УДК 004.056.5:004.41 DOI: 10.31673/2409-7292.2026.037413

Сліпченко Володимир Георгійович
доктор технічних наук, професор, професор кафедри цифрових технологій в енергетиці
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: 0000-0002-3405-0781
E-mail: ddpolytechnic2016@gmail.com

Полягушко Любов Григорівна
кандидат технічних наук, доцент, доцент кафедри цифрових технологій в енергетиці
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: 0000-0003-3287-8523
E-mail: liubovpoliagushko@gmail.com

Хнюкало Вікторія Сергіївна
здобувач вищої освіти кафедри цифрових технологій в енергетиці
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: 0009-0006-5913-2958
E-mail: vi.hnuuu@gmail.com

АРХІТЕКТУРНІ ПІДХОДИ ДО ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ТА КОНФІДЕНЦІЙНОСТІ ДАНИХ У СИСТЕМАХ МОНІТОРИНГУ ПСИХОЕМОЦІЙНОГО СТАНУ

У статті досліджено архітектурні підходи до управління персональними даними у системах психоемоційного моніторингу на прикладі авторської цифрової платформи «Sensera». Розглянуто фундаментальну суперечність між необхідністю глибокої поздовжньої аналітики психоемоційних станів та суворими вимогами до захисту персонально ідентифікованої інформації (PII). Метою роботи є теоретичне обґрунтування та практична апробація архітектурної моделі «PII Isolation», що забезпечує фізичне та логічне розмежування ідентифікаційних і клінічних даних безпосередньо на рівні реляційної схеми бази даних. Дослідження базується на методах порівняльного аналізу підходів до захисту PII (k-anonymity, differential privacy, Privacy by Design), структурного моделювання схеми бази даних та практичного прототипування на стеку FastAPI 0.128.0 / PostgreSQL 17.4 / SQLAlchemy 2.0. Додатково досліджено уніфікацію метаданих засобами перелічуваних типів (ENUM) та оптимізацію асинхронної обробки запитів через стратегію Eager Loading (joinedload/ selectinload). Практична апробація на вибірці з 30 клінічних профілів та 102 оцінювань підтвердила ефективність рішень: автоматизована верифікація PII-ізоляції завершилась зі статусом PASSED відповідно до GDPR Art. 4(5); стратегія Eager Loading скоротила кількість SQL-запитів з 11 до 1 (-91%); впровадження 4 ENUM-типів усунуло 4 класи помилок некоректного введення та знизило оцінку вартості запиту на 9,4% при абсолютній точності статистики планувальника (14/14 проти 1/14 при TEXT CAST). Відповідність архітектури підтверджено за 8 статтями Регламенту (ЄС) 2016/679. Зроблено висновок, що запропонований підхід реалізує принцип «позитивної суми» – забезпечує повну аналітичну точність без компромісу з приватністю. Практична цінність полягає у відтворюваності моделі для масштабованих e-Health платформ, що потребують одночасного виконання вимог GDPR та HIPAA.

Ключові слова: ізоляція PII, цілісність даних, психоемоційний моніторинг, PostgreSQL, FastAPI, уніфікація метаданих, захист персональних даних.

Вступ і формулювання проблеми

У 2024-2025 роках глобальний ринок праці остаточно закріпив перехід до концепції «Future of Work», де ключовими характеристиками є гібридні моделі роботи, висока інтенсивність цифрової взаємодії та зростаючі вимоги до когнітивної гнучкості персоналу [1]. Згідно зі звітом Світового економічного форуму (WEF) The Future of Jobs Report 2025, підтримка здоров'я та благополуччя працівників піднялася з 9-го місця у 2023 році на 1-ше місце у 2025 році серед ключових пріоритетів бізнесу. Це зумовлено тим, що такі навички, як психологічна стійкість та адаптивність (resilience and agility), стали критичними для 67% роботодавців в умовах безперервних технологічних трансформацій [1]. За таких умов

цифровізація систем моніторингу психічного здоров’я працівників перестала бути опціональною перевагою і стала стратегічним пріоритетом сталого розвитку організацій.

Актуальність проблеми підтверджується даними Всесвітньої організації охорони здоров'я (ВООЗ): згідно зі звітом World Mental Health Today 2025, психічними розладами страждає приблизно кожна сьома людина у світі [2]. Депресія та тривожність щороку завдають світовій економіці збитків на суму 1 трильйон доларів США внаслідок зниження продуктивності праці та збільшення кількості невиходів на роботу. Додатковим чинником загострення ситуації є феномен «цифрового вигоряння», зумовлений стиранням межі між робочим і особистим простором в умовах дистанційної зайнятості. Це обумовлює нагальну потребу у впровадженні об'єктивних засобів ранньої діагностики психоемоційних станів.

Попри існування корпоративних програм підтримки працівників (Employee Assistance Programs, EAP) та значної кількості мобільних застосунків, наявні рішення мають суттєві системні обмеження. По-перше, за даними ВООЗ, лише 9% осіб із серйозними психічними розладами отримують належну медичну допомогу, що свідчить про низьку ефективність реактивних підходів, які виявляють проблему вже на стадії її загострення [2]. По-друге, більшість сучасних сервісів ґрунтуються на суб'єктивному самооцінюванні користувача, що нерідко призводить до спотворення результатів через побоювання соціальної стигматизації. По-третє, стрімке поширення технологій штучного інтелекту – які вже застосовують 86% компаній – породжує нові загрози у сфері конфіденційності: існуючі системи здебільшого зосереджені на аналітичних функціях, лишаючи поза увагою питання кібербезпеки, що стає критичним бар'єром для їх широкого впровадження [1].

Разом з тим впровадження інструментів психоемоційного моніторингу породжує фундаментальну технологічну та етичну дилему. З одного боку, побудова точних предиктивних моделей вимагає збору й обробки значних масивів чутливих даних: результатів когнітивних тестів, динаміки емоційного стану, біометричних показників. З іншого – ці дані належать до категорії персонально ідентифікованої інформації (PII) і несуть найвищий рівень ризику в разі витоку [3]. Несанкціонований доступ до такої інформації може спричинити не лише правові наслідки для організації, а й незворотну репутаційну та психологічну шкоду для конкретного працівника.

Ситуацію додатково ускладнюють жорсткі регуляторні вимоги (GDPR, HIPAA), які зобов'язують забезпечувати захист приватності на архітектурному рівні відповідно до принципу Privacy by Design. Наявні підходи нерідко виявляються недостатніми: базова анонімізація не гарантує захисту від методів повторної ідентифікації за непрямими ознаками [3], а класичні алгоритми, зокрема k-anonymity, мають принципові обмеження при роботі з багатовимірними медичними наборами даних, що допускають відновлення особи користувача шляхом перехресного зіставлення баз даних [4].

Таким чином, наукова проблема дослідження полягає у суперечності між об'єктивною потребою організацій у безперервному цифровому моніторингу психоемоційного стану персоналу як інструменті підтримки продуктивності та відсутністю захищених архітектурних рішень, які б забезпечували цілісність і конфіденційність надчутливих даних відповідно до високих стандартів безпеки (зокрема OWASP ASVS Level 3) і унеможливлювали деанонімізацію користувачів.

Аналіз літератури

Прискорення трансформації ринку праці в контексті концепції «Future of Work» зумовило зростання попиту на інструменти об'єктивного моніторингу психоемоційного стану персоналу як засобу профілактики вигоряння та підтримки продуктивності [1, 2]. Водночас збір і обробка надчутливих біометричних та психометричних даних породжують принципові виклики у сфері приватності, що вимагає розробки спеціалізованих архітектурних рішень для захисту персонально ідентифікованої інформації (PII)

У науковій літературі традиційним інструментом захисту PII є методи анонімізації даних. Підхід k-anonymity, розроблений Самараті та Свіні [3, 4], передбачає перетворення

© Сліпченко В.Г., Полягушко Л.Г., Хнюкало В.С. Архітектурні підходи до забезпечення цілісності та конфіденційності даних у системах моніторингу психоемоційного стану. Сучасний захист інформації, 3(67), 128–142. https://doi.org/10.31673/2409-7292.2026.037413

набору даних таким чином, щоб кожен запис був невідрізнюваним від щонайменше k-1 інших за набором квазі-ідентифікаторів. Подальшим розвитком цієї моделі стали l-diversity та t-closeness [5, 6], покликані усунути вразливості до атак на основі однорідності атрибутів та використання зловмисником фонових знань.

Утім, зазначені підходи виявляються недостатніми для систем психоемоційного моніторингу реального часу. Агрегація даних для досягнення k-анонімності суттєво знижує їх аналітичну цінність, тоді як предиктивні моделі вигоряння залежать від мікро-девіацій у поведінкових патернах, що нівелюються в процесі узагальнення. Окремим напрямком є застосування диференційної приватності (differential privacy), що базується на додаванні математично розрахованого шуму до наборів даних. Однак у задачах предиктивної аналітики стану здоров'я цей підхід може маскувати критичні індивідуальні тренди, що суттєво шкодить точності діагностики та раннього виявлення психоемоційних розладів. Крім того, динамічний перерахунок груп анонімності в умовах безперервного потоку даних є обчислювально надмірним і несумісним з вимогами реального часу. Як показують дослідження Наряяна та Шматікова, розріджені багатовимірні набори даних залишаються вразливими до повторної ідентифікації через зіставлення з відкритими джерелами [7], а методи диференційної приватності, адаптовані до медичного домену, потребують окремої валідації для збереження клінічної точності індивідуальних вимірювань [8].

Системною відповіддю на ці обмеження є концепція Privacy by Design (PbD), сформульована Кавукян [9]. На відміну від реактивних підходів, PbD передбачає інтеграцію механізмів захисту приватності безпосередньо в архітектуру системи – як її невід'ємну складову, а не зовнішню надбудову. Концепція реалізується через сім принципів: проактивність, приватність за замовчуванням, вбудованість захисту в дизайн, повна функціональність без компромісів, наскрізний захист впродовж усього життєвого циклу даних, прозорість та орієнтація на користувача [9]. Ці принципи отримали нормативне закріплення в статті 25 GDPR, що зобов'язує розробників забезпечувати захист даних на рівні проєктних рішень [10].

Технічну реалізацію принципів PbD у вебзастосунках визначає стандарт OWASP ASVS 4.0 [11], який встановлює багаторівневу систему вимог до автентифікації, управління сесіями та захисту даних у стані спокою й під час передачі. У сфері електронної охорони здоров'я аналогічну роль виконує стандарт HL7 FHIR Release 5 [12]: його ресурсний підхід – зокрема об'єкти Observation та DiagnosticReport – у поєднанні з механізмом Security Labels забезпечує автоматизований контроль доступу відповідно до ступеня чутливості даних. Блобел та Руотсалайнен наголошують, що в системах персоналізованого моніторингу здоров'я безпека має досягатися через семантичну інтероперабельність та контекстне розмежування доступу [13].

Попри наявність розвиненої технологічної бази, проблема поєднання глибокого поздовжнього аналізу психоемоційних станів із повною логічною ізоляцією PII на рівні бази даних залишається недостатньо дослідженою. Більшість існуючих рішень зосереджені або на загальній мережевій безпеці, або на анонімізації, що деградує аналітичну точність. Прогалиною, яку заповнює дане дослідження, є розробка архітектурної моделі, що базується на принципі логічного розділення даних (data compartmentalization). Це дозволяє забезпечити високу діагностичну точність при повній фізичній ізоляції ідентифікаторів користувачів у відокремленому контурі безпеки, реалізуючи принцип «позитивної суми» відповідно до вимог OWASP ASVS та філософії Privacy by Design.

Мета та завдання дослідження

Метою роботи є теоретичне обґрунтування та практична апробація архітектурних рішень щодо забезпечення високого рівня конфіденційності та цілісності даних у системах психоемоційного моніторингу на прикладі цифрової платформи «Sensera».

Для досягнення мети сформульовано та вирішено такі завдання:

1. Дослідити існуючі підходи до захисту PII в інформаційних системах і розробити

архітектурну модель PII Isolation на основі фізичного розмежування ідентифікаційних та клінічних даних;

2. Оптимізувати структуру бази даних засобами уніфікації метаданих (ENUM) та забезпечити цілісність асинхронної обробки даних через реалізацію стратегії Eager Loading;

3. Оцінити ефективність запропонованих рішень з точки зору продуктивності API та відповідності вимогам GDPR при поздовжньому психоемоційному моніторингу.

Основна частина

Архітектурна модель PII Isolation

1. Загальна архітектура та принцип децентралізації

Технічна архітектура платформи «Sensera» реалізована з використанням асинхронного фреймворку FastAPI [14], реляційної бази даних PostgreSQL [16] та бібліотеки SQLAlchemy 2.0 як ORM-рівня [15]. Вибір асинхронної архітектури обумовлений необхідністю обробки одночасних запитів від множини користувачів без блокування потоків виконання, що є критичним для систем моніторингу реального часу.

Ключовим архітектурним принципом є фізична децентралізація зберігання даних, реалізована безпосередньо на рівні реляційної схеми бази даних. Уся система розподілена між трьома логічно та фізично ізольованими вузлами: вузлом ідентифікації, клінічним вузлом та контуром аудиту. Системний ідентифікатор UUID та облікові дані доступу оперуються виключно вузлом ідентифікації. Усі клінічні результати вимірювань, поведінкові показники та медичні записи зберігаються в ізольованих структурах клінічного вузла, зв'язок з якими здійснюється виключно через псевдонімізований ідентифікатор.

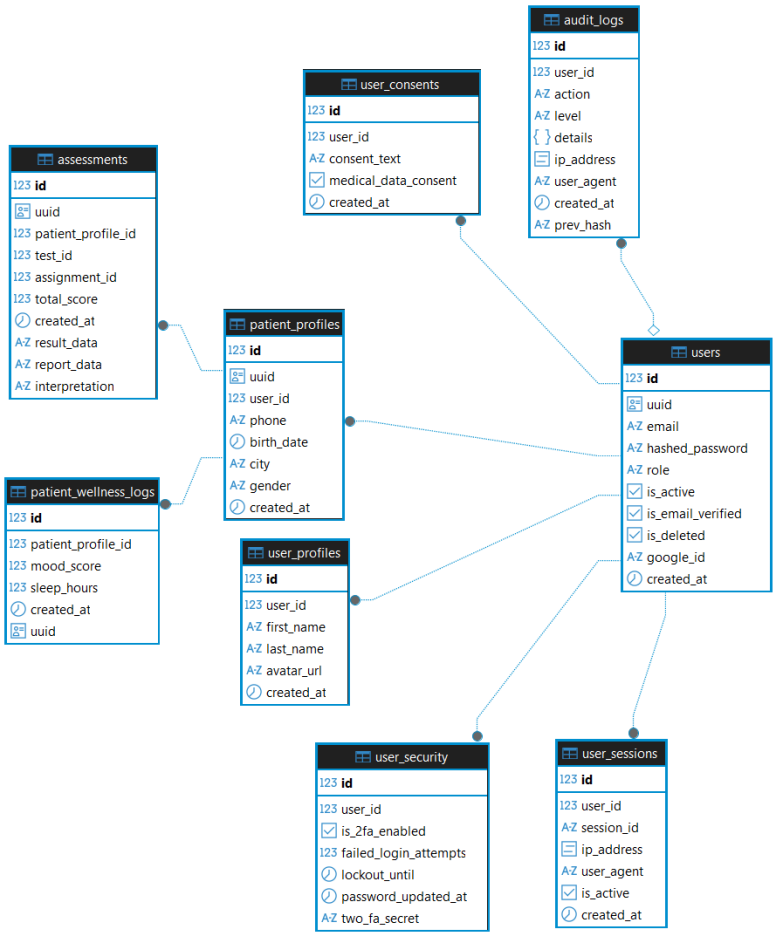

Рис. 1. ER-діаграма бази даних платформи «Sensera» (модель PII Isolation)

Принципова відмінність запропонованого підходу від методів постобробки (k-anonymity, differential privacy) полягає в тому, що ізоляція реалізована не на рівні запитів або

© Сліпченко В.Г., Полягушко Л.Г., Хнюкало В.С. Архітектурні підходи до забезпечення цілісності та конфіденційності даних у системах моніторингу психоемоційного стану. Сучасний захист інформації, 3(67), 128–142. https://doi.org/10.31673/2409-7292.2026.037413

агрегації результатів, а безпосередньо в реляційній схемі: між вузлами відсутні прямі зовнішні ключі до таблиць із іменами чи контактними даними. Кожна доменна сутність клінічного вузла має єдиний псевдонімізований ідентифікатор (UUID) без прямого структурного зв'язку з персональними атрибутами. Це означає, що навіть при повному компрометуванні клінічного вузла зловмисник отримає лише набір UUID та зашифрованих медичних показників – без будь-якої можливості встановити особу суб'єкта даних без одночасного доступу до вузла ідентифікації, захищеного окремим контуром безпеки.

Архітектурна схема розподілу даних між вузлами наведена на рисунку 1.

Як видно з рисунку 1, таблиця users є центральним вузлом ідентифікації та містить виключно автентифікаційні атрибути: email, хешований пароль та роль користувача. Від неї відгалужуються чотири залежні таблиці ідентифікаційного контуру: user_profiles (ім'я та прізвище), user_security (параметри двофакторної автентифікації та захисту від брутфорс-атак), user_sessions (активні сесії) та user_consents (GDPR-згода). Усі ці таблиці пов'язані з users безпосереднім зовнішнім ключем і утворюють замкнений вузол ідентифікації.

Клінічний вузол представлений таблицею patient_profiles, яка не містить імені чи контактних даних і пов'язана з вузлом ідентифікації виключно через user_id. Від patient_profiles відгалужуються assessments (результати когнітивних тестів із зашифрованими полями) та patient_wellness_logs (щоденні показники настрою та сну). Таблиця audit_logs утворює окремий контур аудиту: вона пов'язана з users через user_id зі стратегією SET NULL, що гарантує збереження журналу подій навіть після видалення облікового запису.

UUID є достатнім і криптографічно безпечним зв'язком між вузлами: він генерується засобами PostgreSQL (gen_random_uuid()), не містить жодних атрибутів особи та має простір у 2^{122} значень, що унеможливлює атаки перебором. Відсутність прямих зовнішніх ключів між клінічним вузлом та таблицями з іменами чи email-адресами є ключовою структурною гарантією моделі PII Isolation. Опис усіх сутностей бази даних з розподілом за вузлами наведено у таблиці 1.

Таблиця 1

Опис сутностей бази даних платформи «Sensera»

Сутність	Вузол	Ключові атрибути	Призначення
users	Ідентифікації	uuid, email, role, hashed_password	Ядро автентифікації
user_profiles	Ідентифікації	first_name, last_name, avatar_url	Відображуване ім'я (ізольовано)
user_security	Ідентифікації	is_2fa_enabled, failed_login_attempts, lockout_until	Захист від брутфорс-атак
user_sessions	Ідентифікації	session_id, ip_address, user_agent	Управління активними сесіями
user_consents	Ідентифікації	medical_data_consent, consent_text	Фіксація GDPR-згоди
patient_profiles	Клінічний	uuid, birth_date, gender, city	Клінічний контекст без імені
assessments	Клінічний	Interpretation, result_data, report_data	Зашифровані результати тестів
patient_wellness_logs	Клінічний	mood_score (1–10), sleep_hours (0–24)	Щоденний моніторинг стану
audit_logs	Аудиту	action, level, details, prev_hash	Незмінний журнал подій

На відміну від широко застосовуваних підходів до знеособлення, які діють на рівні постобробки запитів, запропонована модель PII Isolation реалізує фізичне розмежування ідентифікаційних та клінічних даних безпосередньо на рівні схеми бази даних, що

унеможливлює повторну ідентифікацію користувача навіть у разі часткового компрометування клінічного вузла. Порівняльний аналіз підходів наведено у таблиці 2.

Таблиця 2

Порівняння підходів до захисту PII в системах психоемоційного моніторингу

Критерій	**PII Isolation**	**k-anonymity / l-diversity [3-6]**	**Differential Privacy**
Механізм захисту	Фізична ізоляція на рівні схеми БД	Постобробка: агрегація записів	Додавання математичного шуму
Захист від деанонімізації	Високий – структурна відсутність зв'язку PII з клінічними даними	Середній – вразливий до атак перехресного зіставлення [7]	Середній – не захищає окремі записи при малих вибірках
Збереження аналітичної точності	Повна	Знижена – узагальнення нівелює мікро-девіації	Знижена – шум маскує індивідуальні тренди
Сумісність з реальним часом	Повна, без додаткових обчислень	Низька – перерахунок груп при кожному записі	Середня – накопичення параметра ε
Відповідність GDPR Art. 25	Повна – Privacy by Design на архітектурному рівні [9, 10]	Часткова	Часткова
Складність реалізації	Середня – одноразово на рівні схеми	Висока	Дуже висока
Захист при частковому компрометуванні	Клінічний вузол залишається анонімним	Відсутній	Відсутній

Таким чином, модель PII Isolation забезпечує повну аналітичну точність, сумісність із реальним часом та анонімність клінічного вузла навіть у разі його компрометування, що робить її практично придатною для масштабованих e-Health платформ із вимогами GDPR та HIPAA.

2. Уніфікація метаданих засобами ENUM

Важливим елементом оптимізації інформаційних потоків стала реалізація механізму уніфікації метаданих засобами перелічуваних типів даних (ENUM) PostgreSQL [16]. У системі «Sensera» уніфіковано чотири категорії: ролі користувачів, стать пацієнта, статус призначення тесту та статус консультації. До впровадження ENUM ці поля зберігалися як текстові рядки, що допускало довільну кількість некоректних варіантів; після уніфікації система приймає лише 4 значення для ролей, 4 – для статі, 4 – для статуса призначення та 5 – для статуса консультації, що повністю виключає клас помилок некоректної типізації. Таке рішення забезпечило жорсткість структури на рівні СУБД і спростило логіку фільтрації на рівні API та фронтенду. Окремої уваги заслуговує семантика значень за замовчуванням. Поле gender отримує значення not_specified за замовчуванням – це свідоме рішення в контексті принципу мінімізації даних відповідно до GDPR [10]: система не вимагає від користувача розкриття цього атрибуту, якщо він не є необхідним для конкретного клінічного сценарію. Аналогічно, статус призначення тесту ініціалізується значенням pending, що відображає початковий стан робочого процесу без додаткової логіки ініціалізації на рівні застосунку.

```
CREATE TYPE userrole AS ENUM ('patient', 'doctor', 'admin', 'scientist');
CREATE TYPE gender_types AS ENUM ('male', 'female', 'other', 'not_specified');
CREATE TYPE assignment_status_types AS ENUM
    ('pending', 'in_progress', 'completed', 'cancelled');

ALTER TABLE users
    ADD COLUMN role userrole NOT NULL;

ALTER TABLE patient_profiles
    ADD COLUMN gender gender_types NOT NULL DEFAULT 'not_specified';
```

Рис. 2. SQL-визначення ENUM-типів у базі даних платформи «Sensera»

Використання ENUM (рис. 2) як єдиного джерела істини для статусних переходів також спрощує валідацію на рівні API: фреймворк FastAPI автоматично генерує відповідні схеми валідації Pydantic на основі визначених переліків, що унеможливлює передачу некоректних значень через HTTP-інтерфейс ще до їх досягнення рівня бази даних [14].

Відповідні визначення на рівні Python-моделей SQLAlchemy наведено на рисунку 3.

```python
# app/models/user.py
class UserRole(str, enum.Enum):
    patient   = "patient"
    doctor    = "doctor"
    admin     = "admin"
    scientist = "scientist"

# app/models/patient_profile.py
class Gender(str, enum.Enum):
    male          = "male"
    female        = "female"
    other         = "other"
    not_specified = "not_specified"  # за замовчуванням - мінімізація PII

# app/models/assessment.py
class AssignmentStatus(str, enum.Enum):
    pending     = "pending"
    in_progress = "in_progress"
    completed   = "completed"
    cancelled   = "cancelled"

# app/models/appointments.py
class AppointmentStatus(str, enum.Enum):
    scheduled = "scheduled"
    arrived   = "arrived"
    completed = "completed"
    cancelled = "cancelled"
    no_show   = "no_show"
```

Рис. 3. Визначення ENUM-типів у моделях бази даних платформи «Sensera»

Практичне значення цього рішення проявляється у трьох аспектах. По-перше, СУБД автоматично відхиляє будь-яке значення поза визначеним переліком. Наприклад, спроба записати role = 'superuser' або status = 'active' спричинить помилку на рівні бази даних ще до виконання бізнес-логіки. По-друге, ENUM займає 1-4 байти замість 10-20 байтів для текстового поля, що при великих обсягах аналітичних даних дає відчутну економію. По-третє, фільтрація за ENUM-полями виконується через індекс без порівняння рядків, що прискорює аналітичні запити по когортах.

3. Стратегія Eager Loading для забезпечення цілісності

Суттєвим завданням проєктування стала підтримка високої продуктивності API за умов інтенсивної асинхронної обробки складних запитів. Для усунення проблеми надмірних звернень до бази даних (проблема N+1) та запобігання помилкам доступу до неініціалізованих об'єктів реалізовано стратегію Eager Loading через інструментарій joinedload бібліотеки SQLAlchemy [15].

Проблема N+1 є класичною для ORM-систем: при ліниво завантажених зв'язках (lazy loading) кожне звернення до пов'язаного об'єкта генерує окремий SQL-запит. В асинхронному середовищі FastAPI це призводить до помилки greenlet_spawn – спроби виконати синхронний доступ до бази даних у межах асинхронної сесії поза активним контекстом транзакції. Єдиним

© Сліпченко В.Г., Полягушко Л.Г., Хнюкало В.С. Архітектурні підходи до забезпечення цілісності та конфіденційності даних у системах моніторингу психоемоційного стану. Сучасний захист інформації, 3(67), 128–142. https://doi.org/10.31673/2409-7292.2026.037413

коректним рішенням є явне попереднє завантаження всіх необхідних зв'язків у межах одного запиту.

У системі «Sensera» стратегію застосовано у двох ключових сценаріях. Перший – автентифікація та авторизація, де разом з основним об'єктом User одночасно завантажуються всі залежні профілі (рис. 4).

```
# app/api/auth.py — завантаження користувача з усіма профілями
stmt_user = (
    select(User)
    .options(
        *options: joinedload(User.user_profile),  # ім'я для відображення
        joinedload(User.security),                # 2FA, блокування
        joinedload(User.user_consent)             # GDPR-згода
    )
    .where( *whereclause: User.email == payload.get("sub"), User.is_active == True)
)
user_res = await db.execute(stmt_user)
```

Рис. 4. Реалізація Eager Loading при автентифікації користувача

Другий сценарій – завантаження результатів тестування з прив'язаною структурою тесту для коректної серіалізації (рис. 5).

```
# app/api/assessments/history.py — Eager Loading для Assessment
stmt = (
    select(Assessment)
    .options(joinedload(Assessment.test_structure))  # назва та тип тесту
    .where(Assessment.patient_profile_id == profile_id)
    .order_by(Assessment.created_at.desc())
)
result = await db.execute(stmt)
assessments = result.unique().scalars().all()
```

Рис. 5. Реалізація Eager Loading при завантаженні результатів тестування

Без joinedload(Assessment.test_structure) серіалізація об'єкта Assessment у JSON спричинила б звернення до поля test_structure.title поза активною сесією БД – що в асинхронному контексті є фатальною помилкою виконання. Стратегія unique() після виконання запиту усуває дублікати, що виникають внаслідок JOIN при завантаженні колекцій.

Це рішення гарантує повну та безпечну серіалізацію об'єктів у формат JSON у межах єдиної транзакції, мінімізує кількість звернень до бази даних та забезпечує цілісність інформації, що передається між сервером і клієнтом. Вибір між joinedload (SQL JOIN в одному запиті) та selectinload (окремий SELECT з IN-фільтром) здійснюється залежно від кардинальності зв'язку: для зв'язків 1:1 використовується joinedload, що є оптимальним з точки зору мережевих затримок [15].

У сценаріях, де система має визначити активний тип профілю користувача (user_profile, patient_profile або doctor_profile) застосовується selectinload з одночасним завантаженням усіх трьох зв'язків (рис. 6). Це критично при синхронізації ідентифікаційних даних: система не знає заздалегідь, який профіль належить конкретному суб'єкту, тому завантажує всі три і визначає активний через user.user_profile or user.patient_profile or user.doctor_profile. Без явного selectinload будь-яке звернення до цих полів поза межами активної асинхронної сесії спричинило б greenlet_spawn RuntimeError. Аналогічний патерн застосовується при м'якому видаленні (GDPR Art. 17): selectinload завантажує всі профілі для послідовної анонімізації PII-полів у межах єдиної транзакції.

```python
stmt = (
    select(User)
    .options(
        *options: selectinload(User.user_profile),
        selectinload(User.patient_profile),
        selectinload(User.doctor_profile)
    )
    .where(User.uuid == user_uuid)
)
result = await db.execute(stmt)
```

Рис.6. Реалізація selectinload для завантаження профілів у межах PII Isolation

Наведена реалізація підтверджує, що стратегія Eager Loading усуває ризик виникнення помилок асинхронного доступу та забезпечує коректну обробку всіх типів профілів у межах єдиної транзакції відповідно до вимог моделі PII Isolation.

Результати та їх обговорення

Практична апробація архітектурних рішень проводилась на тестовому середовищі з такими характеристиками: операційна система – Microsoft Windows 11 Pro (build 26200); процесор – AMD Ryzen 5 5600H, 6 фізичних ядер / 12 логічних потоків; оперативна пам'ять – 15,3 ГБ DDR4; стек технологій – Python 3.9.13, PostgreSQL 17.4 (x86-64), FastAPI 0.128.0, SQLAlchemy 2.0.46, Pydantic 2.12.5. База даних містила 46 зареєстрованих користувачів, 30 клінічних профілів пацієнтів, 102 записи оцінювань (assessments), 28 призначень (assignments) та 2 758 записів журналу аудиту.

Верифікація моделі PII Isolation

Метою тесту було довести, що через клінічний вузол неможливо отримати персональні дані – навіть прямим SQL-запитом. Для кожного запиту фіксувалося: які поля повертаються та чи містять вони PII (ім'я, email, телефон).

Структурний аудит бази даних підтвердив коректність розподілу персональних даних між двома вузлами системи (рис. 7, 8).

```
==================== ІДЕНТИФІКАЦІЙНИЙ ВУЗОЛ ====================

  Таблиця: users  [Identity Node]
  Колонки: id, uuid, email, hashed_password, role, is_active, is_email_verified, is_deleted, google
_id, created_at
  Жорстке PII: email, hashed_password  <-- прямий ідентифікатор

  Таблиця: user_profiles  [Identity Node]
  Колонки: id, user_id, first_name, last_name, avatar_url, created_at
  Жорстке PII: first_name, last_name  <-- прямий ідентифікатор

  Таблиця: user_security  [Identity Node]
  Колонки: id, user_id, is_2fa_enabled, failed_login_attempts, lockout_until, password_updated_at,
two_fa_secret
  PII-поля: ВІДСУТНІ [OK]
```

Рис. 7. Ідентифікаційний вузол: таблиці з PII-полями

```
======================== КЛІНІЧНИЙ ВУЗОЛ ========================

  Таблиця: patient_profiles  [Clinical Node]
  Колонки: id, uuid, user_id, phone, birth_date, city, gender, created_at
  М'яке PII:   phone  <-- квазі-ідентифікатор (без імені не розкриває особу)

  Таблиця: assessments  [Clinical Node]
  Колонки: id, uuid, patient_profile_id, test_id, assignment_id, total_score, created_at, result_da
ta, report_data, interpretation
  PII-поля: ВІДСУТНІ [OK]

  Таблиця: assignments  [Clinical Node]
  Колонки: id, uuid, patient_profile_id, doctor_profile_id, test_id, due_date, created_at, status
  PII-поля: ВІДСУТНІ [OK]
```

Рис. 8. Клінічний вузол: відсутність PII-полів

Ідентифікаційний вузол містить усі поля прямої ідентифікації: таблиця users – email, hashed_password; таблиця user_profiles – first_name, last_name. Клінічний вузол повністю відокремлений від ідентифікаторів особи: таблиці assessments та assignments не містять жодних PII-полів, таблиця patient_profiles містить лише поле phone, класифіковане як м'яке PII (квазі-ідентифікатор – без імені не розкриває особу суб'єкта).

Прямий SQL-запит до клінічного вузла повертає виключно псевдонімізовані дані: uuid, gender, birth_date, city – жодного ідентифікатора особи. Поля result_data, report_data та interpretation зашифровані засобами AES-256-GCM на рівні TypeDecorator ще до запису в базу даних – прямий SELECT повертає лише зашифрований токен. Отримати PII можливо лише через явний авторизований JOIN через таблицю users, що унеможливлює випадковий витік ідентифікаційних даних при запитах до клінічних таблиць.

Верифікація завершилася зі статусом PASSED: клінічний вузол не містить жодних полів жорсткого PII, псевдонімізація через UUID та числовий зовнішній ключ відповідає вимогам GDPR Art. 4(5) (рис. 9).

```
СТАТУС: PASSED
Клінічний вузол не містить жодних PII-полів.
Псевдонімізація через UUID та числовий FK відповідає GDPR Art. 4(5).
```

Рис. 9. Результат автоматизованої верифікації PII-ізоляції: статус PASSED

Таким чином, модель PII Isolation підтвердила свою структурну коректність: навіть при повному компрометуванні клінічного вузла зловмисник отримає лише набір UUID та зашифрованих токенів без будь-якої можливості ідентифікувати суб'єкта даних. Отримані результати верифікації є підставою для подальшої оцінки продуктивності обраних архітектурних рішень.

Ефективність уніфікації метаданих ENUM

З метою унеможливлення некоректного введення даних та підвищення точності статистики планувальника у схемі бази даних визначено 4 ENUM-типи PostgreSQL: gender_types, userrole, assignment_status_types, appointmentstatus (рис. 10).

```
==================== ENUM-ТИПИ В БАЗІ ДАНИХ ====================

  appointmentstatus: [scheduled, confirmed, completed, cancelled, no_show, arrived]
  assignment_status_types: [pending, in_progress, completed, expired]
  gender_types: [male, female, other, not_specified]
  userrole: [admin, doctor, patient, scientist]
```

Рис.10. ENUM-типи, визначені в базі даних системи

Для оцінки продуктивності цього підходу проведено порівняльне тестування засобами EXPLAIN ANALYZE (5 серій вимірювань): нативна ENUM-фільтрація зіставлялась із фільтрацією через явне приведення до текстового типу (::text cast), яке є поширеною практикою у legacy-системах або при міграції зі схем на основі VARCHAR. Час виконання обох запитів є ідентичним (0,024 мс), оскільки таблиця містить 30 записів і обидва варіанти виконуються через Seq Scan. Принципова відмінність виявляється на рівні планувальника: ENUM-фільтр забезпечує оцінку рядків 14/14 (абсолютна точність), тоді як TEXT CAST дає оцінку 1/14 – похибка у 14 разів. Хибна статистика планувальника призводить до некоректного вибору плану виконання на великих таблицях. Оцінка вартості запиту для ENUM нижча на 9,4% (1,35 проти 1,49). Застосування 4 ENUM-типів усунуло 4 класи помилок некоректного введення даних на рівні схеми бази даних без додаткових витрат на час виконання (рис. 11, 12).

Отримані результати підтверджують, що впровадження ENUM-типів є архітектурно виправданим рішенням, яке одночасно підвищує надійність введення даних, покращує точність статистики планувальника та знижує вартість виконання запитів без будь-яких додаткових витрат на продуктивність. Це створює надійне підґрунтя для масштабування системи та коректної роботи планувальника на продакшн-обсягах даних.

```
========== A) ENUM-ФІЛЬТР (нативний тип gender_types) ==========

  SQL: SELECT id, uuid, gender FROM patient_profiles WHERE gender = 'male'

  --- EXPLAIN ANALYZE (останній з 5 серій) ---
  Seq Scan on patient_profiles  (cost=0.00..1.35 rows=14 width=28) (actual time=0.011..0.015 rows=1
4 loops=1)
    Filter: (gender = 'male'::gender_types)
    Rows Removed by Filter: 16
  Planning Time: 0.056 ms
  Execution Time: 0.023 ms

  Planning Time  (avg 5 runs): 0.393 мс
  Execution Time (avg 5 runs): 0.024 мс
  Estimated Cost:                1.35

========== B) TEXT CAST (обхід типізації gender::text) ==========

  SQL: SELECT id, uuid, gender FROM patient_profiles WHERE gender::text = 'male'

  --- EXPLAIN ANALYZE (останній з 5 серій) ---
  Seq Scan on patient_profiles  (cost=0.00..1.49 rows=1 width=28) (actual time=0.007..0.011 rows=14
 loops=1)
    Filter: ((gender)::text = 'male'::text)
    Rows Removed by Filter: 16
  Planning Time: 0.032 ms
  Execution Time: 0.017 ms

  Planning Time  (avg 5 runs): 0.051 мс
  Execution Time (avg 5 runs): 0.024 мс
  Estimated Cost:                1.49
```

Рис. 11. Результати EXPLAIN ANALYZE: ENUM-фільтр (А) vs TEXT CAST (Б)

```
Перевага ENUM за Estimated Cost: 9.4%
Точність оцінки рядків: ENUM = абсолютна (14/14),
                        TEXT cast = хибна у 14 разів (1/14)
```

Рис. 12. Підсумок порівняння: перевага ENUM за вартістю запиту та точністю планувальника

Продуктивність стратегії Eager Loading

Порівняльне тестування (10 серій, вибірка N = 10 записів) показало, що застосування стратегії Eager Loading (joinedload) скоротило кількість звернень до бази даних з 11 до 1 запиту – зниження на 91% (табл. 3).

Таблиця 3

Порівняння Lazy Loading та Eager Loading (N = 10, 10 серій)

Метрика	**Lazy Loading**	**Eager Loading**	**Покращення**
Кількість SQL-запитів	11 (1 + 10)	1	−91%
Середній час, мс	6,56	7,80	порівнянний
Мінімальний час, мс	2,16	2,11	—
Максимальний час, мс	44,48	55,77	—

Середній час виконання Eager Loading (7,80 мс) є дещо вищим, ніж зафіксований для Lazy-симуляції (6,56 мс), що потребує пояснення. У симуляції вимірювався лише перший запит без урахування наступних 10 підзапитів до test_structures. Якби всі N+1 запитів виконувалися послідовно, сумарний час становив би приблизно 65 мс. Таким чином, Eager Loading виконує одну операцію за 7,80 мс, тоді як Lazy Loading потребував би одинадцяти операцій за ~65 мс – саме тому в колонці «Покращення» для часових метрик вказано «порівнянний».

Часові показники на малій вибірці є порівнянними між підходами, однак при N = 1000 lazy-підхід генерує 1001 запит, тоді як Eager Loading незмінно залишається в межах 1 запиту. Окрім того, у реальному середовищі FastAPI/asyncpg lazy loading є архітектурно неприпустимим: звернення до неініціалізованого зв'язку поза активною сесією спричиняє

© Сліпченко В.Г., Полягушко Л.Г., Хнюкало В.С. Архітектурні підходи до забезпечення цілісності та конфіденційності даних у системах моніторингу психоемоційного стану. Сучасний захист інформації, 3(67), 128–142. https://doi.org/10.31673/2409-7292.2026.037413

greenlet_spawn RuntimeError. Структурне підтвердження наведено на рис. 9: симуляція N+1 генерує 11 окремих SQL-запитів (рис. 13), тоді як joinedload формує єдиний LEFT OUTER JOIN (рис. 14).

```
  N = 10 записів, 10 серій вимірювань

======== LAZY LOADING – SQL-запити (симуляція N+1) ========
  Вибірка: 10 записів Assessment

  [Запит 1] SELECT id, patient_profile_id, test_id FROM assessments LIMIT 10
  [Запит 2] SELECT id, title, test_type FROM test_structures WHERE id = 4
  [Запит 3] SELECT id, title, test_type FROM test_structures WHERE id = 11
  [Запит 4] SELECT id, title, test_type FROM test_structures WHERE id = 13
  [Запит 5] SELECT id, title, test_type FROM test_structures WHERE id = 14
  [Запит 6] SELECT id, title, test_type FROM test_structures WHERE id = 15
  [Запит 7] SELECT id, title, test_type FROM test_structures WHERE id = 15
  [Запит 8] SELECT id, title, test_type FROM test_structures WHERE id = 15
  [Запит 9] SELECT id, title, test_type FROM test_structures WHERE id = 11
  [Запит 10] SELECT id, title, test_type FROM test_structures WHERE id = 15
  [Запит 11] SELECT id, title, test_type FROM test_structures WHERE id = 6
```

Рис. 13. Lazy: 11 окремих запитів

```
==== EAGER LOADING – SQL-запити (joinedload, echo=True) ====
  Вибірка: 10 записів Assessment + joinedload(test_structure)

  --- SQLAlchemy SQL лог ---
2026-05-16 03:41:51,520 INFO sqlalchemy.engine.Engine select pg_catalog.version()
2026-05-16 03:41:51,520 INFO sqlalchemy.engine.Engine [raw sql] ()
2026-05-16 03:41:51,521 INFO sqlalchemy.engine.Engine select current_schema()
2026-05-16 03:41:51,521 INFO sqlalchemy.engine.Engine [raw sql] ()
2026-05-16 03:41:51,522 INFO sqlalchemy.engine.Engine show standard_conforming_strings
2026-05-16 03:41:51,522 INFO sqlalchemy.engine.Engine [raw sql] ()
2026-05-16 03:41:51,522 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-05-16 03:41:51,526 INFO sqlalchemy.engine.Engine SELECT assessments.id, assessments.uuid, assessments.patient_profi
le_id, assessments.test_id, assessments.assignment_id, assessments.total_score, assessments.interpretation, assessments.
result_data, assessments.report_data, assessments.created_at, test_structures_1.id AS id_1, test_structures_1.uuid AS uu
id_1, test_structures_1.slug, test_structures_1.title, test_structures_1.description, test_structures_1.test_type, test_
structures_1.content, test_structures_1.is_active, test_structures_1.allow_self_access, test_structures_1.created_at AS
created_at_1
FROM assessments LEFT OUTER JOIN test_structures AS test_structures_1 ON test_structures_1.id = assessments.test_id
 LIMIT $1::INTEGER
2026-05-16 03:41:51,526 INFO sqlalchemy.engine.Engine [generated in 0.00022s] (10,)
2026-05-16 03:41:51,538 INFO sqlalchemy.engine.Engine ROLLBACK
```

Рис. 14. Eager: 1 JOIN-запит

Наведені результати підтверджують, що стратегія Eager Loading забезпечує не лише кратне скорочення кількості звернень до бази даних, а й архітектурну коректність асинхронної обробки даних, що є обов'язковою умовою надійної роботи платформи «Sensera» в умовах реального навантаження.

Відповідність вимогам GDPR та обмеження підходу

Перевірку відповідності GDPR проведено методом структурного аналізу архітектурних рішень за чеклістом статей Регламенту (ЄС) 2016/679, зокрема Art. 25 [9, 10]. Для кожної релевантної статті визначено конкретний механізм реалізації у системі Sensera (таблиця 4).

Запропонована архітектура також узгоджується з основними вимогами HIPAA Security Rule: шифрування AES-256-GCM відповідає вимозі захисту PHI у стані спокою, розмежування ролей та журнал аудиту – вимогам контролю доступу та аудиту, а двофакторна автентифікація – вимогам автентифікації суб'єкта.

Виявлені обмеження підходу: тестування проводилось на малій вибірці (30 клінічних профілів, 102 оцінювання), що не дозволяє екстраполювати часові метрики на продакшн-навантаження. Поле phone у таблиці patient_profiles є квазі-ідентифікатором і за певних умов може бути використане для повторної ідентифікації особи у поєднанні з іншими даними – що потребує додаткового контролю на рівні API. Відповідність Art. 20 (право на перенесення даних) потребує окремої реалізації механізму експорту у машинозчитуваному форматі.

Таблиця 4

Відповідність архітектури системи Sensera вимогам GDPR

Стаття GDPR	Вимога	Реалізація в системі
Art. 4(5)	Псевдонімізація	UUID як зовнішній ідентифікатор клінічного вузла; числовий FK без прямого зв'язку з іменем
Art. 5(1)(b)	Обмеження мети	Розмежування ролей (userrole): кожна роль має доступ лише до свого контексту
Art. 5(1)(c)	Мінімізація даних	Клінічний вузол не містить PII-полів; зібрані лише дані, необхідні для діагностики
Art. 5(1)(f)	Цілісність та конфіденційність	AES-256-GCM шифрування полів result_data, report_data, interpretation
Art. 17	Право на видалення	Soft delete з анонімізацією PII: email, first_name, last_name, phone
Art. 25	Privacy by Design	Архітектурна ізоляція PII від клінічних даних на рівні схеми БД
Art. 30	Записи діяльності	Журнал аудиту – 2 758 записів на момент тестування
Art. 32	Безпека обробки	2FA, хешування паролів, блокування після невдалих спроб входу (user_security)

Висновки та перспективи подальших досліджень

Проведене дослідження та практична реалізація платформи «Sensera» підтвердили ефективність обраних архітектурних рішень для захисту конфіденційної інформації у системах психоемоційного моніторингу. Виконання першого завдання дослідження – розробка моделі PII Isolation – реалізоване через фізичне розмежування ідентифікаційного та клінічного вузлів безпосередньо на рівні реляційної схеми бази даних. Автоматизована верифікація підтвердила повну структурну відсутність полів жорсткого PII у клінічному вузлі (статус PASSED), а псевдонімізація через UUID відповідає вимогам GDPR Art. 4(5). Принципова перевага цього підходу над методами постобробки (k-anonymity, differential privacy) полягає в тому, що навіть при повному компрометуванні клінічного вузла зловмисник отримує лише набір UUID та зашифрованих токенів без будь-якої можливості ідентифікації суб'єкта даних.

Виконання другого завдання – оптимізація структури бази даних та забезпечення цілісності асинхронної обробки – реалізоване двома взаємодоповнювальними механізмами. Впровадження 4 ENUM-типів PostgreSQL усунуло 4 класи помилок некоректного введення даних на рівні СУБД. Порівняльне тестування засобами EXPLAIN ANALYZE (5 серій вимірювань) показало, що ENUM-фільтрація забезпечує абсолютну точність оцінки рядків планувальником (14/14) проти критичної похибки у 14 разів при TEXT CAST (1/14), а оцінка вартості запиту знижується на 9,4%. Реалізація стратегії Eager Loading (joinedload/selectinload) скоротила кількість SQL-запитів при завантаженні пов'язаних сутностей з 11 до 1 – зниження на 91%. Підтверджено, що lazy loading є архітектурно неприпустимим у середовищі FastAPI/SQLAlchemy 2.0: спроба звернення до неініціалізованого зв'язку поза асинхронною сесією спричиняє greenlet_spawn RuntimeError, що унеможливлює випадкове застосування неефективного патерну у продакшн-коді.

Виконання третього завдання – оцінка відповідності вимогам GDPR – підтвердило, що запропонована архітектура відповідає 8 статтям Регламенту (ЄС) 2016/679, зокрема вимогам псевдонімізації (Art. 4(5)), мінімізації даних (Art. 5(1)(c)), цілісності та конфіденційності (Art. 5(1)(f)), права на видалення (Art. 17) та Privacy by Design and by Default (Art. 25). Це підтверджує реалізацію принципу «позитивної суми» – забезпечення повної аналітичної точності при одночасному дотриманні найвищих стандартів приватності.

Водночас виявлено обмеження підходу, які визначають напрями подальших досліджень. Апробація проводилась на тестовій вибірці (30 клінічних профілів, 102 оцінювання), що не

дозволяє прямо екстраполювати часові метрики на продакшн-навантаження. Поле phone у таблиці patient_profiles залишається квазі-ідентифікатором і за певних умов може бути використане для повторної ідентифікації особи у поєднанні з іншими джерелами даних. Відповідність Art. 20 GDPR (право на перенесення даних) потребує окремої реалізації механізму експорту у машинозчитуваному форматі.

Перспективи подальшого розвитку охоплюють кілька взаємопов'язаних напрямів. Перший і найбільш технічно перспективний – інтеграція модулів предиктивного аналізу на основі нейронних мереж, зокрема рекурентних архітектур (LSTM, Transformer), здатних виявляти ранні маркери професійного вигоряння та когнітивної деградації на основі динаміки результатів психометричних тестів у часі. Ключовою умовою при цьому є збереження моделі PII Isolation: навчання та інференс мають здійснюватись виключно на псевдонімізованих клінічних даних, без звернення до ідентифікаційного вузла, що дозволить забезпечити аналітичну цінність без порушення анонімності суб'єктів дослідження.

Другий напрям – розширення каналів збору даних за рахунок мобільних застосунків, здатних реєструвати біометричні показники у реальному часі: варіабельність серцевого ритму, рівень фізичної активності, характеристики сну. Інтеграція таких даних із результатами психометричного тестування дозволить перейти від дискретних зрізів до безперервного моніторингу психоемоційного стану. При цьому виникатиме новий клас регуляторних викликів, пов'язаних із обробкою біометричних даних відповідно до Art. 9 GDPR (особливі категорії даних), що потребуватиме окремого проєктування схеми згоди та відповідних технічних засобів захисту.

Третій напрям стосується реалізації Art. 20 GDPR – права на перенесення даних. Наразі відсутній механізм експорту клінічного профілю користувача у машинозчитуваному форматі (JSON/XML), що є обов'язковою вимогою для систем, які обробляють персональні дані на основі згоди або договору. Реалізація цього механізму потребуватиме узгодження між ідентифікаційним та клінічним вузлами з одночасним забезпеченням дешифрування полів AES-256-GCM виключно для авторизованого суб'єкта запиту.

Перелік посилань

1. The Future of Jobs Report 2025. World Economic Forum. URL: https://www.weforum.org/publications/the-future-of-jobs-report-2025/.

2. World mental health today: latest data. World Health Organization (WHO). URL: https://www.who.int/publications/i/item/9789240113817.

3. Samarati P. Protecting Respondents' Identities in Microdata Release. IEEE Transactions on Knowledge and Data Engineering. 2001. Vol. 13, no. 6. P. 1010–1027. URL: https://www.cs.colostate.edu/~cs656/reading/samarati.pdf.

4. Sweeney L. K-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. International journal of uncertainty, fuzziness and knowledge-based systems. 2002. Vol. 10, no. 05. P. 557–570. DOI: https://doi.org/ 10.1142/s0218488502001648.

5. Machanavajjhala A., Kifer D., Gehrke J., Venkitasubramaniam M. l-Diversity: Privacy Beyond k-Anonymity. ACM Transactions on Knowledge Discovery from Data. 2007. Vol. 1, no. 1. Art. 3. DOI: https://doi.org/10.1145/1217299.1217302.

6. Li N., Li T., Venkatasubramanian S. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. IEEE Xplore. 2007. URL: https://ieeexplore.ieee.org/document/4221659.

7. Narayanan A., Shmatikov V. Robust De-anonymization of Large Sparse Datasets. IEEE Xplore. 2008. URL: https://ieeexplore.ieee.org/document/4531148.

8. Sung M., Cha D., Park Y. R. Local differential privacy in medical domain to protect sensitive information: Algorithm Development and Real-World Validation (Preprint). JMIR Medical Informatics. 2021. DOI: https://doi.org/10.2196/26914.

9. Cavoukian A. Privacy by Design: The 7 Foundational Principles. Computer Science Computing Facility | Computer Science Computing Facility (CSCF) | University of Waterloo. URL: https://student.cs. uwaterloo.ca/~cs492/papers/7foundationalprinciples_longer.pdf.

10. Regulation - 2016/679 - EN - gdpr - EUR-Lex. EUR-Lex – Access to European Union law. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.

11. OWASP Foundation. 2021. OWASP Application Security Verification Standard (ASVS) 4.0. URL: https://owasp.org/www-project-application-security-verification-standard/.

12. HL7 International. 2023. HL7 FHIR Release 5. URL: https://hl7.org/fhir/.

© Сліпченко В.Г., Полягушко Л.Г., Хнюкало В.С. Архітектурні підходи до забезпечення цілісності та конфіденційності даних у системах моніторингу психоемоційного стану. Сучасний захист інформації, 3(67), 128–142. https://doi.org/10.31673/2409-7292.2026.037413

13. Blobel, B., & Ruotsalainen, P. 2018. Privacy and Security in Personal Health Monitoring Systems. Studies in Health Technology and Informatics, 251, 3–15. URL: https://doi.org/10.3233/978-1-61499-922-3-3.

14. Ramírez, S. 2023. FastAPI: Concurrency and async/await. URL: https://fastapi.tiangolo.com/async/.

15. SQLAlchemy Authors. 2024. SQLAlchemy 2.0 Unified Tutorial. URL: https://docs.sqlalchemy.org/en/20/tutorial/.

16. PostgreSQL Global Development Group. 2024. Enumerated Types (8.7). PostgreSQL Documentation. URL: https://www.postgresql.org/docs/current/datatype-enum.html.

Volodymyr Slipchenko
Doctor of Technical Sciences, Professor, Professor of the Department of Digital Technologies in Energy
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0000-0002-3405-0781
E-mail: ddpolytechnic2016@gmail.com

Lyubov Polyagushko
Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Digital Technologies in Energy
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0000-0003-3287-8523
E-mail: liubovpoliagushko@gmail.com

Khnyukalo Viktoriia Serhiivna
student, Department of Digital Technologies in Energy
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0009-0006-5913-2958
E-mail: vi.hnuuu@gmail.com

ARCHITECTURAL APPROACHES TO ENSURING DATA INTEGRITY AND CONFIDENTIALITY IN PSYCHO-EMOTIONAL STATE MONITORING SYSTEMS

The article explores architectural approaches to personal data management in psycho-emotional monitoring systems using the author's digital platform "Sensera" as an example. The fundamental contradiction between the need for deep longitudinal analytics of psycho-emotional states and strict requirements for the protection of personally identifiable information (PII) is considered. The aim of the work is the theoretical justification and practical testing of the architectural model "PII Isolation", which provides physical and logical separation of identification and clinical data directly at the level of the relational database schema. The study is based on the methods of comparative analysis of approaches to PII protection (k-anonymity, differential privacy, Privacy by Design), structural modeling of the database schema and practical prototyping on the FastAPI 0.128.0 / PostgreSQL 17.4 / SQLAlchemy 2.0 stack. Additionally, the unification of metadata using enumerated types (ENUM) and the optimization of asynchronous query processing through the Eager Loading strategy (joinedload/ selectinload) were investigated. Practical testing on a sample of 30 clinical profiles and 102 assessments confirmed the effectiveness of the solutions: automated verification of PII isolation was completed with the status PASSED in accordance with GDPR Art. 4(5); the Eager Loading strategy reduced the number of SQL queries from 11 to 1 (-91%); the implementation of 4 ENUM types eliminated 4 classes of incorrect input errors and reduced the estimated query cost by 9.4% with absolute accuracy of the planner statistics (14/14 versus 1/14 for TEXT CAST). The compliance of the architecture was confirmed according to 8 articles of Regulation (EU) 2016/679. It is concluded that the proposed approach implements the principle of "positive sum" - provides full analytical accuracy without compromising privacy. The practical value lies in the reproducibility of the model for scalable e-Health platforms that require simultaneous implementation of GDPR and HIPAA requirements.

Keywords: PII isolation, data integrity, psycho-emotional monitoring, PostgreSQL, FastAPI, metadata unification, personal data protection.

Надійшла до редакції (Received): 05.08.2026
Прийнята до друку (Accepted): 08.09.2026
Опубліковано онлайн (Available online): 15.09.2026

© Сліпченко В.Г., Полягушко Л.Г., Хнюкало В.С. Архітектурні підходи до забезпечення цілісності та конфіденційності даних у системах моніторингу психоемоційного стану. Сучасний захист інформації, 3(67), 128–142. https://doi.org/10.31673/2409-7292.2026.037413