

**Ольга Володимирівна Рибак**

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій проектування та дизайну  
Національний університет "Одеська політехніка" (Одеса, Україна)

E-mail: [olga.vol.rybak@gmail.com](mailto:olga.vol.rybak@gmail.com). ORCID: <http://orcid.org/0000-0002-0250-3037>

SCOPUS Author ID: 57208656221

**ПРОЄКТУВАННЯ СИСТЕМИ ЗБЕРІГАННЯ ДАНИХ ДЛЯ САПР  
НА ОСНОВІ ХМАРНОЇ ІНФРАСТРУКТУРИ**

*У статті розглянуто архітектуру системи зберігання даних, призначеної для дизайну деталей у САПР, на основі хмарної платформи. Структурована інформація, яка зберігається в базі даних системи, та неструктурована, записана у вигляді файлів на хмарі, з розподілених і гетерогенних ресурсів об'єднується в уніфікований набір даних про деталі та розміщується на хмарі. Через адаптер функціонального ядра хост системи може отримати доступ до всіх сервісів, імпортуючи надані інструменти у власні фреймворки. Таким чином, інтегроване управління пулом ресурсів забезпечує налаштування системи зберігання та переміщення даних без переривання роботи поточних програм, що сприяє розгортанню бібліотеки деталей на хмарі.*

**Ключові слова:** хмарні технології; система зберігання даних; САПР; платформа як сервіс; хмарне виробництво.  
Рис.: 3. Бібл.: 7.

**Актуальність теми дослідження.** Стрімкий розвиток хмарних технологій докорінно змінює традиційні принципи проектування та новітнього виробництва, коли віртуалізовані ресурси можуть динамічно налаштовуватись та надаватись як послуга через інтернет. Хмарна платформа реалізує можливості та базову інфраструктуру для обміну даними і послугами від віддалених постачальників, а користувачі отримали інструмент для пошуку необхідних матеріалів з розподілених джерел по всьому світу.

Парадигма хмарного виробництва охоплює різні галузі сучасної промисловості, застосовуючи в організації виробничих процесів інтернет речей, штучний інтелект та хмарні обчислення. Таким чином забезпечується управління, оптимізація і надання інформаційних ресурсів через мережу. Перевагами цього підходу є значна гнучкість та масштабованість процесів, віддалений доступ до даних і автоматизація протягом усього життєвого циклу продукту, що знижує витрати підприємств на IT-інфраструктуру [1].

У програмному забезпеченні для конструкційного проектування перехід до роботи з інтегрованою хмарною платформою відкриває широкі можливості завдяки вдосконаленим інструментам, незалежності від вибору САПР і безперешкодному доступу до моделей та технічної документації, що дозволяє спростити роботу з конструювання та дизайну виробів.

**Постановка проблеми.** На відміну від хмарного виробництва, використання САПР у моделюванні та промисловий інтернет речей переважно пов'язані з інтеграцією розподілених ресурсів для виконання одного технічного завдання. Процесам дизайну та виготовлення деталей на базі такого підходу бракує централізованого управління операціями, вибору різних режимів роботи та вбудованого доступу до необхідного обладнання та ресурсів. Зокрема, сучасні САПР пропонують інженерам і розробникам бібліотеки готових елементів та набори макетів, призначені для конкретних програмних комплексів, що прив'язує їх до однієї платформи і ускладнює доступ до них користувачів інших сервісів. Таким чином, розробникам не уникнути дублювання інформації при конструюванні деталей на різних САПР, узгодження суперечностей між різними програмними комплексами чи необхідності багаторазового проектування однакових деталей для різних платформ. Хмарне виробництво, що спирається на систему зберігання даних із загальним доступом, натомість може вирішити проблему допуску до напрацьованих креслень і 3D моделей та їхнього подальшого використання для інших проєктів.

**Аналіз останніх досліджень і публікацій.** Серед робіт, присвячених архітектурній еволюції та стратегічному аналізу провідних хмарних систем зберігання даних, значна

увага присвячена дослідженню їхнього перетворення зі звичайних віртуальних репозиторіїв на складні інтелектуальні платформи, що забезпечують роботу штучного інтелекту, аналітики великих даних та глобальних корпоративних вебресурсів. Однією з провідних таких систем стала Amazon Simple Storage Service (S3), побудована на моделі "eventual consistency" (узгодженості підсумкового результату) для операцій оновлення та видалення об'єктів. Цей підхід було детально розглянуто в науковій праці Вернера Фогельса [2], а дослідження архітектури й динамічна оптимізація параметрів, виконана в роботі [3], продемонструвала переваги використання Amazon S3 для великих аналітичних навантажень.

Пристрої, що працюють на операційній системі Windows, можуть легко інтегруватися з системою зберігання Microsoft Azure Storage, внутрішня структура якої розподілена на три логічні шари: нижній рівень, що відповідає за фізичне зберігання даних на дисках, проміжний рівень, який керує глобальним простором імен, балансуванням навантаження та транзакційною цілісністю, а також рівень front-end на базі серверів, які перевіряють і ці маршрутизують запити [4].

Для тренування нейромереж та інших завдань з великим навантаженням найкращим вибором є система Google Cloud Storage на основі файлової системи Colossus. Завдяки оптимізації продуктивності з використанням машинного навчання, масштабованості та швидкому доступу до архівів, ця система може стати фундаментом високопродуктивної аналітики [5].

Сучасні системи зберігання даних передбачають децентралізоване управління та масивну реплікацію інформації, що дозволяє витримувати одночасний вихід з ладу кількох центрів обробки, забезпечуючи найвищий ступінь надійності. Важливими характеристиками таких систем є швидкість розгортання застосунків на їхній основі, безпека інтелектуальної власності та стійкість архітектури до збоїв. Відтак, хмарне сховище більше не можна вважати ізольованим компонентом, адже воно стало фундаментом розвитку новітньої ІТ-індустрії.

**Виділення недосліджених частин загальної проблеми.** Попри широкі можливості описаних хмарних систем зберігання даних, у них практично відсутній інструментарій для кросплатформного, тобто незалежного від вибору САПР, доступу до моделей деталей у різних форматах файлів і внесення змін до них у режимі реального часу відповідно до вхідних даних віддалених користувачів. Уніфікована бібліотека деталей, що зберігається на хмарі, повинна з'єднувати користувачів, 3D-платформи або інші хост-системи, за допомогою яких клієнти можуть швидко викликати інтерфейси для редагування макетів, а також забезпечувати можливість синхронної підтримки кількох різних сцен проектування.

**Метою статті** є проектування архітектури системи зберігання даних, призначеної для дизайну деталей у САПР, з інтегрованою хмарною платформою, а також встановлення особливостей розгортання такої системи для гетерогенного програмного забезпечення різних САПР.

**Виклад основного матеріалу.** Використання системи зберігання даних на хмарній інфраструктурі для виробництва деталей ґрунтується на моделі, що забезпечує загальнодоступний та зручний доступ за запитом через Інтернет до спільного набору відомостей та макетів, параметри яких можна налаштовувати. Загалом існує два підходи до способу організації структури такої системи. Концепція інтеграції розподілених ресурсів працює за логікою прихованої системи підтримки прийняття виробничих рішень на основі хмари, яка передбачає ідентифікацію різних виробничих процесів, інтеграцію ресурсів та їхнє об'єднання у єдиний сервіс хмарних обчислень. Таким чином, категорії даних та фізичний розподіл джерел є прозорими та відображаються як єдине ціле для споживачів. На противагу цьому, інший підхід базується на розподілі інтегрованих ресурсів, де єдиний інтерфейс провайдера може бути налаштований розподіленими клієнтами через мережу відповідно до їхніх конкретних цілей. На рис. 1 представлено структуру цих підходів та їхні відмінності.

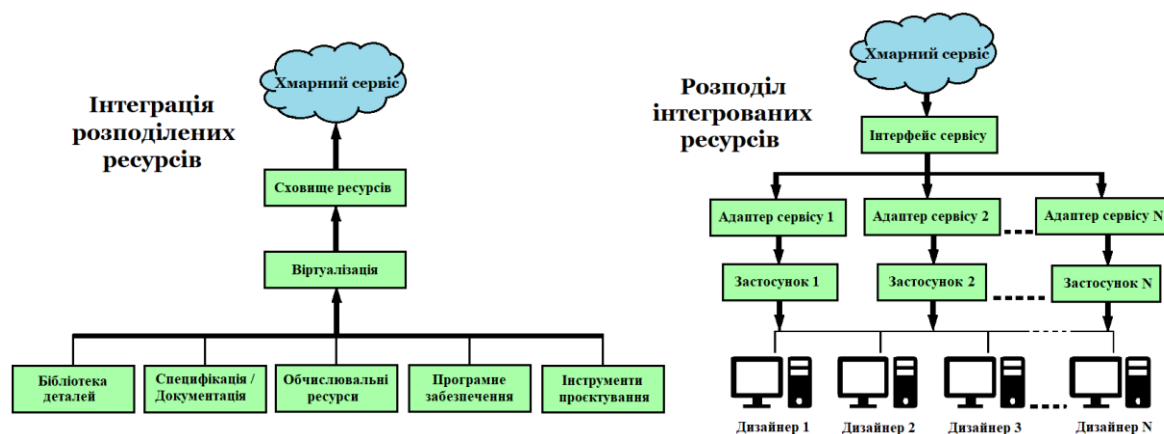


Рис. 1. Два підходи до організації структури системи зберігання даних на хмарі  
Джерело: розроблено автором.

Інтеграція розподілених ресурсів реалізує процес одночасного управління базовою інфраструктурою, включаючи програмне забезпечення, апаратне забезпечення та безпосередньо хмару, а також керує оброблюваними даними, зокрема налаштуванням параметрів, 3D-моделями та іншою пов'язаною з ними інформацією з бібліотеки деталей, розподіленою на різних сайтах і навіть серверах. Отже, функціональне ядро розробленої системи працюватиме як середовище-оболонка на основі схеми PaaS (платформа як сервіс), що сприятиме централізації розподілених ресурсів на хмарі. Модель PaaS дозволяє керувати налаштуваннями програми та її розміщенням на платформі без можливості впливати на хмарну інфраструктуру, а сервери та обчислювальні потужності контролюються провайдером [6]. Розподіл інтегрованих ресурсів, з іншого боку, забезпечує механізм отримання безперешкодного доступу до системи для віддалених користувачів, результат якого може бути найрізноманітнішим залежно від потреб конкретного завдання.

Концепція PaaS, вбудована в адаптери функціонального ядра, об'єднує провайдера із зовнішніми застосунками по взаємосумісному двосторонньому каналу з'єднання. Найбільше складнощів на цьому етапі може виникнути при перенесенні програмного інтерфейсу функціонального ядра системи на локальну платформу. Таким чином, адаптери функціонального ядра системи поєднують обидва підходи та злагоджують суперечності між ними за рахунок обміну даними та сервісами в спільному модулі. Наприклад, САА (Component Application Architecture) може працювати як конектор для провайдера функціонального ядра та програми САТІА, ObjectARX відіграватиме таку ж роль для AutoCAD [7].

Розгортання системи зберігання виробничих даних на хмарі, включаючи конструкторську документацію, специфікації, бібліотеку стандартних деталей тощо, можна розглядати в ролі спрощеної версії хмарного виробництва, більш доступної в реалізації. Система обслуговування хмарного виробництва загалом включає п'ять рівнів: рівень ресурсів, рівень проміжного програмного забезпечення, рівень ядра системи, рівень порталу та рівень застосунків (рис. 2). Рівень ресурсів включає такі елементи, як відомості макетів і креслень для подальшого дизайну, перелік матеріалів, джерела інформації, функції для інтеграції деталей, їхнього тестування та симуляції збірки.

Рівень проміжного програмного забезпечення підтримує віртуалізацію, обслуговування, надання доступу, представлення та координацію даних. Ядро системи ґрунтується на інтерфейсі проміжного програмного забезпечення і містить всі важливі функції хмарного виробництва, зокрема розгортання та реєстрацію сервісів, пошук та зіставлення потрібних результатів, планування розробки та її перевірку на відмовостійкість. Рівень

порталу являє собою загальний канал для провайдерів сервісу, операторів платформи та користувачів для управління виробничими ресурсами та можливостями. Рівень застосунків дозволяє працювати у форматі одного агента, спільному режимі та режимі проектування за запитом.



Рис. 2. Рівні системи обслуговування хмарного виробництва

Джерело: розроблено автором.

Розгортання виробничих ресурсів разом з бібліотекою 3D-моделей деталей передбачає не лише хмарне зберігання різних файлів і документів, адже функціонал, реалізований за рахунок хмарних обчислень, є не менш важливою складовою створюваної програми. У багатьох випадках оптимізація обробки запитів та зручний доступ до необхідних даних, які накопичувалися з різних дизайн-проектів протягом тривалого часу, відіграє вирішальну роль у підвищенні ефективності всієї системи.

Ресурси, представлені в системі зберігання даних для конструкційного проектування включають відомості з бібліотеки моделей деталей, зокрема структуровані дані, такі як специфікації деталей, їхній розмір, параметри, матеріал, точність, шорсткість, ступінь обробки поверхні тощо, а також неструктуровані дані, серед яких фотографії та інші зображення деталей, двовимірні креслення, 3D-моделі, пов'язані документи тощо. Структурована інформація зберігається в базі даних системи, неструктурована – у вигляді файлів на хмарі. Ці розподілені та гетерогенні ресурси, що інколи мають різні формати файлів, об'єднуються на інтеграційному рівні в уніфікований набір даних про деталі та розміщуються на хмарі. Основні функції програми реалізовані на сервері, а на комп'ютерах користувачів потрібно розгорнути адаптер для доступу до локальних програм з налаштуванням інструментів. Через адаптер функціонального ядра хост може отримати доступ до всіх сервісів, імпортуючи надані інструменти у власні фреймворки. Архітектура такої системи наведена на рис. 3.

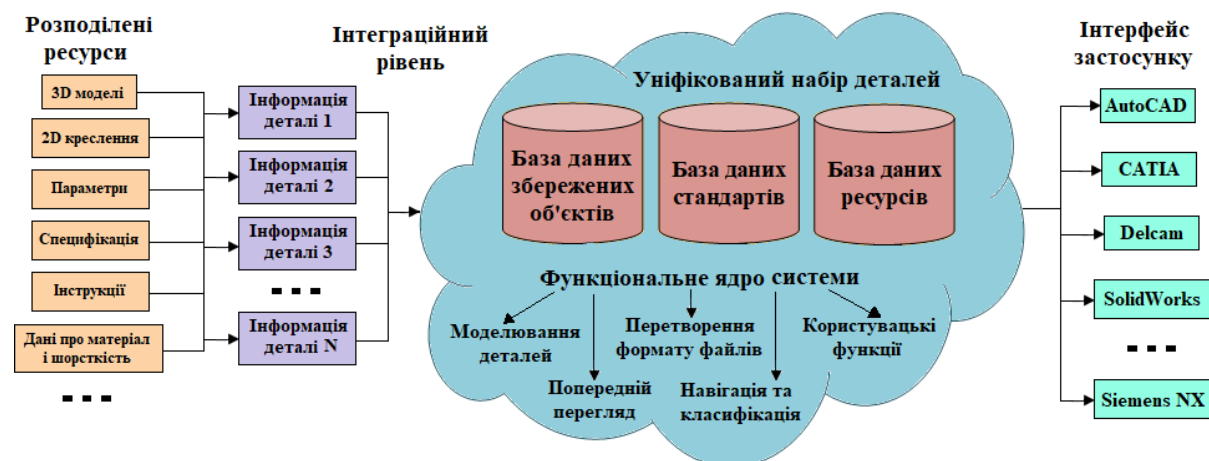


Рис. 3. Загальний вигляд архітектури системи

Джерело: розроблено автором.

Віртуалізація даних, яка забезпечується в запропонованій архітектурі системи зберігання, створює єдиний логічний шар абстракції та надає уніфікований доступ до розрізнених джерел у реальному часі без потреби фізичного копіювання потрібних файлів. Такий підхід підвищує динамічну адаптивність програми завдяки відокремленню користувачів від джерел даних, що дозволяє працювати з нею на гетерогенному програмному забезпеченні різних САПР. Логічна централізація фрагментованих хмарних ресурсів, виконана на інтеграційному рівні, сприяє оптимізації витрат на інфраструктуру, впровадженню єдиних стандартів безпеки та формуванню гнучкого фундаменту програмного забезпечення для сучасного конструкційного проектування.

**Висновки.** У процесі дослідження встановлено, що незважаючи на широкі можливості сучасних хмарних систем зберігання даних, в них не передбачено функцій для кросплатформного доступу до моделей деталей у САПР та інструментів для редагування їхніх параметрів відповідно до потреб користувачів. З огляду на особливості конструкційного проектування, запропоновано архітектуру системи зберігання даних, призначеної для дизайну деталей у САПР, на базі інтегрованої хмарної платформи. Враховано специфіку розгортання такої системи для гетерогенного програмного забезпечення різних САПР. У результаті аналізу розробленої системи визначено, що віртуалізація даних покращує динамічну адаптивність, централізує збережені ресурси та дозволяє не зосереджуватися на конкретних місцях їхнього розташування для користувачів. Таким чином, інтегроване управління пулом ресурсів забезпечує налаштування системи зберігання та переміщення даних без переривання роботи поточних програм, що сприяє розгортанню бібліотеки деталей на хмарі.

### Список використаних джерел

1. Рибак, О. В. (2026). Використання хмарних технологій при розробці програмного забезпечення розподілених систем. In *Proceedings of the 4th International Scientific and Practical Conference "Global Trends in Science, Technology and Economy"* (pp. 113–116). <https://doi.org/10.70286/isu-14.01.2026.004>.
2. Vogels, W. (2009). Eventually consistent. *Communications of the ACM*, 52(1), 40–44. <https://doi.org/10.1145/1435417.1435432>.
3. Tella, K. (2024). From storage to insights: architecting a serverless data lake on AWS with S3 and redshift. *International Journal of Innovative Research of Science, Engineering and Technology*, 13(10), 17867–17868. <https://doi.org/10.15680/IJRSET.2024.1310116>.

4. Calder, B., Wang, J., Ogus, A., Nilakantan, N., Skjolsvold, A., McKelvie, S., et al. (2011). Windows Azure Storage: a highly available cloud storage service with strong consistency. *SOSP'11 - Proceedings of the 23rd ACM Symposium on Operating Systems Principles*, 143–157. <https://doi.org/10.1145/2043556.2043571>.

5. Yang, T., Pollen, S., Uysal, M., Merchant, A., Wolfmeister, H., & Khalid, J. (2023). CacheSack: Theory and experience of Google's admission optimization for datacenter flash caches. *ACM Transactions on Storage*, 19(2), 1–24. <https://doi.org/10.1145/3582014>.

6. Tigariiev, V., Lopakov, O., Rybak, O., Kosmachevskiy, V., & Cioată, V. (2023). Design in modern information systems by applying cloud technologies. *Journal of Engineering Sciences*, 10 (1), 8–13. [https://doi.org/10.21272/jes.2023.10\(1\).e2](https://doi.org/10.21272/jes.2023.10(1).e2).

7. Gibson, I., Rosen, D., & Stucker, B. (2015). *Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing*. 2nd ed. Springer, New York. <https://doi.org/10.1007/978-1-4939-2113-3>.

### References

1. Рибак, О. В. (2026). Використання хмарних технологій при розробці програмного забезпечення розподілених систем. In *Proceedings of the 4th International Scientific and Practical Conference "Global Trends in Science, Technology and Economy"* (pp. 113–116). <https://doi.org/10.70286/isu-14.01.2026.004>.

2. Vogels, W. (2009). Eventually consistent. *Communications of the ACM*, 52(1), 40–44. <https://doi.org/10.1145/1435417.1435432>.

3. Tella, K. (2024). From storage to insights: architecting a serverless data lake on AWS with S3 and redshift. *International Journal of Innovative Research of Science, Engineering and Technology*, 13(10), 17867–17868. <https://doi.org/10.15680/IJRSET.2024.1310116>.

4. Calder, B., Wang, J., Ogus, A., Nilakantan, N., Skjolsvold, A., McKelvie, S., et al. (2011). Windows Azure Storage: a highly available cloud storage service with strong consistency. *SOSP'11 - Proceedings of the 23rd ACM Symposium on Operating Systems Principles*, 143–157. <https://doi.org/10.1145/2043556.2043571>.

5. Yang, T., Pollen, S., Uysal, M., Merchant, A., Wolfmeister, H., & Khalid, J. (2023). CacheSack: Theory and experience of Google's admission optimization for datacenter flash caches. *ACM Transactions on Storage*, 19(2), 1–24. <https://doi.org/10.1145/3582014>.

6. Tigariiev, V., Lopakov, O., Rybak, O., Kosmachevskiy, V., & Cioată, V. (2023). Design in modern information systems by applying cloud technologies. *Journal of Engineering Sciences*, 10 (1), 8–13. [https://doi.org/10.21272/jes.2023.10\(1\).e2](https://doi.org/10.21272/jes.2023.10(1).e2).

7. Gibson, I., Rosen, D., & Stucker, B. (2015). *Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing*. 2nd ed. Springer, New York. <https://doi.org/10.1007/978-1-4939-2113-3>.

Дата першого надходження статті до видання: 02.03.2026

Дата прийняття статті до друку після рецензування: 19.03.2026

UDC 004.75

### **Olha Rybak**

PhD in Technical Sciences

Associate Professor of the Department of Information Technologies and Design  
Odesa Polytechnic National University (Odesa, Ukraine)

E-mail: [olga.vol.rybak@gmail.com](mailto:olga.vol.rybak@gmail.com). ORCID: <http://orcid.org/0000-0002-0250-3037>

SCOPUS Author ID: 57208656221

### **DESIGNING A DATA STORAGE SYSTEM FOR CAD BASED ON CLOUD INFRASTRUCTURE**

*Advancement of structural design software with integrated cloud platform opens up vast opportunities due to upgraded tools, CAD independence and unrestricted access to models and technical documentation that simplifies product design and engineering. Despite the increased capabilities of modern cloud data storage systems, they practically lack tools for cross-platform access to part models in various file formats and options for making changes to them according to the input data from*

## TECHNICAL SCIENCES AND TECHNOLOGIES

remote users. This research considers architecture of a data storage system intended for parts designing in CAD based on a cloud. Features of deploying such a system for heterogeneous software of various CAD are also investigated. There are two approaches to organizing a cloud storage system: distributed resource integration and integrated resource distribution. The PaaS concept, built into the functional core adapters, integrates the provider with external applications and reconciles the contradictions between these two approaches by exchanging data and services in a common module over an interoperable connection channel. Structured information stored in the system database and unstructured information recorded as files in the cloud is combined from distributed and heterogeneous resources at the integration level into unified set of part data and deployed on the cloud. The main functions of program are implemented on the server, and user computers require deployment of an adapter to access local programs with instrument settings. Through the functional kernel adapter, the host of the system can access all services by importing the provided tools into its own frameworks. Thus, integrated resource pool management enables storage system configuration and data movement without interrupting current applications, facilitating the deployment of parts library on the cloud.

**Keywords:** cloud technologies; data storage system; CAD system; platform-as-a-service; cloud manufacturing.

*Fig.: 3. References: 7.*