи та зв’язки між ними



Суб’єкти доступу

users · roles · user_roles

визначають, хто виконує дію

Об’єкти доступу

resources · departments

задають область і рівень захисту

Контроль подій

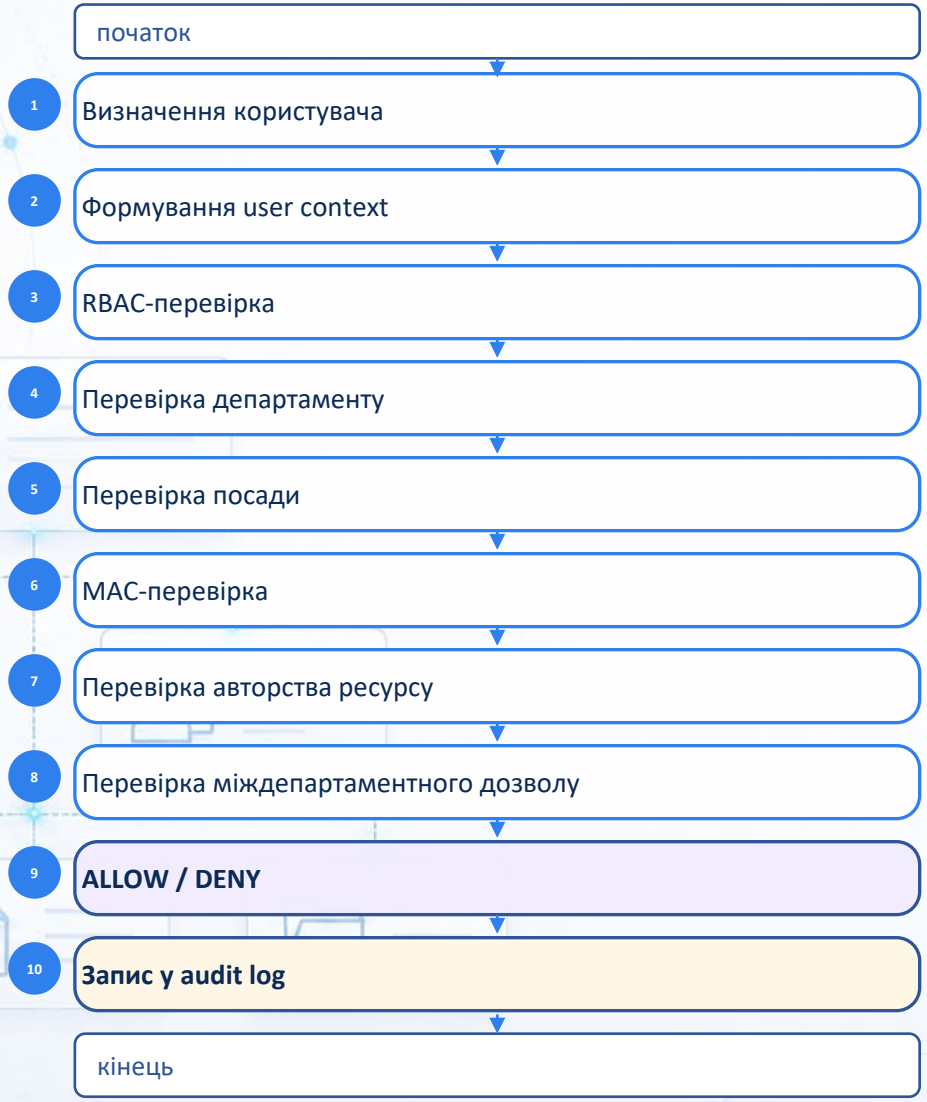
audit_logs · cross_requests

пояснюють рішення і винятки

Структура БД підтримує ролі, департаменти, ресурси, аудит і погоджений міждепартаментний доступ.

Алгоритм перевірки доступу

Кожен запит проходить послідовність обов’язкових перевірок



У системі кожен запит розглядається як комбінація користувача, ресурсу та дії.

Модуль access_policy.py послідовно перевіряє роль, департамент, посаду, MAC-рівень, авторство ресурсу та можливий міждепартаментний дозвіл.

Фінальне рішення формується за принципом deny overrides: якщо хоча б одна обов’язкова умова не виконана, система повертає DENY і записує причину в audit log.

Реалізація RBAC-модуля

Рольові права поєднано з департаментною та посадовою структурою

Департаментна область

однакова роль у різних відділах

не відкриває всю систему

Посадова ієрархія

head > deputy_head > employee

для редагування ресурсів

Системні ролі

main.admin і director

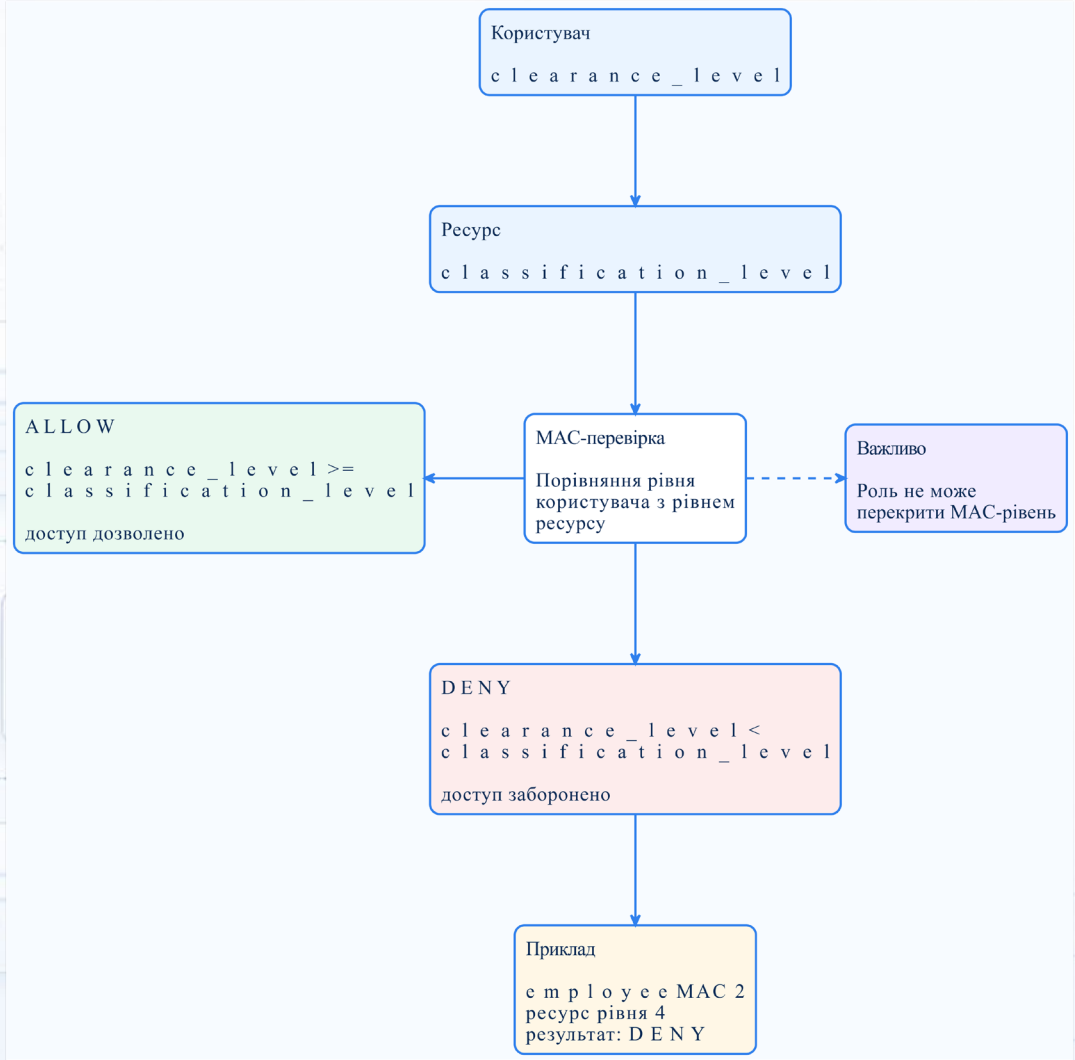
відокремлені від звичайних ролей

Реалізація MAC-модуля

Доступ залежить від співвідношення рівня допуску користувача та ресурсу

Основне правило MAC
clearance_level користувача ≥ classification_level ресурсу

Роль не перекриває допуск
навіть за наявності ролі ресурс вищого рівня блокується

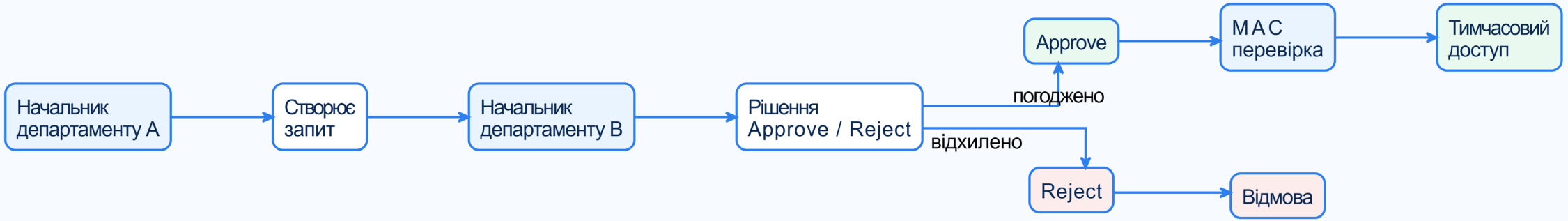


Приклад DENY
employee MAC 2 → ресурс 4
результат: доступ заборонено

Приклад ALLOW
head MAC 5 → ресурс 4
результат: доступ дозволено

Міждепартаментний доступ

Контрольований workflow погодження між відділами

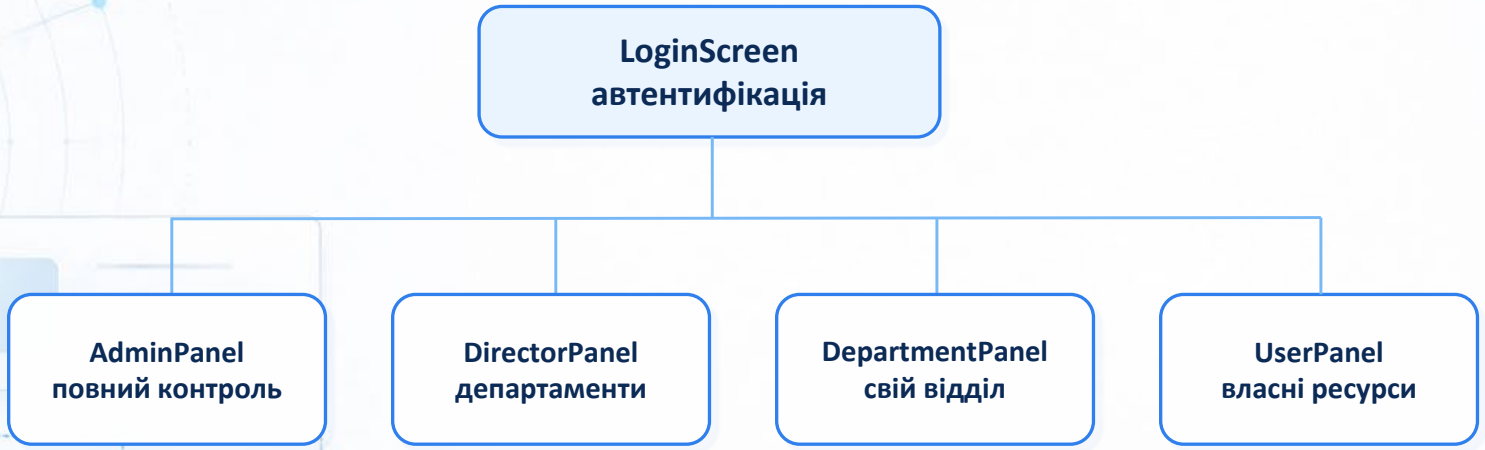


Точковість доступу
дозвіл видається для конкретного
user_id · resource_id · action

MAC залишається обов'язковим
погодження начальника не скасовує
рівень допуску користувача

Audit-ready workflow
approve / reject / revoke
фіксуються в журналі аудиту

Frontend-панелі користувачів



Після автентифікації користувач потрапляє до панелі, яка відповідає його ролі та службовому контексту.

AdminPanel, DirectorPanel, DepartmentPanel і UserPanel відображають різні можливості, однак остаточна авторизація виконується на backend-рівні.

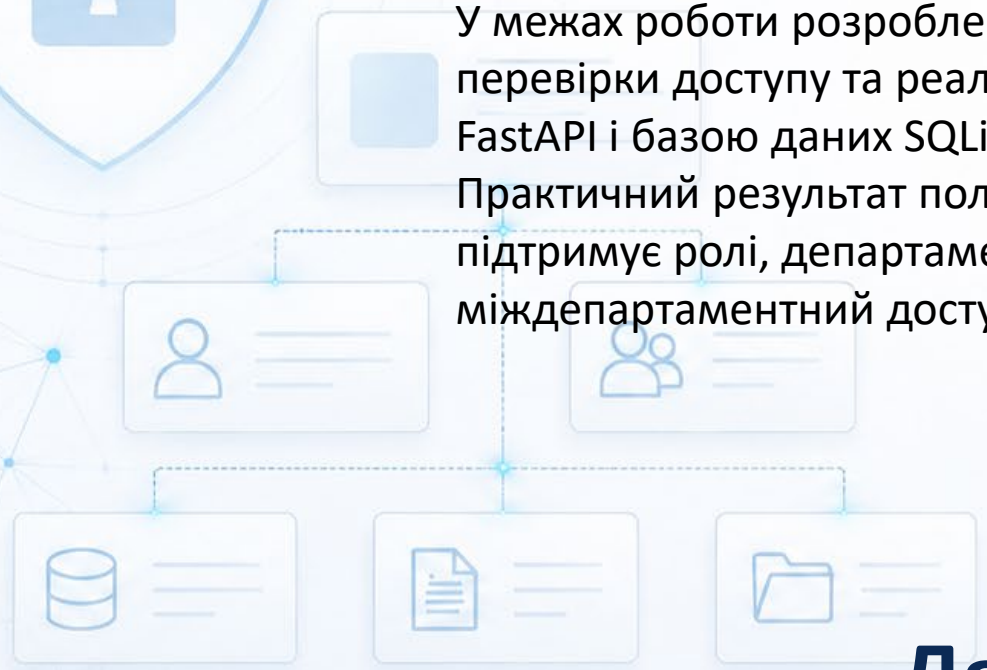
Результати та висновки

Гібридну модель RBAC/MAC реалізовано у вигляді працюючого web-прототипу

Метою роботи є проєктування та програмна реалізація гібридної системи керування доступом, яка поєднує рольову модель RBAC із примусовим контролем MAC.

У межах роботи розроблено архітектуру системи, визначено алгоритм перевірки доступу та реалізовано web-прототип із frontend на React, backend на FastAPI і базою даних SQLite.

Практичний результат полягає у створенні демонстраційного застосунку, що підтримує ролі, департаменти, MAC-рівні, аудит подій та контрольований міждепартаментний доступ.



Дякую за увагу!