УДК 004.85:004.75:004.056 DOI: 10.31673/2412-4338.2026.019001

Барабаш Олег Володимирович
доктор технічних наук, професор, професор кафедри інженерії програмного забезпечення в енергетиці
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ
ORCID 0000-0003-1715-0761
E-mail: bar64@ukr.net

Бандурка Олена Іванівна
доктор філософії, старший викладач кафедри інженерії програмного забезпечення в енергетиці
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ
ORCID 0000-0002-8059-1861
E-mail: o.i.bandurka@ukr.net

Свинчук Ольга Василівна
кандидат фізико-математичних наук, доцент, доцент кафедри інженерії програмного забезпечення в енергетиці
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ
ORCID 0000-0001-9032-6335
E-mail: 7011990@ukr.net

Файдюк Тарас Русланович
магістр кафедри інженерії програмного забезпечення в енергетиці
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ
ORCID 0009-0002-5464-8497
E-mail: taras.fid@gmail.com

ОРГАНІЗАЦІЯ БЕЗПЕРЕРВНОГО НАВЧАННЯ В LMS НА ОСНОВІ АРХІТЕКТУРИ OFFLINE-FIRST ТА ЗАХИЩЕНОГО ТЕСТУВАННЯ

Анотація: *Стаття присвячена проблемі забезпечення безперервності та якості навчального процесу в системах керування навчальним процесом (LMS) в середовищі нестабільного інетернет-з'єднання. Синхронізація та безпечне офлайн-тестування є досить важливою складовою вищої освіти в умовах дистанційного навчання при відсутності стабільного енергопостачання та при невідкладності створення рівноцінних умов для всіх здобувачів освіти. У них вдало поєднані надійність традиційних методів з гнучкістю інформаційних технологій. З огляду на ситуацію в Україні та непередбачуваність в навчальному процесі, забезпечення безперервного функціонування таких систем є критично важливим для ефективності та якості навчального процесу на будь-якому рівні освіти. Зокрема, основними викликами для таких систем зараз є стійкість до зміни мережевого та світло-постачання, захист від фальсифікування, а також забезпечення цілісності та достовірності даних.*

Метою дослідження є розробка архітектури синхронізації навчальних даних та безпечного офлайн-тестування в LMS, яка забезпечить безперервність навчального процесу за умов нестабільного інтернет-з'єднання, перебоїв електропостачання та динамічних змін навантаження. Основний акцент зроблено на моделі Offline-first, механізмах надійної синхронізації та відновлення стану, а також на засобах контролю цілісності й достовірності навчальних даних. Запропонована архітектура передбачає локальне збереження критичних даних і дій користувачів (відповідей на тести, прогресу, подій навчальної активності) з подальшим узгодженням із сервером після відновлення зв'язку. Для зменшення ризиків втрати або підміни інформації застосовуються криптографічні механізми перевірки, журналювання подій та кероване версіонування, що дозволяє виявляти конфлікти синхронізації й коректно їх розв'язувати без втрати якості навчального процесу.

Ключові слова: *система керування навчанням (LMS), модель Offline-first, архітектура системи, синхронізація даних, журнал подій,, цілісність даних, конфлікти версій, криптографічний захист.*

Barabash Oleh
Doctor of Technical Sciences, Professor, Professor of the Department of Software Engineering in Energy Systems
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv
ORCID 0000-0003-1715-0761
E-mail: bar64@ukr.net

Bandurka Olena
PhD, Senior Lecturer of the Department of Software Engineering in Energy Systems
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv
ORCID 0000-0002-8059-1861
E-mail: o.i.bandurka@ukr.net

Svynchuk Olha
Candidate of Physical and Mathematical Sciences, Associate Professor, Associate Professor of the Department of Software Engineering in Energy Systems
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv
ORCID 0000-0001-9032-6335
E-mail: 7011990@ukr.net

Faidiuk Taras
Master's student of the Department of Software Engineering in Energy System
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv
ORCID 0009-0002-5464-8497
E-mail: taras.fid@gmail.com

ORGANIZING CONTINUOUS LEARNING IN LMS BASED ON OFFLINE-FIRST ARCHITECTURE AND SECURE TESTING

Abstract: *The article is devoted to the problem of ensuring the continuity and quality of the educational process in Learning Management Systems (LMS) in an environment of unstable internet connection. Synchronization and secure offline testing are vital components of higher education under distance learning conditions, characterized by the absence of a stable power supply and the urgent need to create equal conditions for all learners. These components successfully combine the reliability of traditional methods with the flexibility of information technologies. Given the situation in Ukraine and the unpredictability of the educational process, ensuring the continuous operation of such systems is critical for the efficiency and quality of education at any level. In particular, the main challenges for such systems today are resilience to network and power supply fluctuations, protection against falsification, and ensuring data integrity and authenticity.*

The purpose of the study is to develop an architecture for educational data synchronization and secure offline testing in LMS, which ensures educational process continuity under conditions of unstable internet connection, power outages, and dynamic load changes. The primary focus is placed on the Offline-first model, reliable synchronization and state recovery mechanisms, as well as tools for controlling the integrity and authenticity of educational data. The proposed architecture provides for the local storage of critical data and user actions (test responses, progress, educational activity events) with subsequent reconciliation with the server after the connection is restored. To reduce the risks of information loss or tampering, cryptographic verification mechanisms, event logging, and managed versioning are used, allowing for the detection and correct resolution of synchronization conflicts without compromising the quality of the educational process.

Keywords: *Learning Management System (LMS), Offline-first model, system architecture, data synchronization, event log, data integrity, version conflicts, cryptographic protection.*

1. Постановка проблеми.

У сучасному високотехнологічному світі дистанційний формат навчання вже давно став нормою, а подекуди і базовою опцією, якій надають перевагу перед очним форматом освітнього процесу. Тому підтримка цього процесу є надважливим, якщо не ключовим, завданням інженера, який працює над системою забезпечення навчання.

Для України додатковим фактором є нерівномірність і непередбачуваність доступу до інфраструктури з частими перериваннями та втратами зв’язку, що підсилює потребу в офлайн-сценаріях навчання й оцінювання. Статистика, сформована на основі даних МОН, демонструє

співіснування очного, дистанційного та змішаного форматів навчання, що об'єктивно зумовлює потребу в цифрових рішеннях, здатних стабільно працювати в різних режимах доступності мережі [1].

В електронному навчанні (e-learning) LMS розглядають як базову інфраструктуру для організації контенту, комунікації, контролю й оцінювання. Також e-learning охоплює інтерактивні лекції, симулятори, інтерактивні тести та інші формати, що активно застосовуються у закладах вищої освіти [2]. Водночас у реальних умовах «офлайн-режим» не зводиться лише до кешування контенту: критичними є збереження прогресу, синхронізація станів між пристроями, а також забезпечення достовірності результатів тестування.

Ці групи проблем є найбільш складними: надійна синхронізація навчальних даних (прогресу, виконаних завдань, оцінок, коментарів) з розв'язанням конфліктів та контролем версій, а також безпечне офлайн-тестування, де необхідно одночасно забезпечити збереження відповідей, конфіденційність, цілісність і можливість перевірки на сервері. Наприклад, досвід однієї з найпопулярніших LMS, Moodle, показує, що офлайн-спроби тестів можливі лише за певних обмежень, а сам офлайн-режим для квізів та уроків за замовчуванням вимкнено і його має активувати викладач [3]. Це підтверджує складність проблеми й потребу в більш формалізованих архітектурних підходах.

2. Аналіз останніх досліджень і публікацій.

Питання цифровізації освіти та еволюції електронного навчання в науковій літературі розглядаються як закономірний етап розвитку освітніх практик під впливом розвитку інформаційно-комунікаційних технологій. У роботі [1] електронне навчання описується як багатокомпонентне явище, що охоплює організацію контенту, комунікацію та контроль навчальних результатів у цифровому середовищі, а також підкреслюється зростання ролі студента як активного суб'єкта освітньої діяльності. Для України додатковим визначальним чинником виступає нестабільність доступу до інфраструктури, що трансформує вимоги до LMS у критично необхідні, наприклад, здатність підтримувати навчальний процес за непередбачуваних перерв зв'язку та відсутності електропостачання. Статичні дані МОН щодо співіснування очного, дистанційного та змішаного форматів навчання підтверджують об'єктивну потребу в системах, які можуть працювати в різних режимах доступності мережі [2].

На рівні прикладних LMS-рішень одним із найпоширеніших прикладів є екосистема Moodle. В документації даної системи зазначається, що користувачі можуть переглядати частину навчальних матеріалів та виконувати окремі активності офлайн з подальшою синхронізацією після відновлення з'єднання [3]. Водночас такий підхід, як правило, залежить від типу активності та налаштувань, а тому не забезпечує універсального сценарію «офлайн-навчання» без додаткових архітектурних механізмів контролю цілісності, відтворюваності і синхронізації результатів. Саме це підсилює актуальність розробки архітектури, де офлайн-робота є не додатковою функцією, а базовою вимогою для критичних функцій LMS.

З погляду реалізації offline-first у вебсередовищі суттєвий внесок роблять стандартизовані Web API. У специфікації Service Workers [4] описано подієво-орієнтований механізм фонової обробки, який надає можливість перехоплювати мережеві запити, підтримувати кешування та сценарії роботи без мережі. Для локального зберігання структурованих даних у браузері ключовим є стандарт IndexedDB [5], що визначає API для транзакційного сховища записів із ключами та індексами, придатного для збереження значних обсягів даних навчальної активності студентів. специфікація Web Background Synchronization [6] пропонує механізм відкладеної синхронізації у фоновому режимі через service worker, що дозволяє здійснювати передачу накопичених подій після стабілізації підключення. У комплексі ці технології створюють технічне підґрунтя для побудови LMS, яка гарантує збереження даних користувача в умовах нестабільного зв'язку, однак самі по собі вони не визначають логіку узгодження станів і розв'язання конфліктів.

Окремою проблемою є синхронізація та узгодження навчальних даних між клієнтом і сервером за наявності конфліктуючих змін, повторних доставок інформації і часткових збоїв під час передачі. У фундаментальній праці Марка Шапіро систематизовано підхід Conflict-free Replicated Data Types (CRDT), який забезпечує збіжність реплік за умов асинхронної реплікації та конкурентних оновлень, зменшуючи потребу в централізованому блокуванні [7]. Такий підхід є перспективним для окремих класів навчальних даних, проте в LMS існують доменно-специфічні обмеження: частина подій має бути незворотною (наприклад, завершення спроби тесту), інші – підлягати строгій валідації правил, а конфлікти – розв'язуватися детерміновано з урахуванням правил-політик. Відтак застосування CRDT у навчальних системах потребує поєднання теоретичних гарантій збіжності з прикладними правилами доменної частини системи.

Також для забезпечення простежуваності та можливості відтворення станів доцільним є використання підходу журналювання змін через події. Можна використати патерн, за якого всі зміни стану системи фіксуються у вигляді послідовності подій, що дозволяє відновлювати попередні стани та виконувати аудит [8]. Для LMS це має принципове значення у контексті офлайн-оцінювання: журнал подій може зберігати атомарні кроки взаємодії студента з тестом, забезпечуючи відтворюваність, доказовість та основу для серверної перевірки після відновлення зв'язку. Водночас, на відміну від звичайного логування, журнал подій у контексті безпеки має підлягати криптографічному захисту від підміни, а також мати механізми дедуплікації та впорядкування для стійкої синхронізації.

Питання конфіденційності та цілісності навчальних даних в офлайн-сценаріях тісно пов'язані з криптографічними стандартами та практиками безпечної розробки. Специфікація «Web Cryptography Level 2» визначає JavaScript API для базових криптографічних операцій (хешування, підпис/перевірка, шифрування/дешифрування) та керування ключовим матеріалом у вебзастосунках, що створює основу для реалізації криптографічного захисту на стороні клієнта [9]. Для забезпечення одночасно конфіденційності та цілісності даних при збереженні відповідей і подій доцільним є використання режимів автентифікованого шифрування [10]. Для керованого формування ключів з урахуванням контексту та вимог застосунку релевантним є HKDF – HMAC-based Extract-and-Expand Key Derivation Function [11]. Такий підхід дозволяє створювати унікальні ключі для кожного сеансу офлайн-навчання, що мінімізує ризики в разі компрометації одного з пристроїв.

3. Мета і задачі дослідження.

Проблематика забезпечення безперервності навчального процесу в LMS за умов нестабільного інтернет-з'єднання та перебоїв електропостачання є комплексною, оскільки охоплює надійність зберігання даних на клієнті, узгодження станів при синхронізації, протидію фальсифікації та виконання вимог інформаційної безпеки й законодавства щодо персональних даних. На відміну від сценаріїв простого кешування контенту, в освітньому процесі критичними є фіксація прогресу та доказовість результатів оцінювання. Система має гарантувати, що офлайн-дії студента не будуть втрачені, а після відновлення зв'язку – коректно доставлені, впорядковані та інтегровані до серверного стану без втрати цілісності. У цьому контексті доцільним є поєднання подієвих моделей фіксації змін, технологій офлайн-функціонування вебклієнтів, алгоритмів узгодження конкурентних оновлень, а також криптографічних стандартів для забезпечення конфіденційності й відповідності правовим вимогам щодо захисту даних [12].

Метою дослідження є розробка архітектури синхронізації навчальних даних та безпечного офлайн-тестування в LMS, що базується на використанні журналу подій, керованого версіонування та криптографічних механізмів для забезпечення цілісності, достовірності й відтворюваності результатів за умов нестабільного інтернет-з'єднання та часткових відмов. Архітектура має враховувати специфіку змішаного навчання та обмеження наявних підходів офлайн-режиму в сучасних LMS, зокрема з позиції повноцінного офлайн-оцінювання.

Основними завданнями дослідження є:

- формалізація структури події та журналу подій LMS для офлайн-сценаріїв із можливістю відтворення станів;
- обґрунтування технологічного стеку вебклієнта для offline-first (service worker, локальне сховище, фонові задачі);
- проєктування протоколу синхронізації пакетів подій (дедуплікація, ідемпотентність, часткові підтвердження, детерміновані правила конфліктів);
- розробка механізму безпечного офлайн-тестування (шифрування відповідей, хеш-ланцюжок подій, «печатка» спроби для серверної перевірки);
- визначення критеріїв оцінювання архітектури та демонстрація практичного сценарію впровадження на типових LMS-процесах.

4. Результати дослідження.

У типових LMS узгодженість даних та коректність оцінювання фактично залежать від безперервного доступу до мережі. За умов стабільного інтернет-з'єднання такий підхід спрощує архітектуру: клієнт передає дії на сервер, сервер одразу валідує та фіксує стан. Проте в реальних умовах, коли можливі тривалі переривання або відсутність зв'язку та електропостачання, така модель має критичні недоліки: втрата частини дій користувача, розриви спроб тестування, дублювання запитів при повторних відправках, а також підвищення ризику фальсифікації результатів через відсутність повного аудиту подій на клієнтському боці.

З огляду на це доцільним є перехід до Offline-first підходу, де первинним джерелом фіксації дії

стає локальне сховище, а мережа розглядається як канал реплікації та верифікації. У межах цієї роботи запропоновано практично орієнтовану архітектуру, в якій навчальна активність описується як послідовність подій із фіксованою схемою та можливістю відтворення станів, синхронізація виконується пакетами з частковими підтвердженнями, дедуплікацією та детермінованими правилами узгодження, а для офлайн-оцінювання формується «печатка» спроби – криптографічно захищений артефакт, що дозволяє серверу перевірити цілісність журналу й коректність відповідей після відновлення з'єднання.

Практична реалізація Offline-first у веб-середовищі спирається на стандартні механізми: service worker забезпечує перехоплення мережевих запитів, кешування та виконання фонових задач, а локальне транзакційне сховище (наприклад, IndexedDB) використовується для збереження журналу подій, метаданих спроб і чернеток відповідей. Синхронізація запускається при відновленні з'єднання та відправляє накопичені події пакетами, що зменшує втрати й дублікати під час нестабільної мережі.

Для порівняльного аналізу архітектурних рішень розглянуто три моделі функціонування LMS в умовах нестабільного з'єднання: традиційну Online-only, гібридну модель із базовим кешуванням та запропоновану Offline-first модель. Візуалізацію структурних відмінностей між цими підходами представлено на рисунку 1.

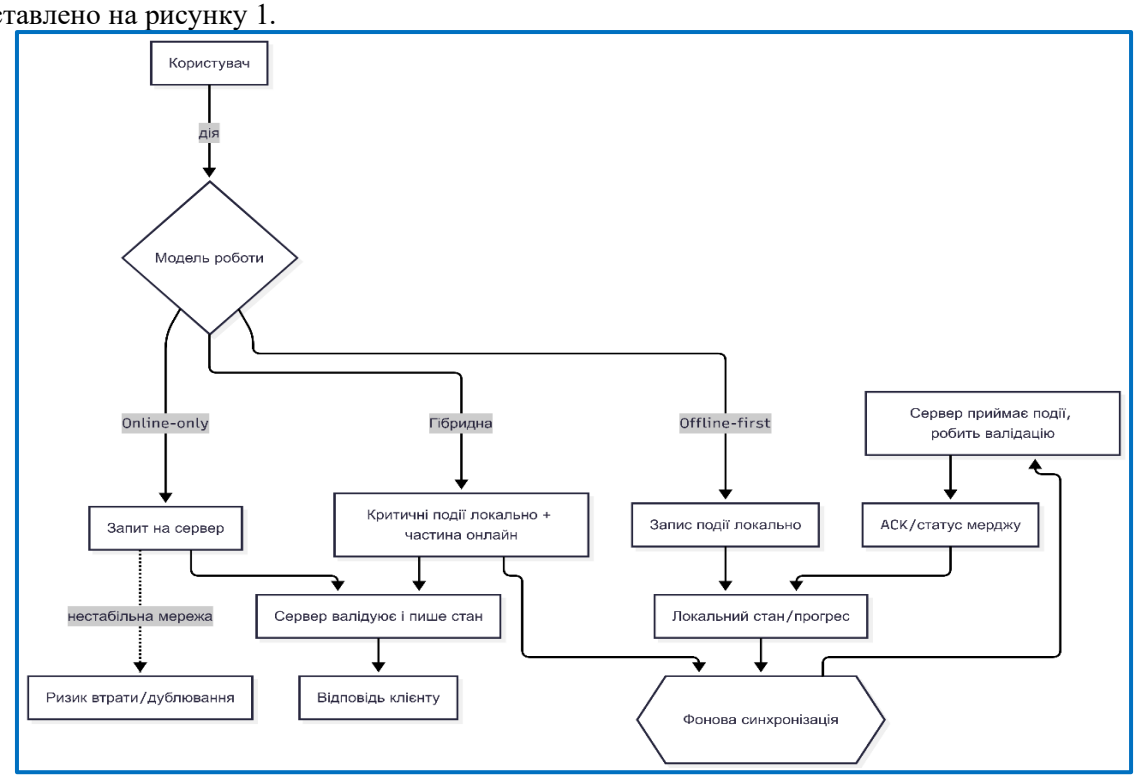

Рис.1. Архітектурні моделі взаємодії компонентів LMS в умовах різного стану мережевого з'єднання

Таблиця 1

Порівняльний аналіз характеристик моделей функціонування LMS

Властивість/Поведінка	Online-only	Offline-first	Гібридна
Точка відмови	Мережа як залежність	Локальне сховище і мережа для реплікації	Мережа як залежність
Збереження прогресу	Ризик втрати при розриві	Гарантоване локальне фіксування подій	Залежить від типу активності
Конфлікти	Мінімальні	Потрібні правила мерджу/версії	Потрібні для частини даних
Масштабування	Сервер обробляє всі дії онлайн	Сервер приймає пакети подій, менше піків	Компроміс
Аудит та відтворюваність	Логи здебільшого серверні	Подієвий журнал на клієнті й аудит	Змішаний підхід
Ризики фальсифікації офлайн	Не актуально	Є	Є для офлайн-частини

Поряд із візуальним представленням архітектурних схем, у таблиці 1 наведено детальний порівняльний аналіз зазначених моделей за ключовими критеріями: надійністю збереження даних, механізмами синхронізації та рівнем забезпечення цілісності результатів оцінювання.

Отже, за умов нестабільного мережевого середовища модель Offline-first із використанням локального журналу подій забезпечує найбільш стійке функціонування системи: прогрес користувача гарантовано зберігається, а процес синхронізації трансформується з окремих онлайн-запитів у надійну реплікацію підтверджених подій.

Відтак, подальший виклад матеріалу зосереджено на формалізації структури події, розробці протоколу пакетної синхронізації та впровадженні механізмів контролю цілісності результатів офлайн-діяльності.

Ключовою одиницею синхронізації виступає подія. Мінімальний набір полів події, достатній для надійної реплікації, може включати: унікальний ідентифікатор, тип події, часову мітку клієнта, ідентифікатор контексту, корисне навантаження, номер версії, а також ключ ідемпотентності. Така структура дозволяє серверу відкинути дублікати, відновити порядок та виконати доменну валідацію. Для частини даних, які допускають комутативні/асоціативні операції, наприклад, накопичення прогресу, деякі типи нотаток, можуть застосовуватись ідеї CRDT для забезпечення збіжності реплік без жорсткого блокування. Проте для оцінювання завершальної спроби чи фінальної відповіді правило має бути суворим і детермінованим. Реалізація такого підходу потребує чітко визначеного циклу обробки даних на пристрої студента. На рисунку 2 представлено алгоритм локальної фіксації навчальних подій на стороні клієнта, що охоплює їх реєстрацію у локальному журналі та підготовку пакета даних до відкладеної доставки.

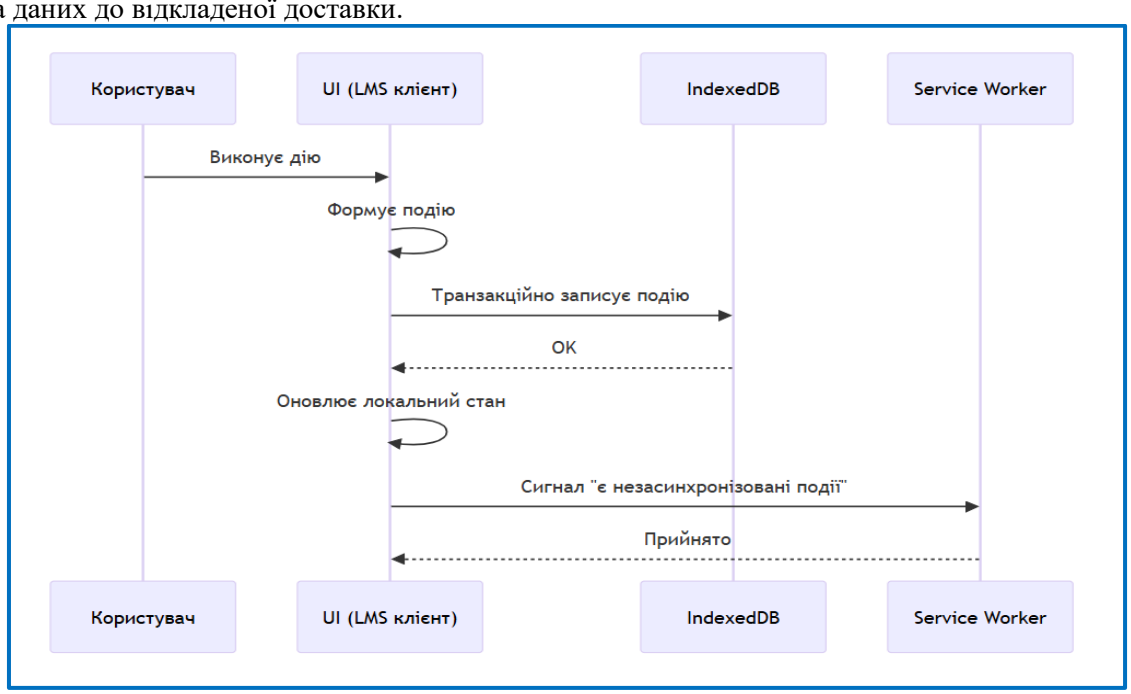

Рис.2. Алгоритм локальної фіксації навчальних подій на стороні клієнта

Синхронізація подій реалізується пакетами. Клієнт формує пакет із подій, що ще не підтверджені сервером, і відправляє його через захищений канал зв'язку. Сервер перевіряє коректність авторизації, валідність структури, виконує дедуплікацію та застосовує події до серверного стану. Після цього сервер повертає підтвердження із переліком застосованих подій і, за потреби, із даними для узгодження. Клієнт позначає підтверджені події як синхронізовані та оновлює локальний стан. На рисунку 3 представлено алгоритм серверної обробки подій та формування підтверджень, необхідних для забезпечення ідемпотентності й дедуплікації даних при синхронізації.

Безпечне офлайн-тестування є найскладнішою частиною, оскільки вимагає поєднати взаємовиключні вимоги: автономність клієнта та результатів, що може перевірити сервер. Практичним підходом є розділення артефактів тесту на: відкриті метадані (ідентифікатор тесту, версія, часові обмеження) та критичні дані (відповіді, токени сесії, проміжні стани), які мають зберігатися у зашифрованому вигляді та супроводжуватись криптографічними доказами цілісності.

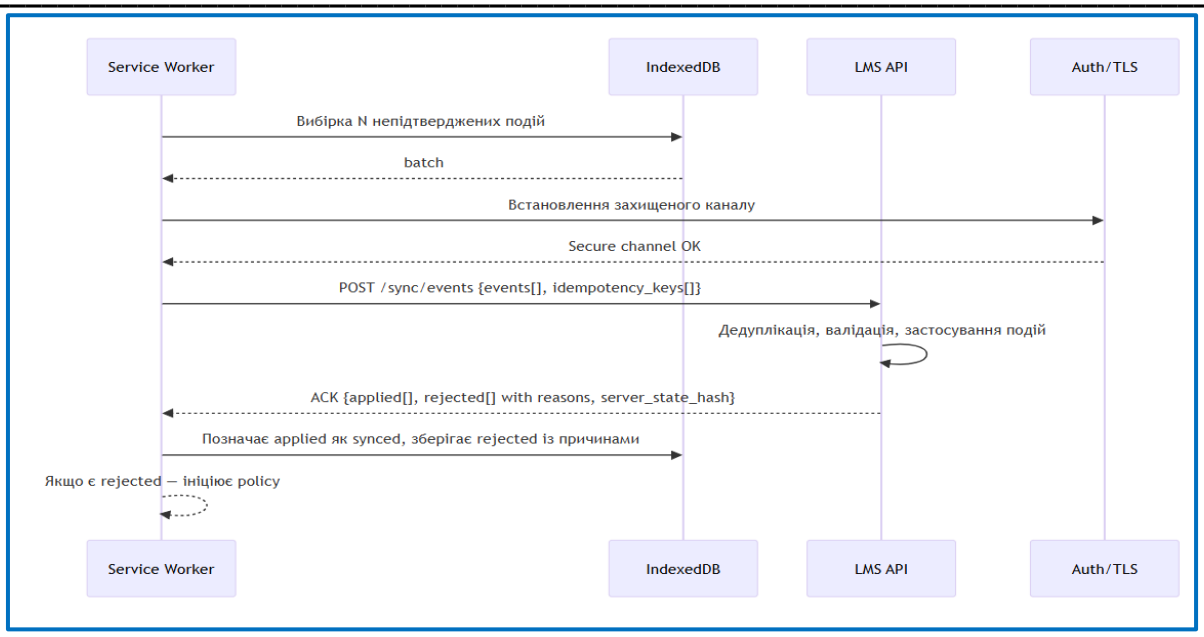

Рис.3. Алгоритм серверної обробки подій та забезпечення ідемпотентності синхронізації

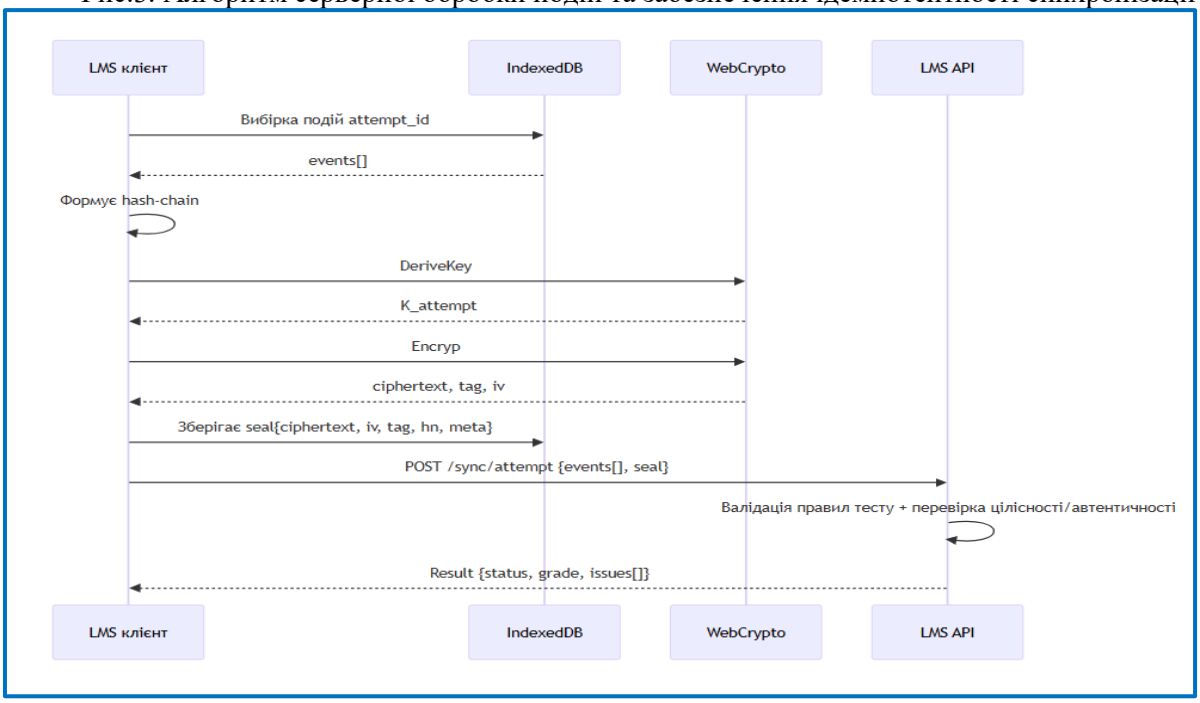

Рис.4. Алгоритм завершення офлайн-спроби

На рівні клієнта доцільно використовувати Web Crypto API для генерації/зберігання ключового матеріалу та виконання криптографічних операцій [9]. Для шифрування офлайн-відповідей і «печатки» спроби придатним є режим автентифікованого шифрування AES-GCM, який забезпечує одночасно конфіденційність і контроль цілісності. Для похідних ключів, наприклад ключ на конкретну спробу або ключ на конкретний блок відповідей, застосовується HKDF, що дозволяє безпечно отримувати ключі з майстер-секрету з урахуванням контексту [11]. Для прив'язки журналу подій тесту до незмінної послідовності доцільно використовувати хеш-ланцюжок на базі стандартизованої хеш-функції, що уможливлює виявлення видалення/вставки/перестановки подій.

У типовому сценарії офлайн-тестування клієнт отримує від сервера параметри тесту та стартовий токен спроби, локально фіксує події відповідей у журналі, періодично формує контрольну суму для поточного стану спроби, після завершення спроби створює «печатку» – криптографічно захищений об'єкт, який включає зашифровані відповіді та підсумковий хеш-ланцюжок журналу. Після відновлення мережі клієнт передає на сервер події та печатку, а сервер виконує перевірку: коректність правил, узгодженість журналу, розшифрування/перевірку автентичності, та обчислення оцінки.

На рисунку 4 наведено узагальнений алгоритм завершення офлайн-спроби як окрему функцію.

На рисунку 5 зображено загальну архітектуру запропонованої системи на рівні компонентів, яка реалізує принцип «події спочатку»: кожна критична дія користувача фіксується локально, а сервер виконує контроль та узгодження після відновлення зв'язку. Це зменшує залежність від безперервної мережі та підвищує відтворюваність навчального процесу за рахунок журнальної моделі.

Архітектура організована як двоконтурна система «клієнт–сервер» із чітко розмежованими зонами відповідальності: клієнт забезпечує запис подій у локальне сховище та формування пакетів синхронізації, тоді як сервер реалізує ідемпотентне приймання подій, дедуплікацію, валідацію послідовності та розв'язання конфліктів. Узгодження стану виконується через кінцеві точки API з частковими підтвердженнями, що дає змогу коректно відновлювати прогрес навіть за повторних запитів і переривань сеансу. Для критичних сценаріїв сервер може постфактум підтвердити незмінність журналу та автентичність результатів без вимоги постійної онлайн-присутності клієнта.

При цьому безпечне офлайн-оцінювання досягається комбінацією криптографічних механізмів на стороні клієнта та процедур серверної перевірки: перевіркою послідовності подій, валідністю часових обмежень, цілісністю «печатки» та доменною валідацією правил тесту.

З позиції захисту персональних даних архітектура передбачає мінімальну кількість локально збережених даних, контроль строків зберігання та розмежування доступу до журналів подій. Практично це реалізується політиками життєвого циклу даних (TTL), шифруванням локальних артефактів та обмеженням експорту журналів із клієнта.

Практичне впровадження запропонованої архітектури можливе як модуль PWA для наявної LMS або як окремий вебклієнт, що підключається до серверного API. На клієнті формується локальний журнал подій у транзакційному сховищі з індексами за user_id, attempt_id і status, що дозволяє: зберігати події навіть при втраті живлення; відновлювати стан інтерфейсу після перезапуску; відправляти накопичені події пакетами після відновлення мережі.

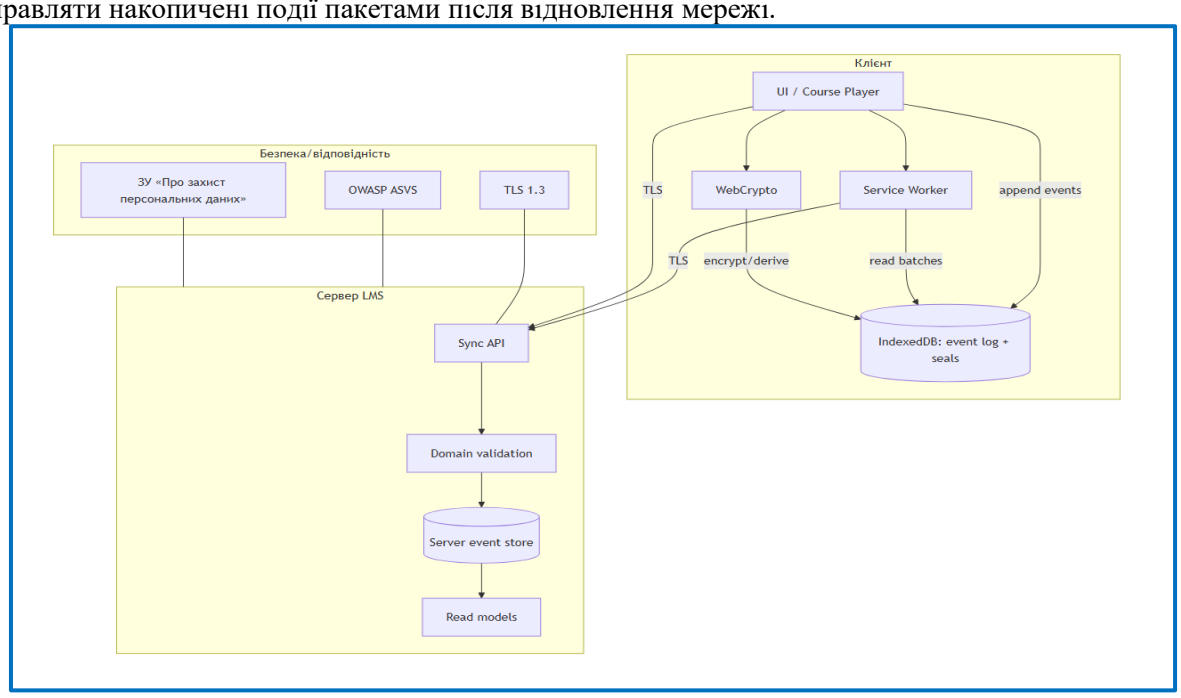

Рис.5. Загальна схема архітектурної взаємодії компонентів Offline-first LMS

На сервері реалізуються дві базові точки отримання запиту: /sync/push (прийом пакету подій і повернення підтверджень) та /sync/pull (видача актуального стану/версій для узгодження). Ідемпотентність забезпечується ключем події та контрольним номером версії, а для контролю цілісності ланцюжка подій пропонується використовувати хеш-зв'язування:

$$h_i = H(h_{i-1} || event_i).$$

де H — криптографічна хеш-функція, $||$ — операція конкатенації, h_{i-1} — попереднє значення хеш-ланцюга, а $event_i$ — канонічно серіалізована подія з журналу (тип, час, дані), подана у детермінованому форматі, що виключає неоднозначність представлення [13]. Дана формула задає спосіб побудови хеш-ланцюжка як криптографічного інваріанта цілісності: будь-яка зміна, вилучення або перестановка подій призводить до зміни фінального контрольного значення. Під час синхронізації сервер відтворює ланцюжок на основі отриманих подій та звіряє контрольний хеш із тим, що міститься в «печатці», отримуючи доказ незмінності послідовності активності клієнта.

Для перевірки ефективності розроблених рішень проведено оцінювання архітектури за раніше визначеними критеріями (див. табл. 1) та апробовано практичний сценарій її функціонування. У межах сценарію розглянуто процес контрольного тестування: студент виконує завдання в офлайн-режимі, ініціюючи генерацію потоку подій. Після відновлення зв'язку клієнтська сторона передає пакет даних разом із криптографічною «печаткою» спроби. Сервер верифікує цілісність хеш-ланцюжка, перевіряє автентичність зашифрованих відповідей та дотримання часових регламентів тесту. За умови успішної валідації сервер опрацьовує події, фіксує підсумковий результат і повертає підтвердження, після чого клієнтський додаток очищує локальне сховище згідно з політикою зберігання артефактів.

5. Висновки і перспективи подальших досліджень.

У статті розглянуто проблему забезпечення безперервності та якості навчального процесу в системах керування навчанням в умовах нестабільного інтернет-з'єднання та перебоїв електропостачання. У реальних сценаріях змішаного навчання ключовими є не лише доступність навчального контенту, а насамперед збереження прогресу, відтворюваність навчальної активності та доказовість результатів оцінювання, що підтверджує актуальність моделі Offline-first для LMS.

Розроблена архітектура ґрунтується на фіксації критичних дій користувача у вигляді журналу подій з керованим версіонуванням і механізмами надійної синхронізації та відновлення стану. Використання Service Workers, IndexedDB та механізмів фонової синхронізації забезпечує технічну можливість стабільної роботи вебклієнта без мережі, а поєднання подієвої моделі з детермінованими правилами узгодження дозволяє виявляти конфлікти, забезпечувати ідемпотентність змін і зменшувати ризики втрати даних під час повторних спроб синхронізації.

Для безпечного офлайн-тестування обґрунтовано застосування криптографічних механізмів на стороні клієнта: автентифікованого шифрування відповідей і артефактів спроби, контролю цілісності журналу подій хеш-ланцюжками та захищеного транспорту даних. З урахуванням практик OWASP та вимог Закону України «Про захист персональних даних» визначено принципи мінімізації локально збережених даних, строків зберігання та контролю доступу.

Запропоновані підходи можуть бути використані при проєктуванні нових систем дистанційного навчання або для модернізації існуючих платформ шляхом інтеграції Offline-first архітектури. Це сприятиме створенню надійних, безпечних і відмовостійких рішень, що дозволить суттєво підвищити стабільність освітнього процесу в умовах нестабільного мережевого з'єднання.

Внесок авторів

Олег БАРАБАШ – формалізація гіпотези дослідження; Олена БАНДУРКА – аналіз джерел та підготовка огляду літератури; Ольга СВИНЧУК – концептуалізація, збір і перевірка емпіричних даних, візуалізація результатів; Тарас ФАЙДЮК – програмне забезпечення, емпіричне дослідження.

Декларація про штучний інтелект

Автор не використовував штучний інтелект при створенні матеріалів статті.

Конфлікт інтересів

Автор заявляє про відсутність конфлікту інтересів та підтверджує, що під час підготовки цієї роботи не існувало жодних комерційних, фінансових чи інших взаємовідносин, які могли б бути розцінені як такі, що здатні вплинути на результати дослідження або їх інтерпретацію. Робота виконана відповідно до принципів академічної доброчесності, етичних норм проведення наукових досліджень та вимог редакційної політики щодо запобігання конфлікту інтересів.

6. Список використаної літератури

1. Топузов О., Локшина О., Головко М. Навчальні втрати: складність проблеми в умовах війни в Україні. Education: Modern Discourses. 2024. № 6. С. 7–17. https://doi.org/10.37472/2617-3107-2023-6-01

2. Глубока С. Електронне навчання (e-learning): сутність, походження та етапи розвитку, особливості застосування у ЗВО. Актуальні питання гуманітарних наук. 2024. Вип. 76, том 1. С. 242-248. https://doi.org/10.24919/2308-4863/76-1-37

3. Cole J., & Foster H. Using MOODLE: Teaching with the Popular Open Source Course Management System (2nd ed.). Sebastopol, California: O'Reilly, 2007. 266 p. URL: https://issuu.com/tparks/docs/moodle

4. Service Workers. Nightly publication history Standards. 13 November 2025. URL: https://www.w3.org/TR/service-workers

5. Васильківськмй М., Нікітович Д., Болдирева О. Керування доступом до інформаційних даних в

інтелектуальних інфокомунікаційних мережах. Вимірювальна та обчислювальна техніка в технологічних процесах. 2022. № 4. С. 5-17. https://doi.org/10.31891/2219-9365-2022-72-4-1

6. Xu H., Yu S., Jin S., Sun R., Chen G., Sun L. Enhancing robustness in asynchronous feature tracking for event cameras through fusing frame steams. Complex & Intelligent Systems. 2024. Vol. 10 (2). P. 6885–6899 https://doi.org/10.1007/s40747-024-01513-0

7. Shapiro M., Preguiça N., Baquero C., Zawirski M. A comprehensive study of Convergent and Commutative Replicated Data Types (CRDTs). 2011. INRIA Research Report. No. 7506. 47 p. URL: https://dsf.berkeley.edu/cs286/papers/crdt-tr2011.pdf

8. Overeem M., Spoor M., Jansen S. An empirical characterization of evensourced systeam their schema evolution – Lessons from industry. Journal of Systems Software. 2021. Vol. 178. 110970, ISSN 0164-1212. https://doi.org/10.1016/j.jss.2021.110970

9. Web Cryptography Level 2. Standart. W3C TR, 22 April 2025. URL: https://www.w3.org/TR/webcrypto-2/

10. Dworkin M. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D. National Institute of Standards and Technology, 2007. 37 p. https://doi.org/10.6028/NIST.SP.800-38D

11. Krawczyk H., Eronen P. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869. Internet Engineering Task Force (IETF), 2010. 14 p. https://www.rfc-editor.org/rfc/rfc5869

12. Закон України «Про захист персональних даних». Верховна Рада України. URL: https://zakon.rada.gov.ua/laws/show/2297-17

13. Schneier B., Kelsey J. Secure Audit Logs to Support Computer Forensics ACM Transactions on Information and System Security. 1999. Vol. 2, No. 2. P. 159–176. https://dl.acm.org/doi/pdf/10.1145/317087.317089

7. References

1. Topuzov, O., Lokshyna, O., & Holovko, M. (2024). Learning losses: the complexity of the problem in the context of the war in Ukraine. Education: Modern Discourses, (6), 7–17. https://doi.org/10.37472/2617-3107-2023-6-01

2. Hluboka, S. (2024). E-learning: essence, origin and stages of development, features of application in higher education institutions. Current Issues of the Humanities, 76(1), 242-248. https://doi.org/10.24919/2308-4863/76-1-37

3. Cole, J., & Foster, H. (2007). Using MOODLE: Teaching with the Popular Open Source Course Management System (2nd ed.). O'Reilly Media. https://issuu.com/tparks/docs/moodle

4. W3C. (2025). Service Workers. Nightly publication history Standards. https://www.w3.org/TR/service-workers

5. Vasylkivskyi, M., Nikitovych, D., & Boldyrieva, O. (2022). Information data access mana5. gement in intelligent infocommunication networks. Measuring and Computing Devices in Technological Processes, (4), 5-17. https://doi.org/10.31891/2219-9365-2022-72-4-1

6. Xu, H., Yu, S., Jin, S., Sun, R., Chen, G., & Sun, L. (2024). Enhancing robustness in asynchronous feature tracking for event cameras through fusing frame streams. Complex & Intelligent Systems, 10(2), 6885–6899. https://doi.org/10.1007/s40747-024-01513-0

7. Shapiro, M., Preguiça, N., Baquero, C., & Zawirski, M. (2011). A comprehensive study of Convergent and Commutative Replicated Data Types (CRDTs). INRIA Research Report No. 7506. https://dsf.berkeley.edu/cs286/papers/crdt-tr2011.pdf

8. Overeem, M., Spoor, M., & Jansen, S. (2021). An empirical characterization of event-sourced systems and their schema evolution – Lessons from industry. Journal of Systems and Software, 178, 110970. https://doi.org/10.1016/j.jss.2021.110970

9. W3C. (2025). Web Cryptography Level 2. Standard. https://www.w3.org/TR/webcrypto-2/

10. Dworkin, M. (2007). Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D. https://doi.org/10.6028/NIST.SP.800-38D

11. Krawczyk, H., & Eronen, P. (2010). HMAC-based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869. IETF. https://www.rfc-editor.org/rfc/rfc5869

12. Verkhovna Rada of Ukraine. (2010). Law of Ukraine "On Personal Data Protection". https://zakon.rada.gov.ua/laws/show/2297-17

13. Schneier, B., & Kelsey, J. (1999). Secure Audit Logs to Support Computer Forensics. ACM Transactions on Information and System Security, 2(2), 159–176. https://dl.acm.org/doi/pdf/10.1145/317087.317089

Надійшла до редакції: 11.10.25
Прийнята до друку: 17.03.26
Опубліковано: 30.03.26