DOI 10.34185/1562-9945-4-165-2026-05
УДК: 004.75:004.056:004.415

А.М. Гриценко, А.А. Азарян, С.А. Рубан

ПРОЕКТУВАННЯ БЕЗПЕЧНОЇ АРХІТЕКТУРИ РОЗПОДІЛЕНОЇ СИСТЕМИ ЗБОРУ ТА НАКОПИЧЕННЯ ПРОМИСЛОВИХ ДАНИХ НА ПЛАТФОРМІ JAVA

Анотація. У статті розглянуто проблему забезпечення цілісності та безпеки технологічних даних у гірничо-видобувній галузі. Метою є проєктування захищеної розподіленої Java-архітектури для збору, збереження та обробки даних. Використано методи системного аналізу, моделювання загроз і архітектурного проєктування. Запропоновано мікросервісну архітектуру з принципами Security-by-Design, mTLS, RBAC та подієвою взаємодією. Результати демонструють підвищення відмовостійкості завдяки гібридному зберіганню з реплікацією та синхронізацією даних, що забезпечує усунення єдиних точок відмови та інтеграцію в сучасну IT-інфраструктуру.
Ключові слова: мікросервісна архітектура, Security-by-Design, розподілені системи, захист даних, відмовостійкість, mTLS, RBAC, подієво-орієнтована архітектура, синхронізація даних.

Постановка проблеми. Забезпечення цілісності та доступності технологічних даних є критично важливим для систем автоматизації гірничо-видобувної галузі України. Програмно-апаратний комплекс НАКС-ПК на базі Delphi 7 тривалий час забезпечував оперативний контроль якості руди гамма-гамма методом, однак експлуатація виявила низку архітектурних вразливостей, що створюють ризики для збору, збереження та доступу до даних.

По-перше, існують проблеми безпеки збору даних у середовищі з радіоактивними джерелами (Am-241). Оскільки система належить до радіоізотопних приладів, програмні збої або несанкціоноване втручання можуть призвести не лише до спотворення вимірювань, а й до втрати контролю над джерелом випромінювання. Використання Modbus RTU/TCP без шифрування та автентифікації робить можливими атаки типу «man-in-the-middle», що дозволяє підмінювати дані інтенсивності та навантаження конвеєра, впливаючи на визначення вмісту заліза та роботу обладнання.

По-друге, актуальною є проблема ненадійності накопичення даних і відсутності відмовостійкості. Використання локальних СУБД (Firebird або SQLite) на одному сервері створює «єдину точку відмови». У разі збоїв обладнання, мережі чи аварій у диспетчерській можливе повне безповоротне втрачання оперативних даних за значні періоди. Це є критичним, оскільки похибки у визначенні вмісту заліза можуть призводити до значних фінансових втрат або штрафів, а відсутність резервного відновлення посилює ризики.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

По-третє, система не відповідає сучасним вимогам щодо керування доступом і аудиту. Десктоп-додатки мають обмежені можливості ідентифікації користувачів, не підтримують рольову модель доступу (RBAC), не забезпечують централізованого логування та захисту критичних налаштувань. Це створює загрозу несанкціонованих змін і внутрішнього саботажу. Додатково, відсутність кросплатформенності обмежує використання сучасних операційних систем і ускладнює інтеграцію в сучасну IT-інфраструктуру підприємства.

Отже, існує потреба у розробці нової програмної архітектури на базі Java з використанням принципів Security-by-Design, яка забезпечить захищений збір даних, відмовостійке збереження в розподілених базах та надійний контроль доступу для користувачів.

Аналіз останніх досліджень і публікацій. Сучасна наукова спільнота визначає Security-by-Design (SbD) не як додаткову опцію, а як фундаментальний архітектурний принцип, де безпека закладається на етапі проектування інтерфейсів та моделей авторизації. Згідно з сучасними підходами до програмної інженерії, архітектура системи повинна враховувати нефункціональні вимоги, зокрема безпеку, ще на ранніх етапах розробки [2].

Розвиток мікросервісних архітектур значно вплинув на підходи до побудови розподілених систем, забезпечивши гнучкість, масштабованість та ізоляцію компонентів [3]. У роботах останніх років підкреслюється важливість використання архітектурних патернів для мікросервісів, що дозволяє систематизувати процес проєктування складних систем [4].

Питання обробки великих обсягів даних у розподілених системах детально розглянуто у фундаментальних дослідженнях, де визначено основні принципи побудови надійних і масштабованих систем [5]. Зокрема, важливу роль відіграє використання подієво-орієнтованих підходів та брокерів повідомлень, таких як Apache Kafka [6], що отримали подальший розвиток у сучасних потокових платформах [7].

У контексті промислових систем значна увага приділяється безпеці Industrial IoT, де дослідження демонструють зростання кількості кіберзагроз та необхідність комплексного підходу до їх нейтралізації [8]. Додатково питання безпеки та конфіденційності у виробничих системах розглядаються у контексті концепції smart manufacturing [9].

Основним стандартом для проектування безпечних промислових систем є IEC 62443, який визначає принципи сегментації та рівні безпеки [10; 11]. У свою чергу, сучасні криптографічні протоколи, такі як TLS 1.3 [12] та механізми авторизації OAuth 2.0 [13], забезпечують захист каналів зв’язку та контроль доступу. Додатково рекомендації NIST щодо цифрової ідентифікації [14] визначають вимоги до управління автентифікацією в розподілених системах.

Питання організації безперервної доставки програмного забезпечення та автоматизації процесів розгортання також є важливими для сучасних систем [15]. У контексті синхронізації даних у розподілених середовищах використовуються спеціалізовані інструменти, зокрема SymmetricDS [16].

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Окрему увагу слід приділити прикладним дослідженням у галузі контролю якості руди, де доведено ефективність використання гамма-випромінювання для оперативного визначення характеристик сировини [1].

Мета дослідження. Метою статті є проектування безпечної розподіленої архітектури системи контролю якості руди на платформі Java, яка за рахунок використання мікросервісного підходу, протоколів взаємної автентифікації (mTLS) та механізму гібридного дублювання баз даних забезпечує захищений збір, надійне накопичення та регламентований доступ до технологічної інформації.

Викладення основного матеріалу дослідження. Проектування архітектури нової системи базується на парадигмі Security-by-Design, що передбачає інтеграцію механізмів захисту як невід'ємних компонентів програмної структури, а не як зовнішніх надбудов. Такий підхід науково обґрунтований тим, що усунення архітектурних вразливостей на етапі дизайну є на порядок економічно ефективнішим порівняно з виправленням дефектів у вже розгорнутій системі.

Для досягнення цільового рівня безпеки SL-3 згідно з IEC 62443, пропонується декомпозиція системи на ізольовані зони безпеки, сполучені суворо регламентованими каналами, функціональна схема якої наведена на рис. 1:

- зона A: Містить детектори розсіяного гамма-випромінювання, детектори магнітної сприйнятливості та контролери, що забезпечують ретрансляцію сигналів та взаємодіють з конвеєрними вагами. Це зона найвищої фізичної критичності. Доступ до неї обмежений лише промисловим сегментом мережі, та зовні не доступний.
- зона B: Рівень Java-мікросервісів, розгорнутих у Docker-контейнерах. Тут реалізовано алгоритми взаємодії з віддаленими пристроями, логіку розрахунків вмісту корисного компонента та локальне кешування даних у PostgreSQL. Ця зона виступає «довіреним кордоном» (Trust Boundary), де відбувається фільтрація сирих даних.
- зона C (Enterprise Access Zone): Рівень хмарних або центральних серверів підприємства та клієнтських інтерфейсів [11].

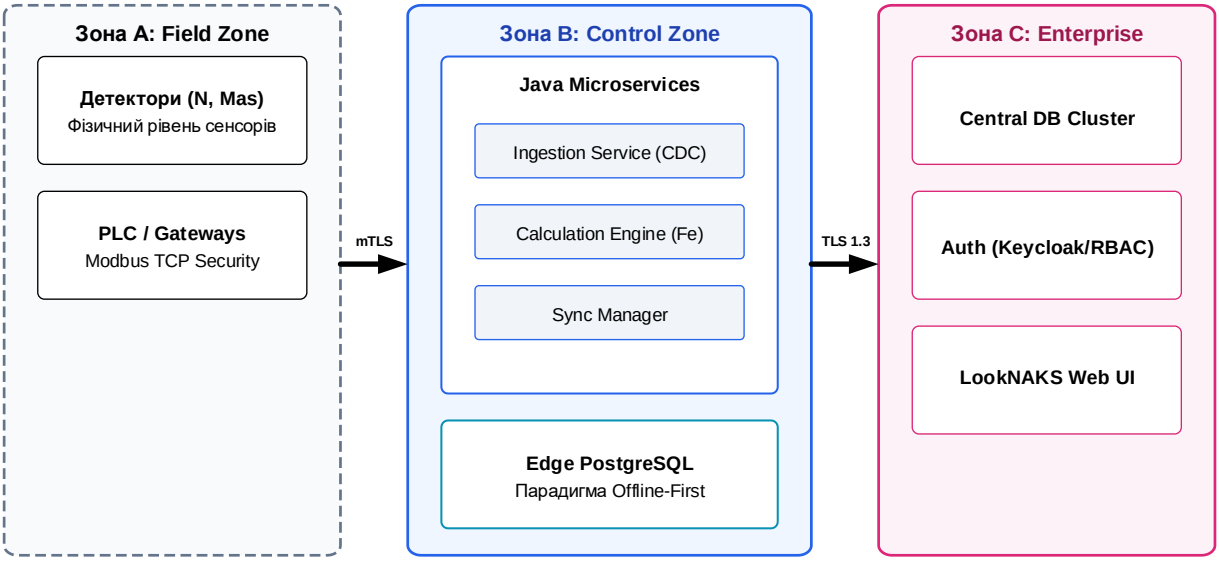

Рисунок 1 - Узагальнена функціональна схема безпечної архітектури

Вибір такої структури дозволяє локалізувати потенційні атаки: компрометація одного мікросервісу в Зоні В не дає прямого доступу до Зони А завдяки використанню Secure Conduits на базі TLS 1.3 [12].

Подальший розвиток запропонованої архітектури передбачає впровадження концепції Zero Trust, відповідно до якої жоден компонент системи не вважається довіреним за замовчуванням, навіть якщо він знаходиться всередині корпоративної мережі. Усі запити між мікросервісами проходять обов'язкову автентифікацію та авторизацію, що реалізується через централізований API Gateway. Такий підхід дозволяє мінімізувати ризики несанкціонованого доступу та забезпечує контроль усіх потоків даних у системі. Взаємодія компонентів базується на використанні протоколів OAuth 2.1 та OpenID Connect, що забезпечують стандартизований механізм управління ідентифікацією користувачів і сервісів [13].

Важливим елементом архітектури є формалізація моделі загроз, яка дозволяє обґрунтувати вибір механізмів захисту. У рамках дослідження виділено основні класи загроз: мережеві атаки (MitM, replay), атаки на рівні прикладного програмного забезпечення, внутрішні загрози та фізичні втручання. Для кожного з них визначено відповідні контрзаходи, що інтегруються безпосередньо у систему. Зокрема, для запобігання replay-атакам використовується комбінація часових міток та одноразових чисел (nonce), що перевіряються на стороні приймача.

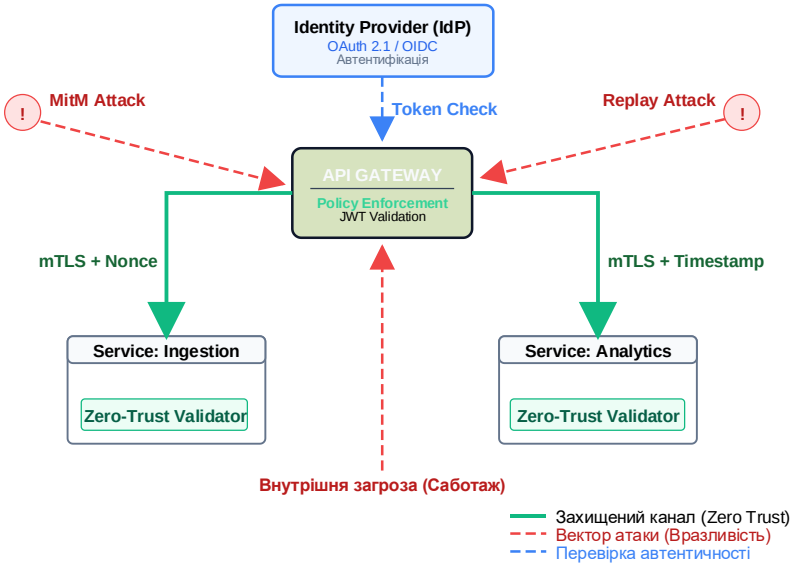

Рисунок 2 – Модель загроз та механізмів захисту

Забезпечення криптографічного захисту даних реалізується на двох рівнях: транспортному та рівні повідомлень. На транспортному рівні використовується протокол TLS 1.3 із взаємною автентифікацією (mTLS), що дозволяє гарантувати конфіденційність та цілісність переданих даних. На рівні повідомлень реалізовано цифровий підпис пакетів, який формується шляхом обчислення хешу (SHA-256) та його підписання за допомогою алгоритму ECDSA. Такий підхід дозволяє забезпечити захист навіть у випадку компрометації каналу зв'язку.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Значна увага приділяється організації обміну даними між компонентами системи. Для цього використовується подієво-орієнтована архітектура (Event-Driven Architecture) з використанням брокера повідомлень. Це дозволяє реалізувати асинхронну взаємодію між сервісами (рис. 3), зменшити залежності між ними та підвищити масштабованість системи.

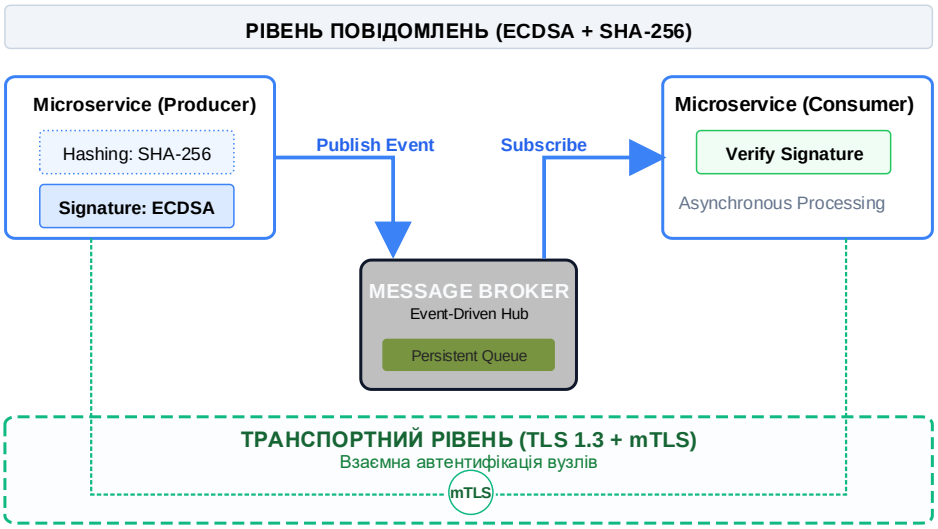

Рисунок 3 – Архітектура обміну даними на основі подій

Одним із ключових аспектів є забезпечення надійності накопичення даних. Для цього запропоновано гібридну модель зберігання, що поєднує локальні бази даних із віддаленими репліками (рис. 4). У кожному вузлі системи функціонує локальна база PostgreSQL, яка працює у режимі offline-first. У разі втрати мережевого з'єднання система продовжує накопичувати дані локально, а після відновлення зв'язку виконується синхронізація.

Механізм синхронізації базується на поєднанні підходів Event Sourcing та Change Data Capture. Усі зміни фіксуються у вигляді подій, що записуються у журнал, після чого передаються до інших вузлів системи у вигляді інкрементальних оновлень.

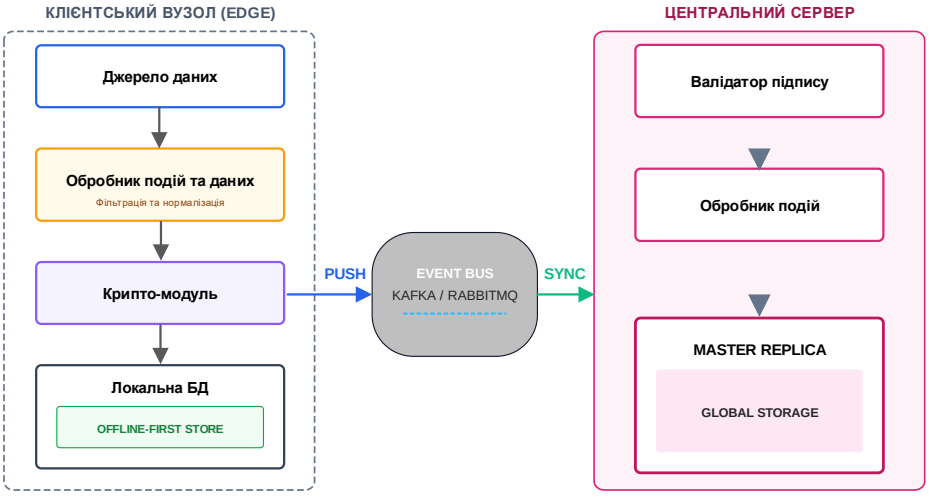

Рисунок 4 – Схема синхронізації даних

Для забезпечення відмовостійкості використовується кластерна архітектура баз даних із підтримкою реплікації та автоматичного перемикання ролей (див. рис. 5). У разі відмови основного вузла одна з реплік автоматично бере на себе його функції, що дозволяє забезпечити безперервність роботи системи.

Рисунок 5 – Архітектура кластеру баз даних

З метою підвищення продуктивності система використовує реактивну модель обробки даних, що базується на неблокуючих потоках. Це дозволяє ефективно обробляти великі обсяги телеметрії у режимі реального часу та забезпечує стабільну роботу навіть при високому навантаженні.

Додатково в архітектуру інтегровано систему централізованого логування та моніторингу, яка дозволяє здійснювати аудит дій користувачів і виявляти потенційні інциденти безпеки. Використання сучасних інструментів аналізу логів забезпечує можливість оперативного реагування на загрози.

Перспективним напрямом розвитку є інтеграція методів машинного навчання для виявлення аномалій у даних. Аналіз потоків телеметрії дозволяє автоматично визначати нетипові значення, що можуть свідчити про несправності обладнання або спроби втручання.

Проведене моделювання показало, що запропонована архітектура забезпечує високу продуктивність та відмовостійкість. Затримка обробки даних не перевищує допустимих значень, а використання реплікації дозволяє уникнути втрати інформації навіть у разі часткових відмов системи.

Висновки. Розроблено науково обґрунтовану архітектуру розподіленої системи збору промислових даних, що базується на принципах Security-by-Design і відповідає стандарту IEC 62443 (рівень SL-3). Запропонований підхід усуває критичні недоліки монолітних систем: єдині точки відмови, низьку масштабованість та вразливість каналів зв'язку.

Наукова новизна полягає у формалізації побудови безпечної мікросервісної архітектури, яка поєднує зональну сегментацію, mTLS-автентифікацію, багаторівневе шифрування та гібридне зберігання offline-first. Внеском у розвиток відмовостійких сис-

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

тем є інтеграція патернів Event Sourcing та Change Data Capture для синхронізації даних у середовищах із нестабільним зв'язком.

Практичне значення результатів підтверджено можливістю модернізації комплексів типу НАКС-ПК. Архітектура гарантує достовірність збору, безпеку передачі та автоматичне відновлення даних. Мікросервісний підхід і контейнеризація забезпечують гнучке масштабування та легку інтеграцію в IT-інфраструктуру підприємства.

Рівень безпеки підвищено завдяки концепції Zero Trust, управлінню доступом (RBAC) та системному аудиту, що мінімізує зовнішні та внутрішні загрози. Моделювання підтвердило високу продуктивність системи та стабільність її роботи за умов часткових відмов інфраструктури.

Перспективи досліджень передбачають інтеграцію алгоритмів машинного навчання для предиктивного аналізу якості руди та впровадження edge computing для підвищення автономності системи. Розроблена архітектура відповідає вимогам Industry 4.0 і рекомендована до впровадження на підприємствах гірничо-видобувної галузі.

ЛІТЕРАТУРА

1. Оперативний контроль якості руд чорних металів з використанням гамма-випромінювання / А. А. Азарян та ін. // Гірничий вісник. 2022. № 110. С. 13–22. DOI: 10.31721/2306-5435-2022-1-110-13-22.

2. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. Sebastopol : O'Reilly Media, 2020. 432 p.

3. Microservices: yesterday, today, and tomorrow / N. Dragoni et al. Cham : Springer, 2021. 281 p.

4. Taibi D., Lenarduzzi V., Pahl C. Architectural patterns for microservices: a systematic mapping study. [B. m.] : Springer, 2020. 250 p.

5. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol : O'Reilly Media, 2017. 616 p.

6. Kreps J., Narkhede N., Rao J. Kafka: a distributed messaging system for log processing. NetDB, 2011.

7. Carbone P. et al. Apache Kafka: a distributed streaming platform. [B. m.] : [b. n.], 2020.

8. Chen Y., Meng W., Kwok L. F. Industrial IoT security: a survey. [B. m.] : [b. n.], 2022.

9. Zhang Y., Deng R. H. Smart manufacturing security. [B. m.] : Springer, 2021.

10. Industrial communication networks – Network and system security – Part 3-3: System security requirements and security levels : IEC 62443-3-3:2013. Geneva : IEC, 2013.

11. Security for industrial automation and control systems – Part 4-2: Technical security requirements for IACS components : IEC 62443-4-2:2019. Geneva : IEC, 2019.

12. Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3 : RFC 8446. 2018. URL: https://datatracker.ietf.org/doc/html/rfc8446 (дата звернення: 05.04.2026).

13. Hardt D. The OAuth 2.0 Authorization Framework : RFC 6749. 2012. URL: https://datatracker.ietf.org/doc/html/rfc6749 (дата звернення: 07.04.2026).

14. Grassi P. A. NIST Digital Identity Guidelines : Special Publication 800-63-3. Gaithersburg : NIST, 2020.

15. Humble J., Farley D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. [B. m.]: Addison-Wesley, 2021.

16. SymmetricDS User Guide. 2024. URL: https://www.symmetricds.org/doc/ (дата звернення: 05.04.2026).

REFERENCES

1. Azaryan, A.A., Azaryan, V.A., Morkun, V.S., Hrytsenko, A.M., & Trachuk, A.A. (2022). Operatyvnyi kontrol yakosti rud chornykh metaliv z vykorystanniam hamma-vyprominiuvannia [Operational quality control of ferrous metal ores using gamma radiation]. Hirnychyi Visnyk, 110, 13–22. https://doi.org/10.31721/2306-5435-2022-1-110-13-22
2. Richards, M., & Ford, N. (2020). Fundamentals of software architecture: An engineering approach. O'Reilly Media.
3. Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2021). Microservices: Yesterday, today, and tomorrow. Springer.
4. Taibi, D., Lenarduzzi, V., & Pahl, C. (2020). Architectural patterns for microservices. Springer.
5. Kleppmann, M. (2017). Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media.
6. Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. NetDB.
7. Carbone, P., et al. (2020). Apache Kafka: A distributed streaming platform.
8. Chen, Y., Meng, W., & Kwok, L. F. (2022). Industrial IoT security: A survey. Journal of Cybersecurity and Privacy (або інше джерело, якщо це стаття).
9. Zhang, Y., & Deng, R. H. (2021). Smart manufacturing security. Springer.
10. International Electrotechnical Commission. (2013). Industrial communication networks – Network and system security – Part 3-3: System security requirements and security levels (IEC 62443-3-3:2013).
11. International Electrotechnical Commission. (2019). Security for industrial automation and control systems – Part 4-2: Technical security requirements for IACS components (IEC 62443-4-2:2019).
12. Rescorla, E. (2018). The Transport Layer Security (TLS) protocol version 1.3 (RFC 8446). IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc8446
13. Hardt, D. (2012). The OAuth 2.0 authorization framework (RFC 6749). IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc6749
14. Grassi, P. A. (2020). NIST digital identity guidelines (Special Publication 800-63-3). National Institute of Standards and Technology.
15. Humble, J., & Farley, D. (2021). Continuous delivery: Reliable software releases through build, test, and deployment automation. Addison-Wesley.
16. JumpMind. (2024). SymmetricDS user guide. https://www.symmetricds.org/doc/

Received 17.04.2026.
Accepted 22.04.2026.
Published 30.04.2026

Design of a secure architecture for a distributed industrial data collection and storage system on the java platform

Abstract. In the context of industrial digitalization, ensuring the integrity, availability, and security of technological data in the mining industry has become highly relevant. Recent studies indicate that the Security-by-Design concept is a fundamental principle for building

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

modern information systems, while microservices and event-driven architectures enhance scalability and flexibility. Particular attention is also paid to Industrial IoT security, IEC 62443 standards, cryptographic protocols such as TLS, and modern authorization mechanisms.

The aim of this study is to design a secure distributed architecture for industrial data collection and storage on the Java platform, ensuring reliable data acquisition, fault-tolerant storage, and controlled access.

The research methodology includes systems analysis, threat modeling, architectural design, and analysis of modern distributed system approaches. A microservices-based architecture is proposed, implementing Security-by-Design principles, zone segmentation in accordance with IEC 62443, and the Zero Trust concept. Secure communication is achieved using TLS 1.3 with mutual authentication (mTLS), while access control is implemented via RBAC. Data exchange is organized using an event-driven architecture. To ensure data reliability, a hybrid storage model (offline-first) with replication and synchronization based on Event Sourcing and Change Data Capture is applied.

The results demonstrate improved fault tolerance, elimination of single points of failure, and continuous system operation under partial failures. The proposed architecture enhances data transmission security, ensures strict access control, and enables comprehensive auditing. The conclusions confirm the effectiveness of the microservices approach and modern security mechanisms for upgrading industrial systems and integrating them into enterprise IT infrastructures in accordance with Industry 4.0 requirements.

Keywords: microservices architecture, Security-by-Design, distributed systems, data protection, fault tolerance, mTLS, RBAC, event-driven architecture, data synchronization.

Гриценко Андрій Миколайович – к.т.н., доцент кафедри можелювання та програмного забезпечення, Криворізький національний університет, Кривий Ріг, Україна.
ORCID: https://orcid.org/0000-0001-8665-853X
Азарян Альберт Арамаїсович – д-р техн. наук, професор, головний науковий співробітник, Криворізький національний університет, Кривий Ріг, Україна.
ORCID: https://orcid.org/0000-0002-1381-579X
Рубан Сергій Анатолійович – к.т.н, доцент, завідувач кафедри автоматизації, комп'ютерних наук і технологій, Криворізький національний університет, Україна.
ORCID: https://orcid.org/0000-0002-4495-6667

Hrytsenko Andrii – PhD in Technical Sciences, Associate Professor at the Department of Modeling and Software Engineering, Kryvyi Rih National University, Kryvyi Rih, Ukraine.
ORCID: https://orcid.org/0000-0001-8665-853X
Azaryan Albert – Doctor of Technical Sciences, Professor, Chief Researcher, Kryvyi Rih National University, Kryvyi Rih, Ukraine.
ORCID: https://orcid.org/0000-0002-1381-579X
Ruban Serhii - PhD, Docent, Head of the Department of Automation, Computer Sciences and Technologies
ORCID: https://orcid.org/0000-0002-4495-6667