

DATA MESH – ІННОВАЦІЙНА СИСТЕМА ОРГАНІЗАЦІЇ ДАНИХ ДЛЯ РОЗПОДІЛЕНИХ ПРОЄКТНИХ КОМАНД

Роман Васьків, Наталія Веретеннікова

Національний університет «Львівська політехніка»,
кафедра інформаційних систем та мереж, Львів, Україна
E-mail: roman.i.vaskiv@lpnu.ua, ORCID: 0000-0002-8549-5035
E-mail: nataliia.v.veretennikova@lpnu.ua, ORCID: 0000-0001-9564-4084

© Васьків Р.І., Веретеннікова Н.В., 2025

У статті представлено Data Mesh як інноваційну систему організації даних, розроблену для підтримки ефективності розподілених проєктних команд у динамічному цифровому середовищі. Показано обмеження централізованих систем організації даних, таких як Data Warehouse та Data Lake, що втрачають адаптивність і масштабованість у багатодомених структурах. Метою дослідження є наукове обґрунтування принципів Data Mesh та визначення її потенціалу.

Data Mesh базується на чотирьох засадах: доменна відповідальність за дані, дані як продукт, інформаційно-технологічна платформа самообслуговування та федеративне управління. Система організації даних Data Mesh забезпечує формалізацію взаємодії через контракти на дані, інтеграцію механізмів спостережуваності, трасування походження та інструментів автоматизації. Застосування продемонстровано на прикладі логістичної компанії, де доменні команди створюють і підтримують власні дата-продукти, а узгодженість забезпечується через стандарти та централізовані політики.

Результати підтверджують, що Data Mesh формує основу для побудови стійких, масштабованих і гнучких систем організації даних, які поєднують автономію доменів із глобальною координацією. Це створює умови для підвищення продуктивності, прозорості та швидкості прийняття рішень у розподіленому середовищі, а також відкриває перспективи подальших досліджень у напрямках управління якістю даних та оптимізації проєктних процесів у розподілених ІТ-командах.

Ключові слова – Data Mesh, розподілені проєктні команди, система організації даних, архітектура даних.

Постановка проблеми

У сучасних умовах глобалізації, розподіленого характеру роботи та активного переходу до цифрових моделей ведення бізнесу, ІТ-компанії все частіше стикаються з необхідністю масштабування управління даними в умовах високої динаміки, географічної розпорошеності команд і зростання кількості джерел даних. Проблематика централізованих моделей управління даними, зокрема архітектур типу Data Lake та Data Warehouse, набуває особливої актуальності, хоча вони й забезпечують уніфікацію та централізований контроль, однак із часом демонструють зниження адаптивності, зростання залежності від вузькопрофільних команд платформної підтримки та обмежену масштабованість у контексті мультидомених організацій.

За оцінками, великі підприємства щороку накопичують і зберігають у десять разів більше даних, проте майже 90% з них залишаються невикористаними або задіяними лише частково. Наприклад, згідно з аналітичним звітом Big Data Analytics News (2024), щодня у світі генерується приблизно 2,5 квінтільйона байтів даних. Загальний обсяг даних, згенерованих у світі до 2023 року, становив 120 зеттабайтів, і очікується, що до кінця 2025 року цей показник сягне 181 зеттабайта. (Fortune Business Insights. 2024).

Централізовані підходи концентрують інженерну відповідальність і знання в межах невеликої групи спеціалістів, що призводить до організаційного перевантаження, затримок у реалізації запитів і неможливості оперативно відповідати на бізнес-виклики. Такий підхід суперечить сучасним принципам гнучкості, автономності команд і швидкого циклу доставки цінності, які є визначальними в умовах розподілених IT-команд і DevOps/DataOps-культур.

У відповідь на ці технічні та організаційні виклики формується концепція Data Mesh (сітка даних) – система організації управління даними, що інтегрує принципи доменно-орієнтованої архітектури, платформного підходу та трактування даних як продукту. Модель Data Mesh передбачає делегування повної відповідальності за життєвий цикл даних на рівень окремих доменів, що створює передумови для масштабованого управління даними в умовах децентралізації, високої динаміки процесів і цифрової трансформації організацій.

Аналіз останніх досліджень та публікацій

У сучасних умовах, коли цифрова трансформація охоплює дедалі більше організаційних систем, зростає потреба у таких підходах до управління даними, які одночасно враховують як технологічні, так і організаційно-структурні аспекти.

У зв'язку із стрімким зростання інформаційно-аналітичних потреб установ та організацій, групи фахівців в галузі аналітичного опрацювання даних є суттєво перевантаженими на всіх етапах життєвого циклу даних. Це в свою чергу потребує системного узгодження цих потреб з процесами формування та запровадження нових архітектур та організації даних (Дуда О., 2025). Традиційні централізовані системи організації даних (зокрема, Data Lake, Data Warehouse) дедалі частіше демонструють обмеження у масштабованості, гнучкості та здатності задовольняти потреби розподілених команд. У відповідь на ці виклики, система організації даних Data Mesh була запропонована Жамак Деггані як інноваційне рішення для управління даними у масштабах великих розподілених організацій (Dehghani, 2022).

Data Mesh – це децентралізований підхід до управління аналітичними даними, що базується на чотирьох основних принципах:

1. Доменно-орієнтоване управління даними (Domain-oriented data ownership and architecture).
2. Дані як продукт (Data as a product).
3. Платформа самообслуговування (Self-serve data platform).
4. Федеративне управління (Federated computational governance).

На відміну від централізованих рішень, де дані з різних джерел збираються у єдине сховище, Data Mesh передбачає делегування відповідальності за створення, обслуговування та обмін даними конкретним доменам. Кожен домен виступає як відповідальний власник дата-продуктів, а їх інтеоперабельність забезпечується стандартами метаданих, API, схемами та контрактами.

Відзначається, що одним із поширених способів подання архітектури інформаційної системи є ієрархічна структура, при цьому виділяються три базових типи архітектури інформаційних систем: інтегрована, розподілена та змішана. (Pasicznyk V., 2025).

Архітектура даних – це те, як різні типи даних і різні потреби в даних поєднуються, у цілісний та згуртований спосіб, що визначається як велика та складна архітектура і сфера досліджень, яка охоплює спосіб, у який дані використовуються, формуються та зберігаються. (Inmon W. H, 2019).

Ідея Data Mesh частково ґрунтується на архітектурі мікросервісів, що застосовується в інженерії програмного забезпечення (Ashraf A, 2023). Архітектура системи організації даних Data Mesh спрямована на подолання обмежень централізованих платформ та забезпечення масштабованості в умовах зростаючої складності бізнес-процесів (Dibouliya, A., 2023). Її структура базується на інтеграції чотирьох фундаментальних принципів, що визначають баланс між автономією доменних команд та глобальною узгодженістю організації (рис. 1).

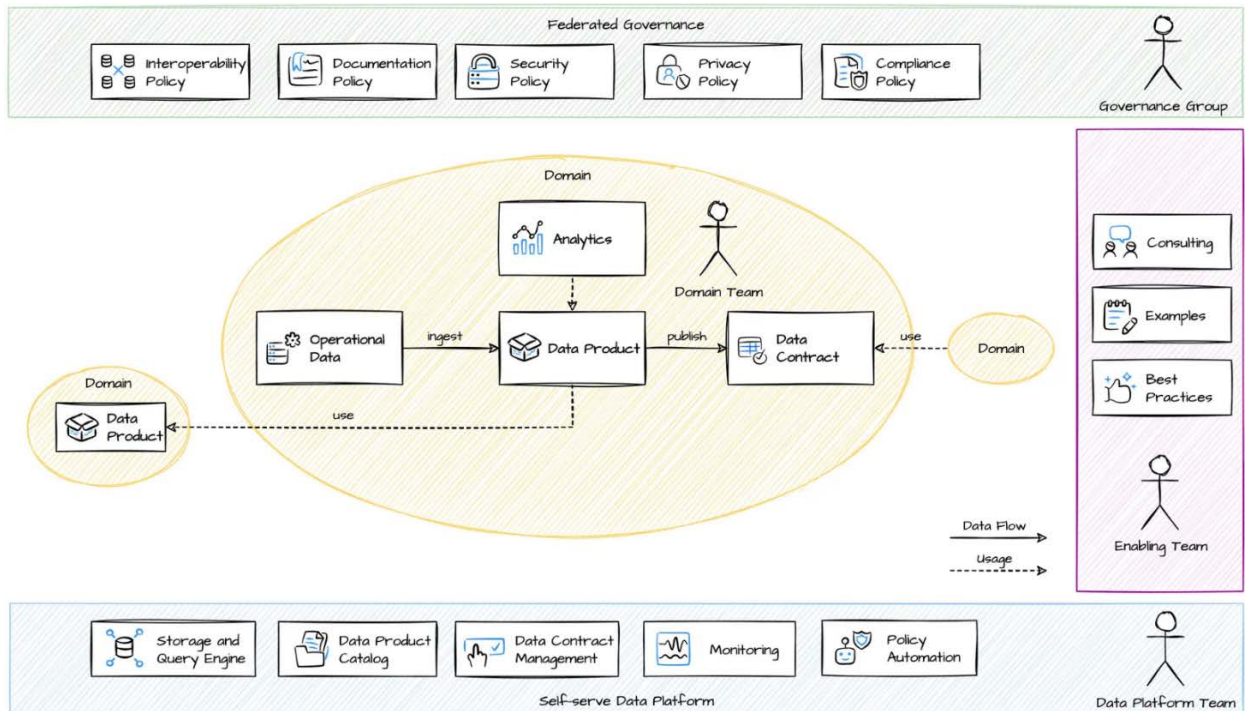


Рис. 1 Архітектура системи організації даних Data Mesh (Jochen Christ, 2025; Dehghani, Z. 2020)

По-перше, доменно-орієнтоване володіння даними (Domain Ownership) передбачає, що відповідальність за створення, якість і підтримку даних належить тим командам, які володіють предметним контекстом. Це забезпечує глибший зв'язок між даними та бізнес-процесами, усуваючи інформаційні розриви між технічними та функціональними підрозділами.

По-друге, дані як продукт (Data as a Product) формалізують підхід, у якому дані набувають статусу повноцінних бізнес-одиниць із власними метаданими, схемами, API, угодами про рівень обслуговування (SLA) та чітко визначеною цінністю для споживачів. Такий підхід унеможливорює неструктуроване використання даних і встановлює нові стандарти прозорості й відтворюваності.

По-третє, платформа самообслуговування (Self-Serve Data Platform) створює єдиний інструментальний шар, який забезпечує доменні команди усіма необхідними сервісами: реєстрами дата-продуктів, системами управління контрактами, моніторингом, механізмами lineage та автоматизованим дотриманням політик. Це дозволяє значно знизити технічні бар'єри для команд і гарантувати масштабованість екосистеми.

По-четверте, федеративне управління (Federated Governance) забезпечує баланс між локальною автономією та глобальними політиками. Воно встановлює єдині стандарти взаємодії, безпеки, документування, конфіденційності та відповідності нормативним вимогам, інтегруючи їх у процеси розробки та експлуатації даних.

У такій архітектурі ключовими структурними елементами виступають дата-продукти, дата-контракти, платформа самообслуговування та governance-рівень. Взаємодія між продюсерами та споживачами даних формалізується через дата-контракти, що задають умови використання, якість, частоту оновлень і комплаєнс. Платформа забезпечує технічну підтримку доменних команд, тоді як федеративне управління координує стандартизацію та контроль.

Таким чином, систему організації даних Data Mesh доцільно розглядати як багаторівневу структуровану систему, яка поєднує автономію доменів із узгодженою організаційною координацією. Це дозволяє досягти стійкого масштабування аналітичних екосистем, підвищити узгодженість управління даними та створити основу для довгострокової еволюції організацій у середовищах з високим рівнем динаміки та розподіленості.

Запропонований підхід відображає інтеграційний характер управління даними, де технічні рішення (контейнери, мікросервіси, інфраструктура як код) тісно пов'язані з організаційною структурою, у якій відповідальність, компетенції та цінність розподілені між командами.

Для кращого розуміння відмінностей між Data Mesh і традиційними підходами, наведемо порівняльну таблицю (таб. 1) за ключовими критеріями.

Таблиця 1

**Порівняння архітектур систем організації даних Data Warehouse,
Data Lake та Data Mesh за ключовими критеріями**

Критерій	Data Warehouse	Data Lake	Data Mesh
Власність на дані	Централізована, зосереджена в IT-відділі	Централізована, відповідальність зберігається на платформній команді	Децентралізована, доменна відповідальність за повний життєвий цикл даних
Масштабованість	Обмежена, масштабування є дорогим і технічно складним	Висока завдяки низькій вартості зберігання й підтримці різних форматів	Висока завдяки автономії доменних команд і відсутності «вузьких місць»
Виявлення даних	Ускладнене, оскільки дані інтегруються в монолітну модель	Може бути складним через відсутність суворої структури («data swamp» як ризик)	Спрощене завдяки стандартизованим контрактам, метаданам та каталогам
Гнучкість адаптації	Низька, орієнтована на структуровані дані й SQL-запити	Середня: підтримує різні типи даних, але потребує додаткових механізмів обробки	Висока завдяки доменній автономії, підтримці різних форматів і швидкій адаптації до змін бізнес-вимог
Управління	Централізоване IT, з жорстким контролем	Частково центра- лізоване, орієнтоване на зберігання, але слабкі механізми якості	Федеративне, з участю доменних команд і глобальних політик
Орієнтація на продукт	Відсутня, дані розглядаються як актив для звітності та BI	Відсутня, основний акцент на акумуляції великих обсягів	Чітко артикульована: дані розглядаються як продукт (Data as a Product) із SLA, схемами, API та документацією

Таким чином, Data Mesh усуває обмеження масштабованості централізованих архітектур завдяки модульному, доменно-орієнтованому проєктуванню. Особливого значення набуває зміна акценту з підходу «дані як актив» (Data as an Asset) на концепцію «дані як продукт» (Data as a Product), що стимулює команди до створення якісних, багаторазово використовуваних дата-продуктів із чітко визначеними угодами про рівень обслуговування (SLA) та належною науково-технічною документацією (Broda, E. 2024; Dehghani, Z. 2022).

Формулювання цілі статті

Мета статті полягає у науковому обґрунтуванні інноваційної системи організації даних Data Mesh, інтегрованої у контекст інформаційних технологій супроводу проєктних процесів у розподілених командах. Автори зосереджуються на подоланні обмежень централізованих систем організації даних (зокрема, Data Lake та Data Warehouse), які втрачають ефективність у середовищі багатодомених організацій із географічною та часовою розподіленістю.

Data Mesh у цьому дослідженні розглядається не лише як технологічне рішення, а як цілісна система організації даних із чітко визначеною структурою, яка забезпечує узгодженість між доменними командами, прозорість процесів прийняття рішень, контроль якості даних і масштабованість в умовах динамічних проектних середовищ. Такий підхід формує методологічну основу для побудови стійких систем управління даними, здатних підтримувати як автономію окремих доменів, так і глобальну координацію, що є ключовим фактором ефективного функціонування сучасних розподілених організацій.

Виклад основного матеріалу

Система організації даних Data Mesh інтегрує технічні компоненти з організаційними принципами управління даними. З одного боку, реалізація архітектури Mesh передбачає впровадження контейнеризованих середовищ, інструментів оркестрації потоків даних (наприклад, Airflow, Dagsster), систем управління схемами (Avro, Protobuf), використання формалізованих угод на дані (data contracts), а також створення інформаційно-технологічної платформи самообслуговування, що забезпечує доступність інструментів для розгортання, моніторингу й обслуговування дата-продуктів. З іншого боку, підхід Data Mesh передбачає формування міждисциплінарних доменних команд, здатних самостійно відповідати за повний життєвий цикл своїх продуктів даних — від проектування до експлуатації та еволюції.

Інтеграція організаційних та технологічних механізмів у системі організації даних Data Mesh реалізується через такі структуро-утворювальні аспекти, зазначені нижче:

- Трансформація ролей – поява власника продукту даних (Data Product Owner) як нової відповідальної особи за якість, доступність і еволюцію продуктів даних у межах домену.
- Перерозподіл відповідальності – зменшення навантаження на центральну ІТ-команду та перехід до моделі співпраці між доменами.
- Формалізація контрактних відносин – запровадження чітких правил взаємодії між командами через контракти на дані (Data Contracts) та стандартизовані програмні інтерфейси (API) (DataContract, 2025).

Ефективне впровадження Data Mesh потребує дотримання низки ключових організаційно-технологічних умов, зокрема:

- Зрілості організаційної культури, що передбачає готовність до делегування повноважень, міждоменної взаємодії та забезпечення інженерної автономії команд.
- Наявності стратегічного платформеного мислення, за якого централізована команда фокусується на розвитку інформаційно-технологічної платформи (self-serve data platform), а не на ручному опрацюванні запитів.
- Налагодження системи спостережуваності метаданих і якості даних, що реалізується шляхом використання інструментів контролю спостережуваності даних (Data Observability), каталогів даних (Data Catalogs) та відстеження походження (Lineage Tracking).

Також важливо розглядати Data Mesh не як повну заміну Data Lake/ DWH, а як надбудову, яка покликана усунути їхні обмеження у динамічному середовищі, де швидкість прийняття рішень і локальний контекст мають критичне значення (Pongrech, W. A., 2023).

Упродовж останніх трьох десятиліть системи організації даних пройшли еволюцію від централізованих архітектур до більш гнучких, децентралізованих моделей, які краще адаптовані до потреб розподілених команд.

У традиційних архітектурах (зокрема, у сховищах даних – Data Warehouse), дані зберігалися централізовано, а доступ до них мали переважно аналітики та спеціалізовані ІТ-команди. Однак зі зростанням обсягів, різноманітності, швидкості даних і поширенням географічно розподілених команд постала необхідність у впровадженні нових систем організації даних, таких як Data Mesh та Data Fabric (Jonkman C., 2023).

Таблиця 2

Еволюція систем організації даних

Період	Архітектура	Характеристика	Ключові інструменти	Основні споживачі даних
1990–2005	Data Warehouse	Централізована, контроль з боку IT; запити через SQL, звіти	Oracle, Teradata, Cognos, BusinessObjects	IT, аналітики
2005–2012	Big Data	Розподілена, масштабована; низький рівень доступності для бізнесу	Hadoop, Hive, Pig, HDFS	Data engineers, науковці з даних
2012–2018	Cloud Data Platform	Масова доступність, масштабування, інтеграція BI	Snowflake, BigQuery, Redshift, Power BI	Бізнес-користувачі, аналітики
2018–2022	Data Lakehouse	Поєднання lake + warehouse; гнучкість і структура	Delta Lake, Apache Iceberg, MLflow	Кросфункціональні команди
2020 – теперішній	Data Mesh / Data Fabric	Децентралізація, орієнтація на домени / інтелектуальна автоматизація	Data catalogs, federated governance, AI tooling	Доменні команди, продукти з даних

Таблиця 3

Порівняльна характеристика систем організації даних за ключовими критеріями

Критерій	Data Warehouse / Data Lake	Data Fabric	Data Mesh
Архітектура	Централізована	Метадані-керована, автоматизована	Доменна, децентралізована
Власність	IT-відділ	Центральна, з розширеним доступом	Доменні команди
Масштабованість	Обмежена	Висока через автоматизацію	Висока через автономію доменів
Виявлення даних	Через ETL/SQL	AI-навігована каталогізація	Через self-service API, каталоги data products
Гнучкість адаптації	Низька	Середня	Висока
Управління (governance)	Централізоване	AI/ML-кероване	Федеративне, політики як код
Інтероперабельність	Складна інтеграція з legacy	Інтеграція через абстракцію	Сумісність через контракти, API, доменні угоди

Джерела: Dehghani (2022).

У підсумку, інноваційна система організації даних Data Mesh формує нову методологію управління інформаційними потоками у великих організаціях. Її значущість особливо актуалізується у контексті супроводу проєктних процесів у розподіленому середовищі, де географічна, функціональна та часова фрагментація є сталою характеристикою сучасного цифрового бізнесу.

Data Mesh дозволяє таким командам працювати незалежно, узгоджуючи свої дії через формальні контракти та політики. Цей підхід знижує навантаження на центральну архітектурну команду, розподіляє відповідальність за дані, а також підтримує узгодженість між доменними командами (Domain Teams), незалежно від їхньої локації чи часових поясів. У таких умовах підвищується швидкість прийняття рішень, зростає продуктивність, а ризики через затримки або неякісні дані – суттєво знижуються.

На практиці це означає, що кожна доменна команда бере участь у плануванні, постачанні та еволюції інформаційних ресурсів (даних), за які вона несе відповідальність. У межах ітеративного процесу розробки (спринтів), створення, оновлення та оцінювання дата-продуктів інтегруються у перелік завдань (беклог), а метрики якості – у визначення критеріїв прийнятності (definition of done) (Vestues, K, 2022). Інформаційно технологічна платформа, у свою чергу, забезпечує наскрізну спостережуваність (observability), реєстрацію походження даних (data lineage), управління версіями та автоматизовані сповіщення (alerting), а також централізовану підтримку стандартів авторизованого доступу.

Інноваційна система організації даних Data Mesh охоплює не лише технічні інструменти, але й змінює підходи до мислення про дані у великих організаціях. Її практична значущість особливо проявляється у супроводі проєктних процесів, що реалізуються розподіленими командами – зокрема в умовах цифрової економіки, де географічна, функціональна та часова фрагментація стали типовим викликом.

Інноваційну систему організації даних Data Mesh доцільно представити як багаторівневу організаційно-технологічну структуру, у якій кожен компонент виконує чітко визначену функцію в управлінні даними у розподіленому середовищі. Основними учасниками цієї структури виступають доменні команди, відповідальні за повний життєвий цикл дата-продуктів – від моделювання та публікації до контролю якості, оновлень і дотримання угод про рівень обслуговування (SLA).

Продукти даних розглядаються як самодостатні одиниці бізнес-цінності з формалізованими метаданими, керованою еволюцією схем, стандартизованими програмними інтерфейсами (API) та прозорими правилами доступу.

Взаємодія між командами-виробниками й командами-споживачами даних реалізується через контракти на дані, які визначають обов'язки сторін, структуру передавання, періодичність оновлень, вимоги до конфіденційності та інші параметри узгодженості.

Функціонування системи організації даних Data Mesh спирається на інформаційно-технологічну платформу, яка забезпечує інструментальну підтримку доменних команд: засоби CI/CD, реєстри схем, механізми спостережуваності (observability) та трасування походження даних (data lineage). Над усіма цими процесами функціонує рівень федеративного управління, що забезпечує централізоване формування політик, глосаріїв, стандартів і автоматизованих перевірок відповідності, вбудованих у процеси розгортання.

Архітектура системи організації даних Data Mesh трактується як впорядкована структура, що охоплює множину ключових компонентів (доменів, дата-продуктів, акторів, контрактів, схем, політик тощо) та відношень між ними, необхідних для забезпечення узгодженого функціонування у децентралізованому середовищі. Формалізація цієї архітектури базується на системі множин і відношень, яка математично моделює основні сутності та логіку їх взаємодії. Формальне ядро визначається множинами: $D, P, U, C, S, M_t, R_v, E, K$ (таб. 4), що відповідно позначають домени, дата-продукти, акторів, контракти, схеми, метадані, правила управління, перевірки та платформні компоненти.

Таблиця 4

Формальне ядро Data Mesh: визначення базових множин

Назва множини	Опис множини
D	Множина доменів
P	Множина дата-продуктів
U	Множина акторів/команд (продюсери, споживачі, аналітики тощо)
C	Множина контрактів на дані
S	Множина схем
M_t	Множина записів метаданих
R_v	Множина правил та політик управління
E	Множина перевірок і тестів
K	Множина платформних компонентів

Для формального опису кожен дата-продукт у множині $p \in P$ задається кортежем, що відображає його домену приналежність, схему, метадані, показники якості, часові характеристики та відповідального власника:

$$p = \langle d(p), s(p), m(p), q(p), t(p), o(p) \rangle, \quad (1)$$

де $d(p) \in D$ – домен-власник; $s(p) \in P$ – схема; $m(p) \in M_t$ – метадані (версії, SLA, опис); $q(p) \in [0, 1]^r$ – вектор метрик якості (повнота, точність, своєчасність тощо); $t(p) \in R_{\geq 0}$ – часовий атрибут (мітка оновлення, латентність, актуальність); $o(p) \in U$ – власник продукту (команда або актор).

Відношення доступу:

$$acc \subseteq U \times P \quad (2)$$

$$(u, p) \in acc \Rightarrow u \text{ має санкціонований доступ до } p.$$

де U – множина акторів/користувачів, а P – множина дата-продуктів. Якщо $(u, p) \in acc$, це означає, що актор $u \in U$ має санкціонований (авторизований) доступ до продукту $p \in P$. Таким чином, відношення acc є основою для реалізації механізмів контролю доступу, які в Data Mesh контексті можуть ґрунтуватися на політиках федеративного управління та реалізовуватися на рівні інформаційно-технологічної платформи (Self-serve data platform).

Контракти на дані формалізують взаємні зобов'язання між продюсерами та споживачами, забезпечуючи структурну, якісну й організаційну узгодженість у використанні дата-продуктів.

Кожен контракт $c \in C$, описується як кортеж:

$$c = \langle prod(c), cons(c), sc(c), r(c), a(c), g(c), e(c) \rangle, \quad (3)$$

де $prod(c) \in P$ – продукт-джерело (producer); $cons(c) \in P$ – продукт-споживач (consumer); $sc(c) \in S$ – узгоджена схема контракту; $r(c) \in R^m$ – SLA-вектор (затримка, частота оновлення, строк зберігання даних тощо); $a(c) \subseteq U$ – дозволені актори (ACL); $g(c) \subseteq R_v$ – набір політик, яких слід дотримуватись; $e(c) \subseteq E$ – перевірки, що мають виконуватись.

Функція виконання політик і результат перевірки.

$$enf : K \times E \rightarrow R_v \cup C \quad (4)$$

визначає, яка перевірка $e \in E$, виконувана на певному платформному компоненті $k \in K$, забезпечує дотримання конкретного правила з множини R_v або контракту з множини C .

Результат виконання перевірки формалізується функцією:

$$pass : K \times E \rightarrow \{0, 1\},$$

де $pass(k, e) = 1$ означає успішне проходження перевірки, а $pass(k, e) = 0$ – її провал.

Інваріанти кодують базові принципи Data Mesh, забезпечуючи узгодженість між продуктами, контрактами та політиками. Кожна фактична залежність між продуктами має бути формалізована через контракт.

$$F \subseteq \{(prod(c), cons(c)) \mid c \in C\}.$$

Схема продукту-виробника має бути сумісною з цільовою схемою контракту.

$$\forall c \in C: Compat(s(prod(c)), sc(C)) = 1.$$

Кожен користувач, допущений контрактом, повинен мати авторизований доступ як до продукту-виробника, так і до продукту-споживача.

$$\forall c \in C, \forall u \in a(c) : (u, prod(c)) \in acc \wedge (u, cons(c)) \in acc.$$

Затримка між оновленням продукту-виробника і споживача не повинна перевищувати встановлений SLA-поріг.

$$t(cons(c)) - t(prod(c)) \leq r_{max-lag}(c), \forall c \in C.$$

Кожен продукт має однозначно визначений домен-власник.

$$\forall p \in P : d(p) \in D.$$

На операційному рівні архітектуру Data Mesh доцільно формалізувати через систему матриць та їх інваріантів, що відображають відношення контрактів, залежностей, доступу, SLA та якості. Представимо матрицю дата контрактів:

$$A^{contr} \in \{0,1\}^{n \times n},$$

де,

$$A_{ij}^{contr} = 1 \Leftrightarrow \exists c \in C : prod(c) = p_i, cons(c) = p_j.$$

Матриця контрактів фіксує наявність дата контракту між продуктами p_i та p_j .

При $n = |P|$ - відображає кількість дата продуктів, а при $A_{ij} = 1$ - існує дата контракт с від p_i до p_j .

Матриця залежностей матиме, наступне представлення:

$$O \in \{0,1\}^{n \times n}, O \leq A^{contr},$$

що відображає фактичні залежності між продуктами. Вимога $O \leq A^{contr}$ гарантує, що будь-яка залежність підтримується контрактом.

Представимо матрицю доступу користувачів:

$$R \in \{0,1\}^{h \times n},$$

де,

$$R_{ui} = 1 \Leftrightarrow u \in U \text{ має доступ до } p_i.$$

Сімейство матриць контрактів для користувачів (ACL-подання) – замість тензора вводимо множину:

$$A^{contr} = \{A^{contr}(u) \in \{0,1\}^{n \times n} \mid u \in U\},$$

де,

$$A^{contr}(u)_{ij} = 1 \Leftrightarrow \exists c \in C : prod(c) = p_i, cons(c) = p_j, u \in a(c).$$

Таким чином, для кожного користувача u формується власна матриця контрактів. Умова узгодженості доступів:

$$A^{contr}(u)_{ij} \leq R_{ui} \wedge R_{uj}, \forall u \in U, \forall i, j.$$

Представимо SLA-матрицю:

$$\Lambda \in R_{\geq 0}^{n \times n},$$

де Λ_{ij} задає максимальний допустимий лаг (затримку оновлення) між продуктами p_i та p_j .

Представимо вектор часових атрибутів:

$$T \in R_{\geq 0}^n,$$

де T_i – останній час оновлення продукту p_i .

Матриця сумісності схем, матиме наступний вигляд:

$$S \in \{0,1\}^{n \times n}, S \geq A^{contr},$$

що фіксує факт сумісності схем між продуктами.

Матриця якості даних матиме наступне представлення:

$$Q \in [0,1]^{n \times r},$$

де Q_{ik} – оцінка якості продукту p_i за метрикою k (повнота, точність, узгодженість тощо).

Глобальний індикатор узгодженості матиме наступне вираження:

$$Conf = 1[0 \leq A^{contr}] \wedge 1[(T_j - T_i) \leq \Lambda_{ij}, \forall A_{ij}^{contr} = 1] \wedge 1[A^{contr}(u)_{ij} \leq R_{ui} \wedge R_{uj}] \wedge 1[S_{ij} \geq A_{ij}^{contr}].$$

Цей індикатор показує, чи виконуються всі ключові умови системи: формалізація залежностей даних контрактами, відповідність SLA, коректність доступів та узгодженість схем. Значення $Conf = 1$ означає глобальну відповідність Data Mesh принципам і гарантує, що архітектура перебуває у валідному стані.

Data Mesh дозволяє таким командам працювати незалежно, узгоджуючи свої дії через формальні контракти та політики. Цей підхід знижує навантаження на центральну архітектурну команду, розподіляє відповідальність за дані, а також підтримує узгодженість між доменними командами, незалежно від їхньої локації чи часових поясів. У таких умовах підвищується швидкість прийняття рішень, зростає продуктивність, а ризики через затримки або неякісні дані – суттєво знижуються.

Роль Data Mesh у процесах функціонування розподілених проєктних командах

У сучасному інформаційно насиченому середовищі організації дедалі частіше функціонують у контексті розподіленості, яка проявляється на трьох рівнях: функціональному (розподіл завдань між різними ролями та спеціалізаціями), географічному (наявність команд у різних країнах, містах чи офісах), а також часовому (відкладена взаємодія між учасниками, які працюють асинхронно). Особливо це актуально для IT-компаній, логістичних підприємств тощо, де одночасно функціонують десятки чи навіть сотні мікросервісів, внутрішніх інструментів та клієнтських інтерфейсів, підтримуваних командами, що розташовані у різних регіонах.

У таких умовах постають специфічні виклики, пов'язані з ефективною координацією та управлінням даними: розмитість відповідальності за якість даних, затримки в обміні інформацією між командами аналітики та розробки, дублювання логіки, складність у версіонуванні схем та нестача узгоджених стандартів. Централізовані підходи до управління даними (на кшталт Data Lake чи DWH) не здатні масштабуватися у таких гетерогенних організаціях без виникнення «вузьких місць» у розподілі обов'язків та синхронізації процесів.

Data Mesh як інноваційна системи організації даних для розподілених проєктних команд пропонує децентралізований підхід до подолання проблем узгодженості та масштабованості, притаманних централізованим архітектурам. Це досягається шляхом формування незалежних доменних команд, кожна з яких несе повну відповідальність за створення, розвиток і підтримку власних дата-продуктів. Такі команди взаємодіють на основі формалізованих контрактів на дані (data contracts), які визначають схему даних (data schema), правила доступу, угоди про рівень обслуговування (SLA), вимоги до версіонування, а також умови обміну та відповідності. У такий спосіб забезпечується передбачуваність, повторне використання та узгодженість продуктів даних між доменами.

Data Mesh виступає платформою узгодження між бізнес-ролями (власники продуктів, аналітики), технічними командами (DevOps, backend, QA, mobile, DWH) та координуючими функціями (архітектори, платформа, безпека), синхронізуючи очікування щодо якості, доступності та інтероперабельності даних. Важливою роллю тут є Data Product Owner – особа, відповідальна за життєвий цикл дата-продукту, яка діє на перетині бізнесу, розробки й аналітики. Впровадження Data Mesh в організаціях з розподіленою структурою дозволяє уникнути залежності від централізованих команд, прискорити ітерації, покращити прозорість взаємодії та підвищити зрілість у поводженні з даними як зі стратегічним активом.

Приклад використання архітектури Data Mesh для логістичної компанії

Інституціоналізація принципів Data Mesh у логістичному середовищі потребує ретельної адаптації до специфіки доменів, взаємозв'язків між командами та інформаційної архітектури організації (TheOpenGroup, 2024). Особливості логістичного бізнесу – мультिकанальність, взаємодія з партнерами, високий рівень автоматизації операцій та географічна децентралізація – створюють сприятливе середовище для впровадження децентралізованої моделі управління даними.

Представлена нижче архітектура (рис. 2) демонструє спосіб імплементації Data Mesh у великій логістичній компанії, що працює як у сегментах B2B (вантажні перевезення, контейнерні ланцюги) так і в B2C/eCommerce (доставка до кінцевих клієнтів, C2C-сервіси). У моделі враховано як функціональні домени, пов'язані з основною діяльністю, так і технічні домени, відповідальні за інфраструктуру, підтримку та інновації.

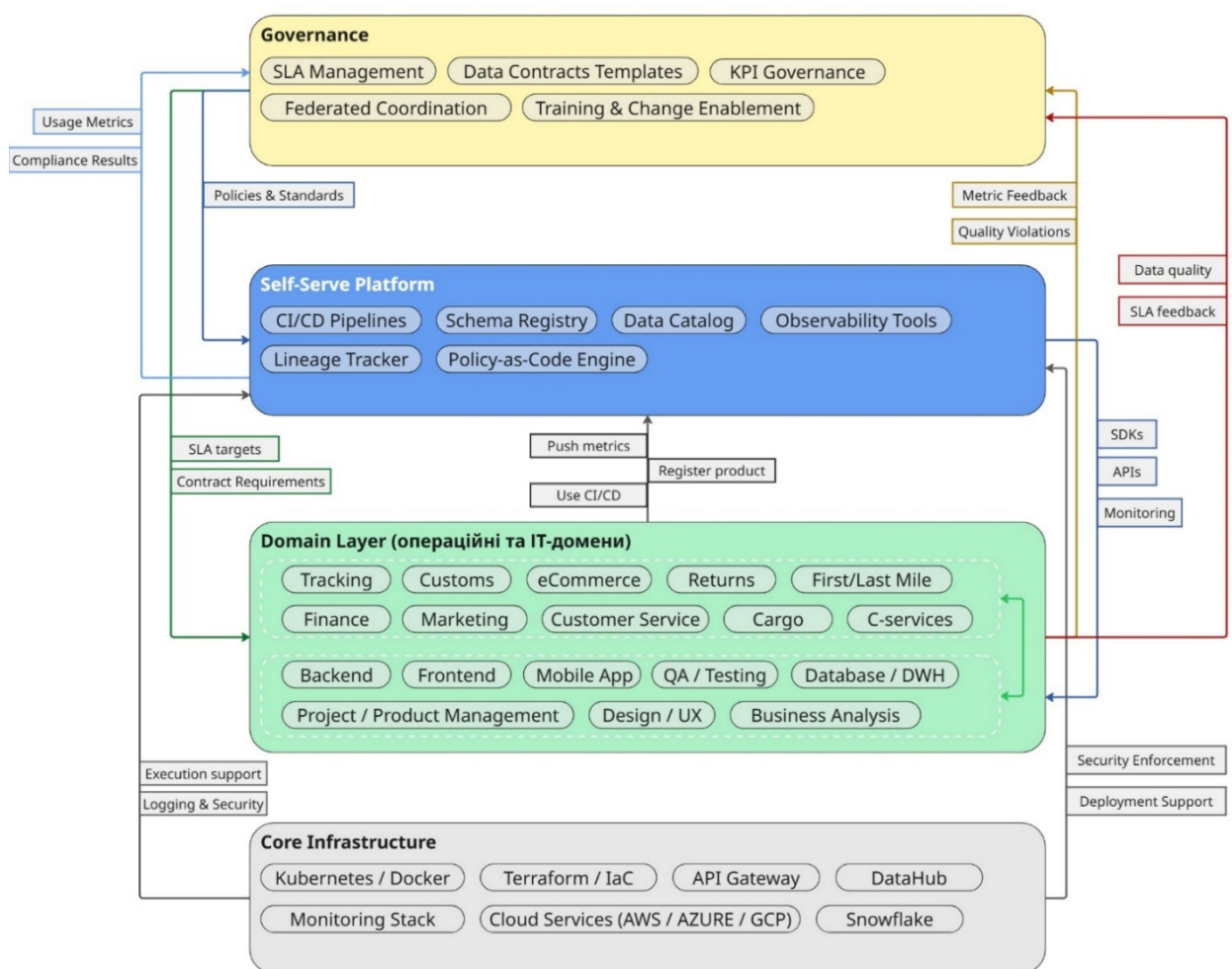


Рис. 2. Архітектура система організації даних Data Mesh для логістичної компанії з розподіленими проєктними командами

На рис. 2 представлено узагальнену архітектуру інноваційної системи організації даних Data Mesh у логістичній компанії з високим рівнем цифрової зрілості та географічно розподіленими командами. Запропонована архітектура включає чотири взаємопов'язані рівні: керування (governance), інформаційно-технологічної платформи, domenів та базової інфраструктури.

Рівень керування (governance) реалізує функції стратегічного управління даними: формування шаблонів контрактів на дані (data contracts), визначення цільових значень угод про рівень обслуговування (SLA), встановлення ключових показників ефективності (KPI), а також координацію між доменами. Цей рівень також відповідає за онбординг, навчання та супровід змін в організації. Зворотний зв'язок на основі метрик якості даних спрямовується вгору, забезпечуючи ітеративні цикли покращення.

На рівні платформи самообслуговування функціонують компоненти, що забезпечують інструментальну підтримку дата-команд: CI/CD пайплайни, реєстр схем, каталоги даних, модулі спостережуваності (observability), трекери лінійності даних (lineage) та рушії політик (Policy-as-Code Engine). Цей рівень виступає як інтерфейс між інфраструктурою і доменами, забезпечуючи масштабованість і стандартизацію процесів.

Доменний рівень (Domain Layer) охоплює як операційні, так і IT-домени, кожен з яких створює незалежні продукти даних зі специфічними SLA, API, схемами та споживачами. Доменні команди несуть відповідальність за повний життєвий цикл продукту – від моделювання до реєстрації у каталозі та забезпечення споживання іншими доменами. У представленій архітектурі виділено ключові операційні домени (наприклад, Tracking, Customs, eCommerce, Returns, Customer Service), а також домени технічної підтримки, зокрема Backend, Frontend, Mobile App, QA/Testing, Database, UX, Business Analysis та Project/Product Management. Важливим є те, що IT-домени розглядаються не лише як сервісні, а як активні учасники Data Mesh, що виробляють власні стратегічні продукти даних (наприклад, release-success-rate, bug-distribution, crash-report-analytics, schema-change-log).

Базова інфраструктура (Core Infrastructure) реалізує базову підтримку: хостинг контейнерів (Kubernetes), інфраструктура як код (Terraform), керування доступом, логування, API-шлюзи та інтеграцію з хмарними провайдерами (AWS, Azure, GCP). Саме ця інфраструктура забезпечує стійкість і масштабованість Data Mesh-архітектури в умовах розподілених систем.

У таблиці 5 узагальнено приклади дата-продуктів у межах domenів, які формують ядро операційної та аналітичної роботи в логістичній компанії.

Таблиця 5

Приклади потенційних дата-продуктів у доменах логістичної компанії

Домен	Приклади дата-продуктів	
Tracking (Відстеження)	parcel-status-stream, delivery-eta-prediction, hub-scan-log	потік-статусів-відправлень, прогноз-часу-доставки, журнал-сканувань-хабу
Customs (Митниця)	declaration-events, tax-rule-api, customs-delay-forecast	події-декларування, API-податкових-правил, прогноз-митних-затримок
eCommerce (Електронна комерція)	order-fulfillment-timeline, drop-off-point-usage	хронологія-виконання-замовлення, використання-пунктів-прийому
Returns (Повернення)	return-reasons-log, refund-time-analysis	журнал-причин-повернень, аналіз-часу-відшкодувань
Customer Service (Клієнтський сервіс)	ticket-resolution-metrics, response-time-distribution	метрики-вирішення-звернень, розподіл-часу-відповіді
Backend (IT) (Серверна логіка IT-систем)	shipment-core-events, pricing-rule-log, fraud-alert-stream	ключові-події-доставки, журнал-цінових-правил, потік-сповіщень-про-шахрайство
Mobile App (Мобільний застосунок IT-системи)	crash-reports, session-lengths, device-compatibility	звіти-про-збої, тривалість-сеансів, сумісність-пристроїв
QA / Testing (Забезпечення якості / Тестування)	test-suite-runs, bug-type-ratio, release-failure-log	виконання-наборів-тестів, співвідношення-типів-помилки, журнал-збоїв-релізу
Business Analysis (Бізнес-аналіз IT-систем)	story-estimation-accuracy, feature-adoption-tracker	точність-оцінки-історій, трекер-прийняття-функціоналу

Наведена архітектура дозволяє доменним командам автономно створювати, публікувати та масштабувати дата-продукти, зберігаючи при цьому загальну відповідність політикам організації. Це є визначальною умовою ефективного функціонування Data Mesh у розподіленому середовищі логістичних підприємств.

Інформаційно-технологічна платформа самообслуговування для підтримки проєктних процесів

Ефективне функціонування Data Mesh-архітектури у розподілених організаціях неможливе без побудови високорівневої self-serve платформи, яка забезпечує доменні та технічні команди необхідними інструментами для повного життєвого циклу роботи з даними. Така платформа повинна включати компоненти, що дозволяють командам створювати, реєструвати, розгортати та моніторити свої дата-продукти без залучення централізованого ІТ-відділу.

Ключові компоненти платформи:

- Data Catalog – система каталогізації дата-продуктів, що містить їхні метадані, контактні особи, SLA, та глосарії доменів;
- Pipeline Orchestrator – модуль оркестрації обробки даних (наприклад, Apache Airflow, Dagster) із підтримкою DAG-графів, графіків виконання та відновлення;
- Schema Registry – централізоване сховище схем (Avro, JSON Schema, Protobuf) із перевіркою сумісності версій;
- Access Control – інструменти управління доступом на основі політик, ролей і доменної відповідальності;
- Observability Tools – моніторинг пайплайнів, валідація якості (наприклад, OpenLineage, Great Expectations, Soda).

Суттєвим аспектом є підтримка DataOps – методології автоматизації розгортання, тестування, контролю версій та якості даних. У поєднанні з GitOps-практиками, де всі зміни до пайплайнів, контрактів та метаданих зберігаються у Git-репозиторії і розгортаються через CI/CD, це створює передумови для масштабованої, надійної та audit-ready інфраструктури управління даними.

Інтеграція AI-асистентів значно розширює можливості платформи. Інструменти на зразок Claude, GPT-4 або Perplexity можуть генерувати технічну документацію (PRD, risk assessment), пояснювати терміни з глосаріїв, формувати запити до Data Catalog, а також створювати візуальні артефакти (Mermaid, Excalidraw, Whimsical). Наприклад, система може автоматично формувати шаблони data contracts або підтримувати семантичний пошук по історії змін у продуктах даних. Платформи типу DataMesh Manager вже включають такі функції: управління контрактами, Data Product Canvas, Change Request Workflow, інтеграцію з Git, CI/CD, автоматизовані перевірки якості даних і валідації схем.

Приклад 1. Дата-контракт для домену "Митниця"

Назва продукту: customs_delay_forecast;

Домен: Customs (митне оформлення);

Тип: аналітичний прогностичний дата-продукт;

Формат контракту: YAML (опис специфікації);

Формат даних продукту: Parquet (таблична модель з support for time series);

Призначення: *побудова прогностичної моделі затримок на митниці з урахуванням країни відправлення, обсягу відправлень і типу товару. Продукт використовується відділом ризик-менеджменту для SLA-коригування та пріоритезації обробки.*

Огляд YAML-структури:

```
id: customs_delay_forecast
info:
  title: "Forecast of Customs Delays"
  version: "1.0.0"
domains:
  - customs
...
```

Приклад 2. Дата-контракт для домену "Клієнтський сервіс"

Назва продукту: *ticket_resolution_times*;

Домен: *Customer Service* (клієнтська підтримка);

Тип: *операційний агрегований дата-продукт*;

Формат контракту: *YAML* (опис специфікації);

Формат даних продукту: *JSON Lines* (event stream);

Призначення: *моніторинг середнього часу закриття звернень у системі підтримки з поділом за каналами (email, чат, дзвінки). Використовується для контролю SLA та визначення точок навантаження.*

Огляд *YAML*-структури:

```
id: ticket_resolution_times
info:
  title: "Customer Ticket Resolution Metrics"
  version: "0.9.5"
  domains:
    - customer-service
...
```

Приклад 3. Дата-контракт для домену "Маркетинг"

Назва продукту: *promo_conversion_funnel*;

Домен: *Marketing* (маркетинг);

Тип: *Аналітичний дата-продукт з часовими агрегатами*;

Формат контракту: *YAML* (опис специфікації);

Формат даних продукту: *Parquet* (таблична форма з таймстемпами);

Призначення: *відстеження ефективності маркетингових кампаній через побудову воронки конверсії – покази, переходи, реєстрації, замовлення. Продукт використовується для A/B тестування, оптимізації витрат на рекламу та прогнозування ROI (Return on Investment).*

Огляд *YAML*-структури:

```
id: promo_conversion_funnel
info:
  title: "Conversion Funnel Metrics for Promotional Campaigns "
  version: "1.0.0"
  domains:
    - marketing
...
```

У Data Mesh підхід до управління даними відходить від централізованих директивних моделей на користь федеративного керування, яке поєднує автономію доменів із дотриманням єдиних стандартів. Ключові принципи:

- Data Contracts – формалізовані угоди між продюсерами та споживачами даних, що включають структуру, SLA, політики доступу.
- Глосарії та стандарти метаданих – загальні доменні словники, що уніфікують інтерпретацію понять, полів та моделей.
- Автоматизовані перевірки (Data Governance as Code) – перевірки на кожному етапі пайплайна (schema compatibility, freshness, completeness, PII scan) виконуються автоматично.
- Governance Team / Federated Council – міждисциплінарна структура, яка підтримує стандарти, проводить аудит, фасилітує впровадження політик.

Таким чином, Data Mesh Governance функціонує як «платформа правил», де перевірки та аудит відбуваються не вручну, а вбудовані в автоматизований контекст пайплайна, забезпечуючи масштабовану довіру до даних і мінімізацію ризиків при збереженні гнучкості.

Висновки

У дослідженні проведено системний аналіз і формалізоване обґрунтування доцільності впровадження інноваційної системи організації даних Data Mesh як ефективного інструменту підтримки проєктних процесів у розподілених командах, зокрема в умовах географічної, функціональної та часової децентралізації. Запропоновано структурну багаторівневу архітектуру, що поєднує доменну відповідальність за дані, контрактну взаємодію між командами, інформаційно технологічну платформу самообслуговування, а також механізми федеративного управління політиками. На відміну від традиційних централізованих архітектур (Data Warehouse, Data Lake), підхід Data Mesh дозволяє забезпечити масштабованість, автономність та прозорість опрацювання даних у межах багатодомених організацій.

Архітектура Data Mesh дозволяє не лише структурно моделювати взаємодію між доменами, а й закласти основу для подальшої автоматизації перевірок, трасування змін та контролю якості. Окремо обґрунтовано роль інформаційно технологічної платформи, яка забезпечує інструментальну підтримку команд за допомогою інструментів CI/CD, Data Catalog, Schema Registry, Policy-as-Code, систем спостережуваності та інших компонентів DataOps-інфраструктури.

Результати дослідження інтегруються з дисертаційними дослідженнями, які спрямовані на розробку інформаційних технологій супроводу проєктних процесів у розподілених ІТ-командах. Запропонована модель Data Mesh відкриває нові напрями для подальших наукових досліджень, зокрема в напрямі вимірювання надійності та узгодженості продуктів даних, інтеграції підходів «Інженерії хаосу» (Chaos Engineering) для управління ризиками, автоматизованої генерації та перевірки Data Contracts, а також розробки метрик еволюції та адаптивності дата-екосистем. Розроблені підходи можуть стати основою для формування методологій цифрової стійкості у розподілених організаційних структурах. Тому система організації даних Data Mesh для розподілених команд демонструє потенціал для формування стійкої інфраструктури управління даними у багатодомennomu середовищі. Подальший розвиток цього підходу передбачає створення інформаційно-технологічної платформи з підтримкою Data Catalog, Schema Registry та Policy-as-Code, що забезпечить автоматизовану контрактну взаємодію між доменами й контроль якості даних у реальному часі. Перспективними напрямками досліджень є розроблення формальних методів оцінки узгодженості й надійності дата-продуктів, інтеграція методів імітаційного моделювання та інструментів DataOps для безперервного моніторингу, а також апробація архітектури на прикладах масштабних розподілених проєктів. Це дозволить поглибити теоретико-методичні засади Data Mesh та забезпечити практичні рекомендації щодо його застосування у проєктних процесах розподілених ІТ-команд.

СПИСОК ЛІТЕРАТУРИ

- Ashraf, A., Hassan, A., & Mahdi, H. (2023). Key lessons from microservices for Data Mesh adoption. 2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), 1–8. <https://doi.org/10.1109/MIUCC58832.2023.10278300>
- Christ, J. (2025). Data Mesh Architecture. <https://www.datamesh-architecture.com/>
- DataContract. (2025). Data contract specification [Online specification]. <https://datacontract.com/>
- Dehghani, Z. (2019, May 20). How to move beyond a monolithic data lake to a distributed data mesh. MartinFowler.com. <https://martinfowler.com/articles/data-monolith-to-mesh.html>
- Dehghani, Z. (2020, December 3). Data Mesh principles and logical architecture. MartinFowler.com. <https://martinfowler.com/articles/data-mesh-principles.html>
- Dehghani, Z. (2022). Data Mesh: Delivering data-driven value at scale. O'Reilly Media.
- Dibouliya, A., & Jotwani, V. (2023). Review on Data Mesh architecture and its impact. Journal of Harbin Engineering University, 44(7), 2353–2363. <https://harbinengineeringjournal.com/index.php/journal/article/view/809>
- Duda, O., Zakharija, O., Kramar, T., Melnyk, A., & Skaletskyi, P. (2025). System architecture of data organization for smart cities based on the data mesh concept, Journal of Lviv Polytechnic National University "Information Systems and Networks", 17, 411–424. <https://doi.org/10.23939/sisn2025.17.411>
- Fortune Business Insights. (2024). Big data analytics market size, share & COVID-19 impact analysis 2023–2030. <https://www.fortunebusinessinsights.com/big-data-analyticsmarket-106179>

- Inmon, W. H., Linstedt, D., & Levins, M. (2019). *Data architecture: A primer for the data scientist* (2nd ed.). Academic Press.
- Jonkman, C. (2023). Organisational maturity assessment during the paradigm shifts from monoliths to Data Mesh [Master's thesis, Delft University of Technology]. <http://resolver.tudelft.nl/uuid:294d7df5-511c-4149-9507-21be6379375d>
- Perrin, J.-G., & Broda, E. (2024). *Implementing Data Mesh: Principles and practice to design, build, and implement Data Mesh*. O'Reilly Media.
- Pongpech, W. A. (2023). A distributed Data Mesh paradigm for an event-based smart communities monitoring product. *Procedia Computer Science*, 220, 584–591. <https://doi.org/10.1016/j.procs.2023.03.074>
- The Open Group. (2024). *The ArchiMate enterprise architecture modeling language*. The Open Group. <https://www.opengroup.org/archimateforum/archimate-overview>
- Vestues, K., Hanssen, G. K., Mikalsen, M., Buan, T. A., & Conboy, K. (2022). Agile data management in NAV: A case study. In V. Stray, P. M., & K. P. Stol (Eds.), *Agile processes in software engineering and extreme programming* (pp. 220–235). Springer International Publishing. 10.48550/arXiv.2204.09979
- Zhovnir, Y., Kunanets, N., Burov, Y., Duda, O., & Pasichnyk, V. (2025). Designing the structure and architecture of situation-aware security information systems for residential complexes. *Eastern-European Journal of Enterprise Technologies*, 1(9 (133), 6–23. <https://doi.org/10.15587/1729-4061.2025.315248>

DATA MESH – AN INNOVATIVE DATA ORGANIZATION SYSTEM FOR DISTRIBUTED PROJECT TEAMS

Roman Vaskiv, Nataliia Veretennikova

Lviv Polytechnic National University,
Information Systems and Networks Department, Lviv, Ukraine
E-mail: roman.i.vaskiv@lpnu.ua, ORCID: 0000-0002-8549-5035
E-mail: nataliia.v.veretennikova@lpnu.ua, ORCID: 0000-0001-9564-4084

© Vaskiv R., Veretennikova N., 2025

The article presents Data Mesh as an innovative system of data organization designed to enhance the efficiency of distributed project teams operating in a dynamic digital environment. It highlights the limitations of centralized data organization systems, such as Data Warehouse and Data Lake, which tend to lose adaptability and scalability within multi-domain structures. The purpose of this research is to provide a scientific rationale for the principles of Data Mesh and to determine its potential.

Data Mesh is built on four core pillars such as domain-oriented data ownership, data as a product, a self-serve data platform, and federated governance. The Data Mesh framework enables the formalization of interactions through data contracts, the integration of observability mechanisms, lineage tracking, and automation tools. Its application is demonstrated through the case of a logistics company, where domain teams create and maintain their own data products, while consistency is ensured through standards and centralized policies.

The findings confirm that Data Mesh establishes a foundation for building resilient, scalable, and flexible data organization systems that combine domain autonomy with global coordination. This creates favorable conditions for increased productivity, transparency, and decision-making speed in distributed environments, while also opening new avenues for further research in the areas of data quality management and the optimization of project processes in distributed IT teams.

Keywords: Data Mesh, distributed project teams, data organization system, data architecture.