

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки та програмної інженерії
Кафедра інженерії програмного забезпечення

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

Катерина НЕСТЕРЕНКО
“ ____ ” _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
МАГІСТРА

Тема: “Методологічні засади проектування автоматизованих систем на базі IPaaS”

Виконавець: Тарасевич Тетяна Олександрівна

Керівник: к.т.н., доцент Шибицька Наталія Миколаївна

Нормоконтролер: к.ф.-м.н. Гололобов Дмитро Олександрович

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки та програмної інженерії

Кафедра інженерії програмного забезпечення

Освітній ступінь магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Програмне забезпечення систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Катерина НЕСТЕРЕНКО

" ___ " _____ 2023 р

ЗАВДАННЯ

на виконання кваліфікаційної роботи студентки

Тарасевич Тетяни Олександрівни

1. Тема кваліфікаційної роботи: «Методологічні засади проектування автоматизованих систем на базі IPaaS»
затверджена наказом ректора від 29.09.2023 р. № 1994/ст.
2. Термін виконання проекту: з 02.10.2023 р. по 31.12.2023 р.
3. Вихідні дані до роботи: інтеграційне рішення служб Trello та Gmail в платформі MS Power Automate.
4. Зміст пояснювальної записки:
 1. Теоретичні аспекти проектування автоматизованих систем.
 2. Інтеграція технології iPaaS у проектування автоматизованих систем.
 3. Методологія проектування автоматизованих систем на базі iPaaS.
 4. Інтеграція систем з використанням MS Power Automate.
5. Перелік обов'язкових слайдів презентації:
 1. Підходи до синхронізації даних.
 2. Програмні технології та протоколи до інтеграції даних.
 3. Схема інтеграції з використанням iPaaS платформ.
 4. Приклади інтеграційних платформ.
 5. Інтеграція служб Trello та Gmail з використанням Power Automate.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Розробка та затвердження графіка роботи. Постановка задачі та вивчення літератури.	02.10.2023-05.10.2023	
2.	Написання реферату, списку скорочень, змісту, вступу.	06.10.2023-09.10.2023	
3.	Написання першого розділу. Проходження перевірки від керівника.	10.10.2023-18.10.2023	
4.	Написання другого розділу. Проходження перевірки від керівника.	19.10.2023-31.10.2023	
5.	Написання третього розділу. Проходження перевірки від керівника.	01.11.2023-14.11.2023	
6.	Написання четвертого розділу. Проходження перевірки від керівника.	14.11.2023-27.11.2023	
7.	Редагування пояснювальної записки, графічного матеріалу.	28.11.2023-10.12.2023	
8.	Проходження антиплагіату. Проходження нормо-контролю.	11.12.2023-15.12.2023	
9.	Перепліт пояснювальної записки. Отримання відгуку керівника. Підготовка презентації та тексту доповіді.	11.12.2023-15.12.2023	
10.	Передзахист кваліфікаційної роботи. Отримання рецензії.	17.12.2023-20.12.2023	
11.	Здача секретарю ДЕК: ПЗ, ГМ, CD-R з електронними версіями ПЗ, ГМ, презентацію, відгук керівника, рецензію, довідку про успішність, 2 папки, 2 конверта)	22.12.2023	
12.	Захист кваліфікаційної роботи перед ЕК	27.12.2023-28.12.2023	

Дата видачі завдання 02.10.2023 р.

Керівник кваліфікаційної роботи:

Завдання прийняла до виконання:

к.т.н., доцент Наталія ШИБИЦЬКА

Тетяна ТАРАСЕВИЧ

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Методологічні засади проектування автоматизованих систем на базі IPaaS»: 90 сторінок, 27 рисунків, 3 таблиці, 15 використаних джерел.

ІНТЕГРАЦІЯ, ХМАРНІ ТЕХНОЛОГІЇ, IPaaS, БЕЗПЕКА ПЕРЕДАЧІ ДАНИХ, КОНЕКТОРИ, ПРОЕКТУВАННЯ ПРОГРАМНИХ СИСТЕМ, МЕТОДОЛОГІЯ, АВТОМАТИЗАЦІЯ ПРОЦЕСІВ.

Об'єкт дослідження – процес проектування автоматизованих систем з використанням рішень орієнтованих на використання IPaaS технологій.

Мета кваліфікаційної роботи – створення практично орієнтованих методологічних засад і принципів, що лежать в основі проектування автоматизованих систем на базі IPaaS, для розробки ефективних стратегій та підходів до інтеграції, оптимізації використання IPaaS технологій у сучасних вимогах бізнесу.

Метод дослідження – впровадження інтеграції між двома службами з використанням IPaaS рішень для розробки загального огляду можливостей інтеграційних платформ.

В процесі роботи було здійснено аналіз існуючих ручних підходів до інтеграції систем та програм та було запропоновано альтернативне автоматизоване рішення з використанням IPaaS платформ. В результаті було визначено, що IPaaS рішення не потребують залучення технічного персоналу для автоматизації процесів інтеграції.

Результати роботи можуть бути використані як основа для автоматизації інтеграції даних між службами Trello та Gmail, створення звітів та автоматизації контролю над станом дошок.

Розробка та дослідження проводилися під управлінням браузеру Google Chrome та вбудованих інструментів та конструктора платформи Power Automate.

ABSTRACT

Explanatory note to the thesis "Methodological principles of designing automated systems based on iPaaS": 90 pages, 27 figures, 3 tables, 15 sources used.

INTEGRATION, CLOUD TECHNOLOGIES, IPAAS, SECURITY OF DATA TRANSMISSION, CONNECTORS, DESIGN OF SOFTWARE SYSTEMS, METHODOLOGY, AUTOMATION OF PROCESSES.

The object of research is the process of designing automated systems using solutions oriented to the use of iPaaS technologies.

The purpose of the thesis is creation practically oriented methodological foundations and principles underlying the design of automated systems based on iPaaS, for the development of effective strategies and approaches to integration, optimizing the use of iPaaS technologies in modern business requirements.

The research method is the introduction of integration between two services using iPaaS solutions to develop an overview of the capabilities of integration platforms.

In the process the existing manual approaches to the integration of systems and programs were analyzed and an alternative automated solution using iPaaS platforms was proposed. As a result, it was determined that iPaaS solutions do not require the involvement of technical personnel to automate integration processes.

The results of the work can be used as a basis for automating the integration of data from Trello and Gmail services, creating reports and automating control over the status of boards.

Development and research was conducted using the Google Chrome browser and the built-in tools and builder of the Power Automate platform.

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ ПРОЕКТУВАННЯ АВТОМАТИЗОВАНИХ СИСТЕМ	11
1.1. Підходи синхронізації даних	11
1.2. Технології та протоколи для комунікації між програмними системами	17
1.3. iPaaS як альтернатива програмних інтеграцій	20
Висновок	24
РОЗДІЛ 2 ІНТЕГРАЦІЯ ТЕХНОЛОГІЇ IPAAS У ПРОЕКТУВАННЯ АВТОМАТИЗОВАНИХ СИСТЕМ	26
2.1. Загальний огляд технології iPaaS	26
2.2. Вплив iPaaS на архітектуру систем	31
2.3. Переваги та обмеження використання iPaaS.....	35
Висновок	40
РОЗДІЛ 3 МЕТОДОЛОГІЯ ПРОЕКТУВАННЯ АВТОМАТИЗОВАНИХ СИСТЕМ НА БАЗІ IPAAS	42
3.1. Етапи проектування автоматизованих систем	43
3.2. Постачальники та інструменти iPaaS.....	46
3.3. Правила забезпечення безпеки та надійності використання iPaaS рішень.....	63
Висновки	67
РОЗДІЛ 4 ІНТЕГРАЦІЯ СИСТЕМ З ВИКОРИСТАННЯМ MS POWER AUTOMATE	69
4.1. Загальний огляд Microsoft Power Automate платформи	69
4.2. Побудова інтеграції Trello та Gmail за допомогою Power Automate	74
Висновки	86
ВИСНОВОК.....	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	90

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

ПЗ — програмне забезпечення

API (англ. Application Programming Interface) — прикладний програмний інтерфейс

iPaaS (англ. Integration platform as a service) — інтеграційна платформа як послуга

SaaS (англ. Software as a Service) — програмне забезпечення як послуга

ESB (англ. Enterprise Service Bus) — інтеграційна шина даних

DI (англ. Data Integration) — інтеграція даних

RPA (англ. Robotic Process Automation) — роботизовані процеси

БД — база даних

ВСТУП

З розвитком комп'ютерних технологій все більшої актуальності набуває питання інтеграції даних. Це безпосередньо пов'язано з тим, що життя та бізнес стали надзвичайно залежними від потоку інформації. Оскільки обсяг та різноманітність даних, що створюються та зберігаються у різних системах, постійно збільшується, з'являється виразна потреба в ефективному їх обміні та зберіганні. Ці проблеми вирішуються завдяки процесам інтеграції даних. Інтеграція даних (data integration) – це процес об'єднання та спільного використання інформації з різних джерел в одній або різних системах. Головною метою інтеграції даних є процес подання даних в зрозумілому, доступному та ефективному для аналізу та прийняття рішень.

З інтеграцією даних пов'язані питання забезпечення захисту приватності та безпеки інформації, що стають особливо актуальними у зв'язку з сучасними вимогами до конфіденційності інформації.

Зважаючи на те, що дані можуть бути збережені в різних системах, форматах, базах даних та місцях, інтеграція даних допомагає вирішувати проблему фрагментації та розсіювання інформації. Також різні типи користувачів мають різні потреби щодо доступу до даних, тому data integration допомагає забезпечити доступ до необхідних даних для кожної групи в потрібному представленні.

З кожним днем невпинно зростає обсяг даних, що обробляються різними системами. Тому зі збільшенням кількості інформації стає важливим прийняття рішень на основі об'єктивних фактів та аналізу. Інтеграція даних забезпечує прийняття більш ефективних та оптимальних рішень у роботі з даними.

Основні аспекти інтеграції даних включають:

- Збір інформації з різних джерел, таких як бази даних, текстові файли, журнали, веб-сервіси та інші джерела.
- Трансформація даних для подання їх в необхідному вигляді або структурі, адже інформація з різних джерел може мати різний формат чи кодування.

- Зберігання даних у спільному сховищі, щоб забезпечити доступ їх подальшого використання.
- Синхронізація даних з джерелами для забезпечення актуальності даних.
- Доступ до даних.
- Автоматизація бізнес-процесів та взаємодії між різними системами.
- Безпека та захист даних, контроль доступу до них.

Інтеграція даних допомагає організаціям зменшити дублювання та копіювання інформації, підвищує точність та доступність даних, зменшує ризики при роботі з інформацією.

Станом на сьогодні, компаніям необхідно мати рішення для інтеграції, які можуть ефективно охоплювати постійно зростаючі вимоги до інтеграції між різними екосистемами, додатками та даними. Це важливо, щоб залишатися конкурентоспроможним, задовольняти попит клієнтів і підвищити гнучкість та масштабованість компанії. Тому все більше організацій прагнуть використати великий потенціал інтеграційних можливостей, які пропонує нове поле на ринку додатків і проміжного програмного забезпечення – iPaaS (Integration platform as a service – інтеграційна платформа як послуга). Тобто iPaaS – це інтеграційна платформа, яка надає середовище та засоби для побудови та управління зв'язками між різними системами, програмними додатками, тощо.

Актуальність теми визначається декількома ключовими причинами, чому потрібні систематизація та оптимізація процесів інтеграції. Перш за все необхідність в ефективній інтеграції між різними системами стає ключовим завданням для забезпечення безперебійності діяльності програмних засобів. А також хмарні технології посідають все більш поширеною технологією у розробці, де iPaaS платформи виступають як ключові складові хмарної інфраструктури, адже сприяють зручній інтеграції та обміну даними між різними локальними та хмарними системами.

Мета досліджень полягає в аналізі та створенні практично орієнтованих методологічних засад і принципів, що лежать в основі проектування автоматизованих систем на базі iPaaS, з метою розробки ефективних стратегій та підходів до інтеграції, оптимізації використання iPaaS технологій у сучасних вимогах бізнесу, а також забезпечення безпеки та надійності; розкритті можливостей, що надає використання технології iPaaS.

Завдання дослідження включає аналіз існуючих методів проектування систем, розкриття ключових понять, що стосуються iPaaS платформ, визначення їх переваг та недоліків у розрізі методології проектування, аналіз потенційних проблем використання інтеграційних платформ, формування конкретних рекомендацій щодо ефективного використання iPaaS під час проектування автоматизованих програмних засобів.

Предметом дослідження є методологічні аспекти, підходи та принципи, що визначають ефективність, надійність та необхідність у проектуванні автоматизованих систем на базі iPaaS. Це включає вивчення конкретних методів та кращих практик, які використовуються під час розробки систем з використанням iPaaS технологій.

Об'єктом дослідження є автоматизовані системи, що використовують iPaaS рішення.

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ ПРОЕКТУВАННЯ АВТОМАТИЗОВАНИХ СИСТЕМ

Розробка та проектування програмних систем є ключовими процесами, що забезпечують ефективність ПЗ. Це призводить до гострої необхідності у розумінні та аналізі теоретичних аспектів та основних концепцій процесу проектування автоматизованих систем. Необхідність у синхронізації різних програм та систем стає невід'ємних процес роботи ПЗ.

Синхронізація даних здатна забезпечити наступні функції систем:

- узгодженість даних, що забезпечує ідентичність даних у різних місцях чи системах;
- перевірка та очищення даних, коли вони надходять в систему, щоб унеможливити дублювання інформації;
- цілісність даних: модифікації даних застосовуються до всіх систем у режимі реального часу, щоб підтримувати точність запису даних.

1.1. Підходи синхронізації даних

Інтеграція систем – це процес об'єднання різних компонентів однієї системи або різних програмних систем для того, щоб вони обмінювалися даними за принципами роботи єдиної системи. Для виконання інтеграційної функції застосовуються різні підходи. Нижче представлені найпоширеніші підходи до синхронізації даних [1]:

1. Інтеграція пункт-до-пункту (Point-to-Point).

Цей підхід використовується для створення прямих з'єднань між двома системами, які обмінюються даними між собою. Кожен такий зв'язок налаштовується окремо. Такий тип інтеграції може бути ефективним, якщо обмін даними відбувається між двома системами, проте він стає складно керованим та масштабованим, коли кількість систем для обміну інформацією зростає. Схема

комунікації між системами за принципом Point-to-Point зображено на рисунку 1.1 [6].

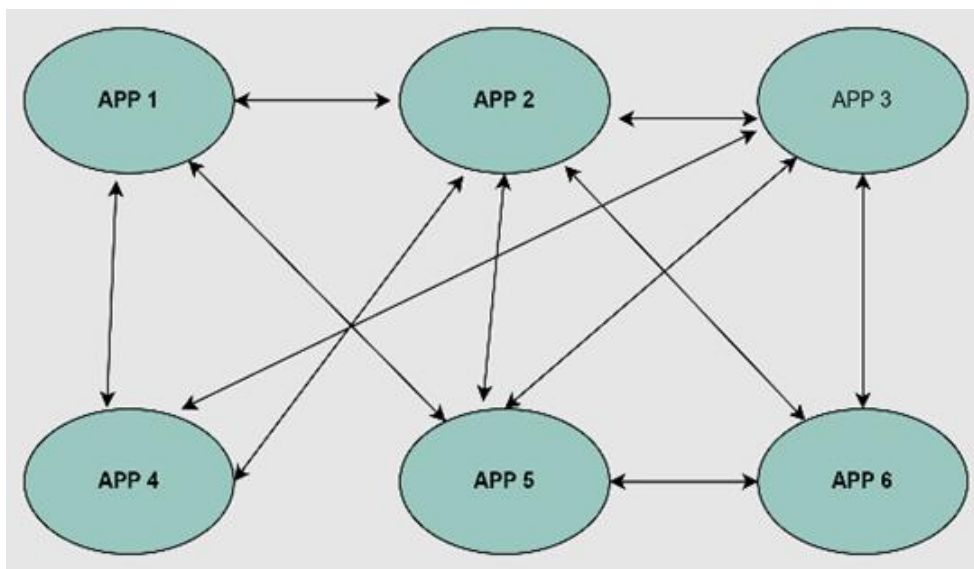


Рис. 1.1. Взаємодія систем за принципом Point-to-Point

2. Централізована (Centralized) інтеграція.

Цей підхід передбачає використання централізованої системи, яка функціонує як центральна точка для всіх інших систем. Це означає, що всі системи комунікують з спільною (центральною) системою, що відповідає за обробку, розподіл та передачу даних. Централізована інтеграція може спрощувати керування та підтримку, проте підвищуються ризики повної непрацездатності всіх систем, у разі якщо центральна виходить із ладу.

3. Інтеграція через Middleware.

Це підхід до об'єднання різних програмних систем або компонентів, за допомогою проміжного програмного забезпечення, Middleware, що працює посередником між різними системами та додатками, дозволяє їх обмінюватися інформацією. Middleware приймає дані від одного додатку, обробляє їх, якщо це необхідно, і передає їх до іншого додатку. Цей процес зображено на рисунку 1.2. Він може виконувати перетворення даних різного формату, забезпечуючи відповідність вимогам інших систем. Middleware може підтримувати різні комунікаційні протоколи, такі як HTTP, MQTT, SOAP, REST, і багато інших, що дозволяє інтегрувати різні системи, незалежно від того, які протоколи вони використовують.

Цей підхід надає засоби для безпечного обміну даними між системами, використовуючи авторизацію, автентифікацію, шифрування та інші механізми. Також Middleware може бути масштабованим для обробки великих обсягів даних і великої кількості запитів [6].

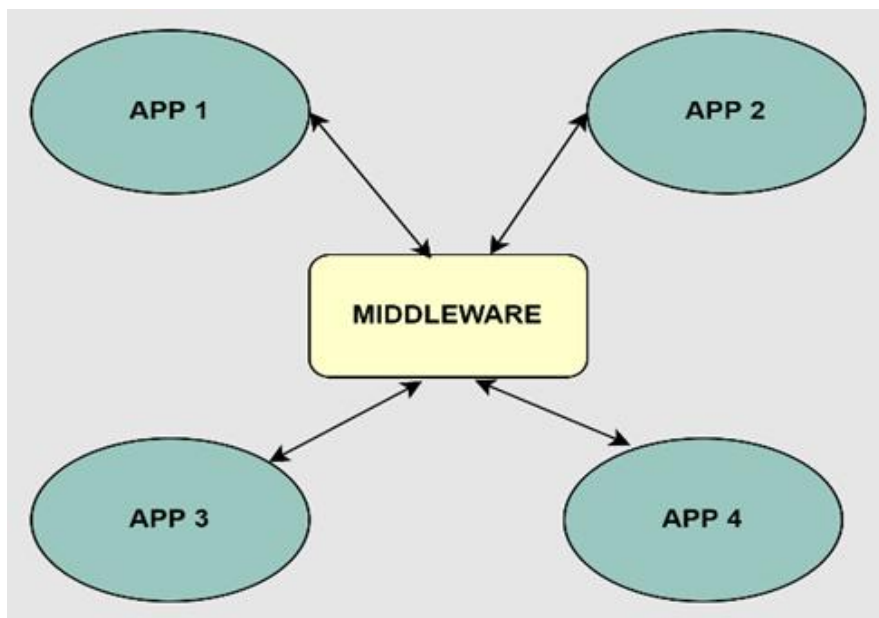


Рис. 1. 2. Взаємодія систем за принципом Middleware

Підхід інтеграції через Middleware особливо корисний у великих організаціях, де існують різні додатки, системи та сервіси, які потребують спільної роботи. Middleware спрощує цей процес, допомагаючи забезпечити гладку та надійну інтеграцію між різними компонентами і зменшуючи складність взаємодії між ними.

4. Extract, Transform, and Load (ETL) інтеграція.

Це процес збору непідготовлених даних із окремих джерел, передачі у проміжну базу даних для перетворення та завантаження підготовлених даних у єдину цільову систему. Інструменти ETL використовуються для інтеграції даних, щоб задовольнити вимоги систем управління реляційними базами даних та/або традиційних сховищ даних з підтримкою OLAP (online analytical processing, аналітичної онлайн-обробки). Для інтеграції даних у великому масштабі ETL є найбільш можливим доступним варіантом. Програмне забезпечення ETL може витягти потрібні дані з вихідної системи, перетворити їх у потрібний формат, а

потім перемістити в систему призначення (процес інтеграції зображено на рисунку 1.3) [4]. Процес здебільшого автоматизований і керується картами даних для завершення. Ці карти даних дозволяють користувачам автоматизувати потоки даних.

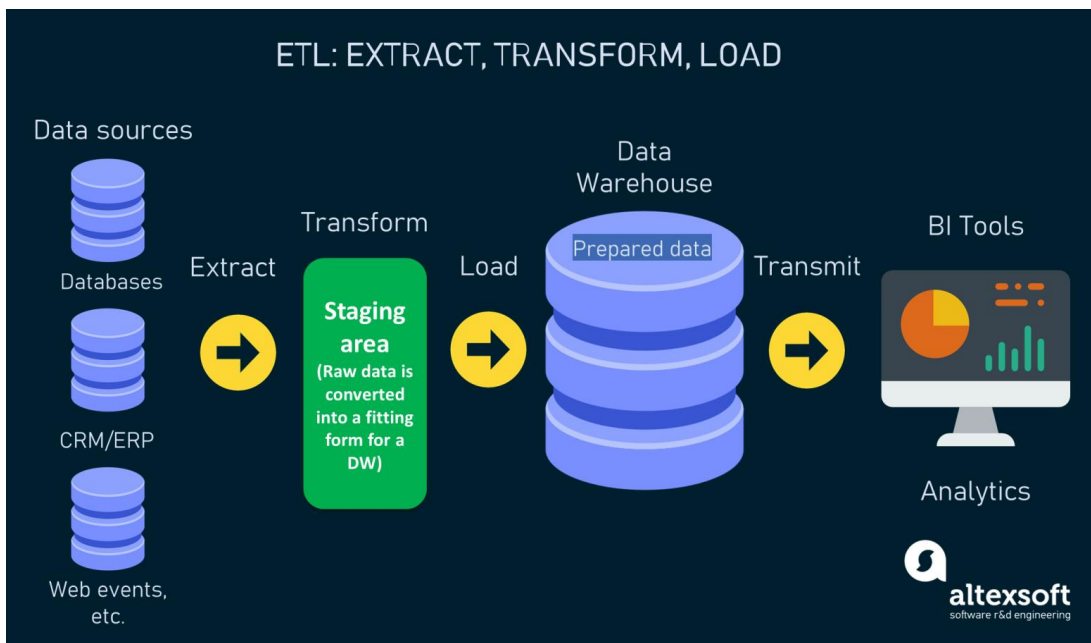


Рис. 1.3. Процес роботи ETL інтеграції

Тобто ETL процес включає такі етапи: вилучення, трансформація, завантаження.

На етапі Extract (Вилучення) інформація вилучається з різних джерел (наприклад, баз даних, файлів, логів, веб-сервісів). Це може відбуватися у реальному часі або після збирання великого пакету даних, залежно від потреб інтеграції.

Далі слідує процес Transform (Трансформація), де дані піддаються обробці та трансформації. Це може включати фільтрацію, очищення, об'єднання, агрегацію, перетворення форматів та інші операції, які забезпечують відповідність структури та якості даних потребам системи.

Останнім етапом визначають процес Load (Завантаження). Після обробки даних, вони завантажуються у централізовану базу даних або сховище даних. Цей

етап може включати додавання нових записів, оновлення існуючих даних та видалення застарілих записів.

5. Extract, Load, and Transform (ELT) інтеграція.

Це підхід до обробки та інтеграції даних, у якому послідовність операцій дещо відмінна від попереднього підходу ETL, що є зручнішими при роботі з великими обсягами даних. Основними етапами цього підходу є етапи підходу ETL, проте в іншому порядку. Процес роботи ELT підходу зображено на рисунку 1.4 [4].

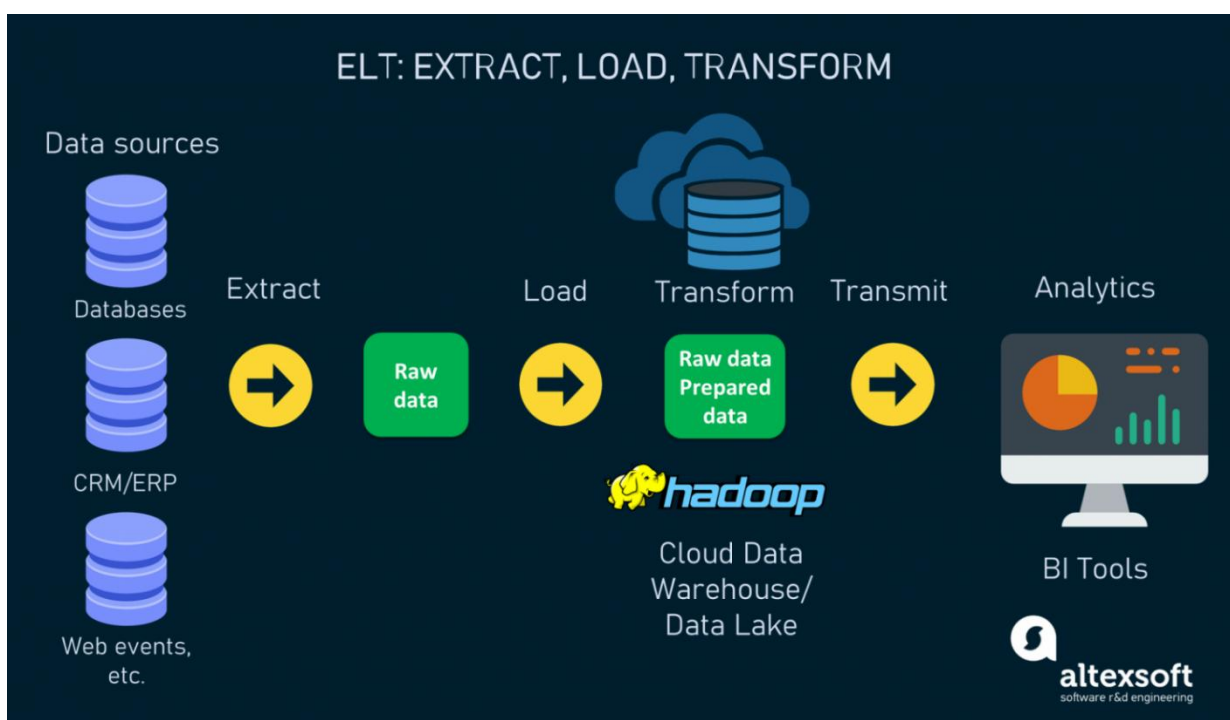


Рис. 1.4. Процес роботи ELT інтеграції

Першим етапом йде Extract (Вилучення). Важливою особливістю ELT є те, що дані не трансформуються в цьому етапі, а вилучаються "як є", зберігаючи свою структуру та формат.

Після Вилучення даних відбувається їх Load (Завантаження). Вилучені дані завантажуються безпосередньо в необхідне сховище даних, це можуть бути хмарні сховища (наприклад, Google Cloud Storage, Amazon S3) або data lake. Особливістю є

те, що ці дані завантажуються без обробки та перетворень, що дозволяє зберігати оригінальну структуру даних для їх подальшого аналізу та обробки.

Завершуючим кроком є Transform (Трансформація) даних, який відбувається після Завантаження (на відміну від ETL підходу). На цьому етапі відбувається перетворення, очищення, поєднання або обробка даних за допомогою різних інструментів та технологій.

Основними перевагами інтеграції ELT є: швидкість (оскільки вилучення та завантаження даних відбуваються без зайвої трансформації), скорочення витрат (оскільки потреби у додаткових ресурсах інфраструктури та обчислювальних потужностях для роботи з даними є значно меншими), збереження оригінальних даних, масштабованість (це забезпечується тим, що обробка даних та їх трансформація можуть виконуватися паралельно у хмарних середовищах). Проте важливо враховувати, що інтеграція ELT найкраще підходить для аналітичних завдань, де необхідно зберегти оригінальні дані для подальшого дослідження.

6. Імпорт/Експорт даних.

Це простий підхід до інтеграції даних. У цьому підході дані експортуються з основного джерела даних, як правило, у формі CSV або файлу Excel, а потім завантажуються до місця призначення. Проте у цього підходу є дві головні проблеми:

- відсутність трансформацій (усі дані вручну додаються до центрального репозиторію, при необхідності внесення будь-яких змін до даних, їх доведеться редагувати вручну);
- повільна робота (мануальний експорт займає багато часу, ще більшою проблемою є завантаження даних у сховище даних у сумісному форматі).

Вибір підходу до інтеграції систем залежить від конкретних потреб, архітектури систем, бюджету та інших факторів. Зазвичай вирішенням є поєднання різних підходів для вирішення конкретних завдань інтеграції.

Проте всі ці підходи можуть бути реалізовані, використовуючи різні протоколи для взаємодії систем.

1.2. Технології та протоколи для комунікації між програмними системами

Технології та протоколи для комунікації між програмними системами – це засоби та стандарти, що дозволяють різним системам та програмам обмінюватися даними, повідомленнями і запитам через мережу або на локальному рівні. Вони відіграють критичну роль у створенні інтегрованих, зручних у використанні та ефективних програмних рішень. З їх допомогою об'єднуються дані з різних джерел і систем, що в свою чергу створює єдину інформаційну картину.

Існує безліч технологій та протоколів для комунікації між програмними системами, і вибір конкретного методу залежить від потреб вашого проекту, характеристик додатків та вимог до безпеки та ефективності. Найбільш поширеними технологіями для комунікації систем є:

1. REST (Representational State Transfer) – це архітектурний стиль для проектування мережевих додатків, який використовує HTTP для комунікації. Він базується на ресурсах та операціях над ними, що дозволяє створювати прості та легко розширювані API. Це не такий протокол, як SOAP чи AMQP, а скоріше набір принципів і обмежень, які керують структурою веб-служб. REST зазвичай використовується для інтеграції даних через його простоту, масштабованість і легкість використання. Служби REST не мають стану, тобто кожен запит від клієнта до сервера повинен містити всю інформацію, необхідну для розуміння та обробки запиту. Це спрощує реалізацію на сервері та забезпечує горизонтальну масштабованість. REST є популярним вибором для інтеграції даних завдяки своїй простоті, універсальності та сумісності з веб-інфраструктурою. Він забезпечує ефективний спосіб обміну даними між системами та програмами, що робить його широко поширеним підходом у сучасних проектах розробки програмного забезпечення та інтеграції даних.

2. SOAP – це протокол для обміну структурованими повідомленнями в форматі XML. Він часто використовується для створення веб-сервісів, які надають функціональність, доступну через мережу.

3. GraphQL – це мова запитів та виконавче середовище, призначене для взаємодії з серверами додатків. Вона дозволяє клієнтам вказувати, які дані вони

хочуть отримати, і надає більшу гнучкість в порівнянні з традиційними REST API. На відміну від REST, який часто вимагає кількох кінцевих точок для різних ресурсів, GraphQL зазвичай надає одну кінцеву точку для всіх запитів даних. Це спрощує процес інтеграції та зменшує потребу в кількох викликах API. GraphQL дозволяє клієнтам запитувати лише ті конкретні дані, які їм потрібні, уникаючи надмірного чи недостатнього отримання даних. Це зменшує обсяг даних, що передаються через мережу, що забезпечує ефективнішу інтеграцію даних. Незважаючи на те, що GraphQL пропонує переваги з точки зору гнучкості та ефективності, він також може внести складність, і для цього потрібна реалізація на стороні сервера, здатна обробляти запити GraphQL. Цей протокол є переконливим вибором для інтеграції даних, коли ви хочете більше контролювати дані, які ви отримуєте, зменшити надмірну вибірку даних і спростити інтеграцію з різними джерелами даних, зберігаючи надійну типізацію та документацію.

4. WebSocket – це протокол для двосторонньої комунікації в реальному часі між клієнтом та сервером. Він підходить для додатків, які потребують миттєвої обміну даними, таким як чати та стрімінгові сервіси. WebSocket можна використовувати для надсилання push-повідомлень і сповіщень клієнтам. В інтеграції даних це може бути корисним для сповіщення клієнтів про важливі події або зміни в джерелі даних. WebSocket зазвичай вимагає підтримки на стороні сервера та клієнтських бібліотек для ефективною реалізації протоколу.

5. AMQP (Advanced Message Queuing Protocol) – це відкритий протокол для обміну повідомленнями між програмами або компонентами системи, що дозволяє реалізувати ефективну та надійну інтеграцію даних. AMQP був розроблений для вирішення завдань обробки повідомлень, передачі даних у реальному часі та побудови розподілених систем. AMQP реалізує асинхронний обмін повідомленнями між програмними компонентами [5]. Це дозволяє відправнику надсилати повідомлення, не чекаючи на відповідь, і отримувачу обробляти повідомлення у зручний для нього момент часу. Принцип роботи AMQP протоколу зображено на рисунку 1.5. AMQP базується на системі черг повідомлень. Відправлені повідомлення зберігаються в черзі і оброблюються отримувачами в порядку їхнього

надходження. Це допомагає забезпечити надійність обміну даними, оскільки дані не втрачаються навіть у разі збоїв в системі.

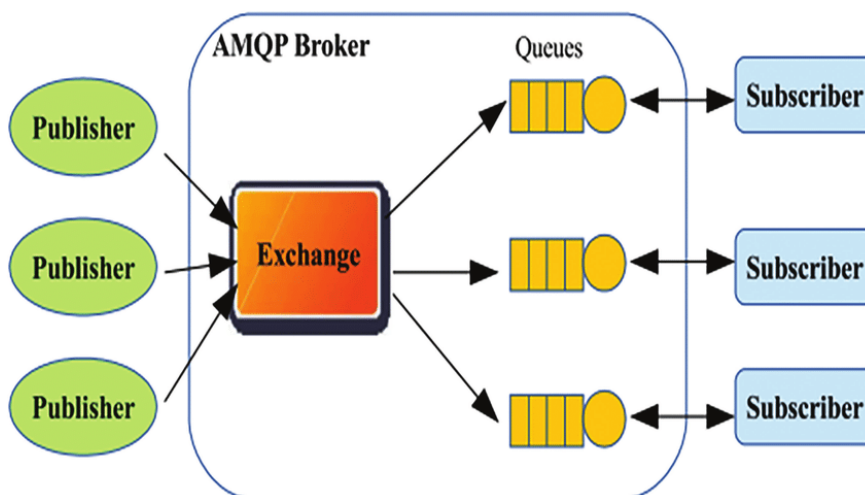


Рис. 1.5. Робота протоколу AMQP

6. Корпоративна службова шина (ESB) – один з ключових компонентів у сучасних інтеграційних архітектурах. Цей підхід сприяє безперебійному обміну даними та функціями між різними системами в межах підприємства. Він діє як проміжне програмне забезпечення, забезпечуючи уніфікований і стандартизований підхід до інтеграції. ПЗ маршрутизує повідомлення між програмами та службами, трансформує протоколи зв'язку, перетворює дані для забезпечення їхньої сумісності та забезпечує композицію та оркестровку служб. ESB зазвичай включає інструменти для моніторингу потоків повідомлень, відстеження продуктивності та керування загальним станом інтеграційного середовища. Рішення може підтримувати архітектури, керовані подіями, що дозволяє системам реагувати на події та зміни в реальному часі. Оскільки ESB є посередником між системами, він забезпечує централізовану обробку помилок і та надає можливості моніторингу, що підвищує надійність процесів інтеграції. Завдяки своїй здатності працювати з різними протоколами та форматами, ESB пропонує гнучкість у адаптації до нових технологій і бізнес-вимог.

Вибір конкретної технології залежить від потреб проекту, технічних обмежень і специфікацій додатків, які будуть інтегруватись.

Використання технологій та протоколів для комунікації між програмними системами може бути складним завданням і включає ряд потенційних викликів, наприклад:

- складність налаштування інфраструктури для взаємодії між програмними системами, особливо якщо система складається з багатьох компонентів та працює з різними технологіями;
- сумісність – системи програми мають бути сумісними за версіями і правильно налаштованими;
- забезпечення безпеки під час взаємодії між системами є важливим завданням (включає в себе шифрування даних, автентифікацію та авторизацію);
- моніторинг і управління, щоб відслідковувати стан інтеграцій та виявляти помилки;
- управління помилками і відмовами для забезпечення надійності інтеграції, також необхідно забезпечити резервне копіювання;
- тестування інтеграцій може бути складним процесом, оскільки складно тестувати взаємодію даних між різними системами.

Хоча використання технологій та протоколів для комунікації може бути складним завданням, правильно розроблена інфраструктура та стратегія можуть значно полегшити цей процес і забезпечити надійну та ефективну комунікацію між програмними системами.

Все ж незважаючи на значну кількість програмних технологій та протоколів для синхронізації даних, все більшої розповсюдженості набуває так званий iPaaS підхід. iPaaS (Integration Platform as a Service) є важливою альтернативою традиційним програмним інтеграціям. Вона надає інтерфейс та інструменти для спрощення та полегшення інтеграції даних і додатків у корпоративних середовищах.

1.3. iPaaS як альтернатива програмних інтеграцій

iPaaS став все більш популярним рішенням для інтеграції даних та систем в сучасних компаніях. Його популярність визначається кількома чинниками, які роблять його більш зручним і ефективним способом роботи з інтеграціями,

порівняно з традиційними програмними технологіями та протоколами. IPaaS надає готові коннектори для багатьох популярних додатків та сервісів, що спрощує підключення та інтеграцію цих систем без необхідності розробки власних рішень.

Архітектура iPaaS сприяє інтеграції всіх додатків, технологій і навіть систем у хмарне середовище, де хмара може бути як локальною, так і зовнішньою структурою (схема керування платформи інтеграції потоком даних між програмами та базами даних зображена на рисунку 1.6). iPaaS дозволяє розгортати процес інтеграції без будь-якого зовнішнього апаратного чи проміжного програмного забезпечення компанією, що займається програмним забезпеченням, або окремим користувачем. Архітектура інтеграційної платформи як послуги (iPaaS) розроблена для полегшення інтеграції різних програм, систем і джерел даних. Платформи iPaaS зазвичай використовують хмарну архітектуру, яка забезпечує масштабоване та гнучке середовище для створення, керування та моніторингу інтеграцій.

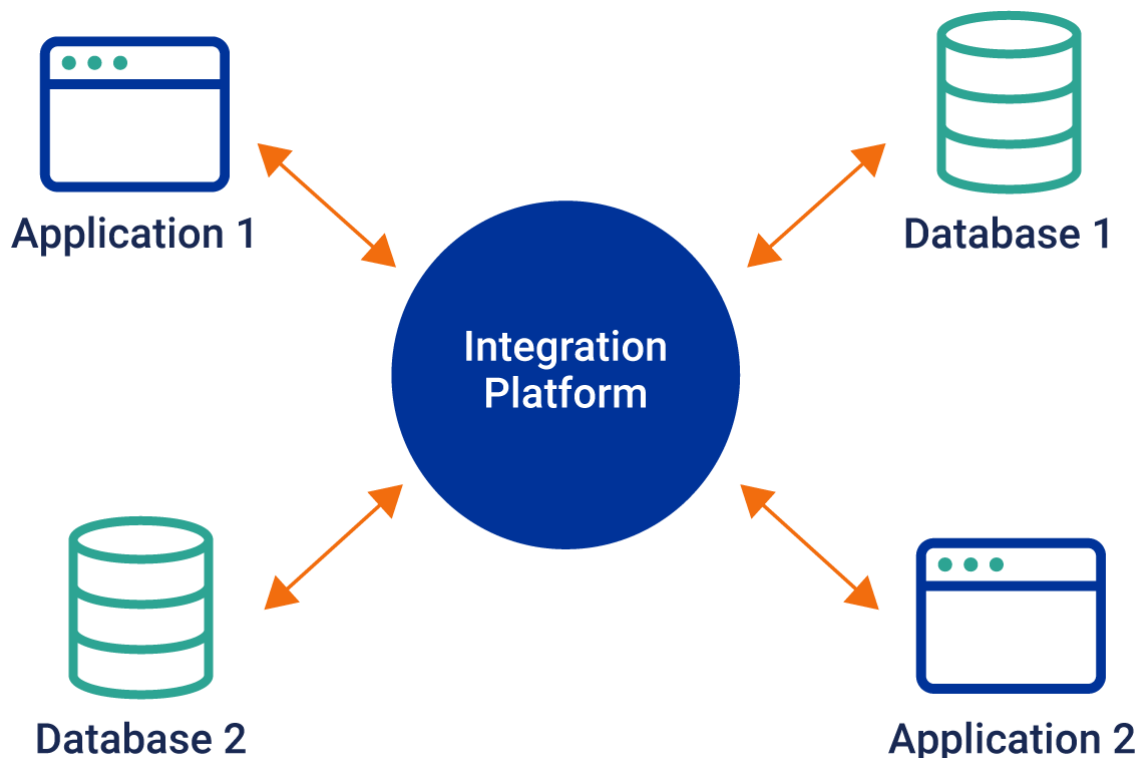


Рис. 1.6. Схема інтеграції iPaaS із застосунками та базами даних

Подібно до програмного забезпечення як послуги (SaaS), платформа інтеграції – це набір інструментів на основі підписки. Це означає, що потрібно підписатися на програмну платформу, яка надає інфраструктуру для калібрування та розгортання протоколів з'єднання між бізнес-додатками. Це, по суті, попередньо встановлені параметри зв'язку між програмами, які можна вважати інтерфейсами прикладного програмування (API). Ці API схожі на правила, яких дотримуються програми, наприклад, які програми можуть спілкуватися одна з одною та що відбувається з даними, коли вони досягають місця призначення. API визначають, як програми спілкуються та взаємодіють одна з одною. Тому інтеграційні платформи є більш ефективними та безпечними, ніж ручна інтеграція. Ця екосистема схожа на центральний центр керування, що дозволяє переглядати та змінювати інтеграції. Наприклад, ви можете налаштувати інтеграцію даних, автоматизувати потоки інтеграції та обмінюватися інформацією з пристроями [8].

Тобто різні хмарні системи та SaaS застосунки можливо інтегрувати між собою за допомогою iPaaS, що в свою чергу абстрагує користувача від написання коду і дозволяє сконцентруватись на реалізації бізнес-логіки. Зазвичай, iPaaS надає зручний користувацький інтерфейс для побудови різних процесів обробки та передачі даних. iPaaS є зв'язуючим елементом між різноманітними автоматизованими системами та порталами в хмарному середовищі [7].

iPaaS дозволяє вирішувати різноманітні бізнес-задачі, пов'язані з інтеграцією даних та систем, наприклад:

- Інтеграція додатків: iPaaS дозволяє підключати різні додатки та сервіси разом, незалежно від того, чи вони розташовані в хмарі чи на місцевому сервері. Це робить можливим обмін даними та спільну роботу між різними системами.
- Обмін даними: iPaaS допомагає забезпечити стабільний обмін даними між різними джерелами даних і цільовими системами, забезпечуючи автоматичну передачу даних.
- Синхронізація даних: iPaaS дозволяє забезпечити консистентність даних між різними джерелами.
- Моніторинг і аналітика: iPaaS надає засоби для моніторингу та аналізу

інтеграційних процесів. Надає можливість відстежувати продуктивність, виявляти проблеми та оптимізувати інтеграції на підставі зібраних даних.

- **Забезпечення безпеки:** iPaaS забезпечує захист даних під час їх транспортування і зберігання, використовуючи шифрування, аутентифікацію та контроль доступу. Це допомагає запобігти витоку даних та іншим загрозам для безпеки.

- **Сприяння інноваціям:** iPaaS дозволяє швидко інтегрувати нові технології та додатки у ваш існуючий стек, що допомагає вдосконалювати бізнес-процеси та забезпечувати конкурентні переваги.

- **Зменшення витрат і скорочення часу виходу на ринок:** Використання iPaaS може зменшити витрати на інфраструктуру та розробку, а також дозволить швидко розгорнути нові інтеграції і продукти на ринку.

iPaaS платформи зазвичай адаптовані для інтеграції додатків практично без кодування, тому вони вимагають невеликих попередніх знань програмування. Low-code — це підхід до розробки програмного забезпечення, спрямований на створення IT-додатків із невеликим кодуванням, а в крайніх випадках навіть без кодування (платформи без коду). Іншими словами, мета полягає в тому, щоб уникнути звичайного ручного програмування, наскільки це можливо. Зазвичай це досягається шляхом забезпечення цих платформ графічним інтерфейсом користувача та утилітами перетягування, які дозволяють нам «програмувати» практично без написання коду.

iPaaS може застосовуватися для широкого спектру корпоративних інтеграційних потреб на різних платформах і джерелах даних. Наприклад для інтеграції:

- від програми до програми, які розташовуються в різних середовищах;
- мікросервісів;
- B2B із різноманітними сторонніми партнерами та постачальниками без потреби в спільному API чи користувацьких інтеграціях;
- даних;
- платформ.

Загалом, iPaaS створює фундамент для ефективної інтеграції даних та систем, що допомагає організаціям досягти своїх бізнес-цілей, підвищити продуктивність та покращити взаємодію з клієнтами та партнерами.

Висновок

Інтеграція даних стає все більш складним процесом, тому кожен підхід до неї має право на існування залежно від задач та вимог, які треба вирішити. Необхідно ретельно зважувати альтернативи перш ніж обрати конкретний підхід до інтеграції в системі. В останні роки iPaaS платформи набирають обертів у IT середовищі завдяки хмарному розгортанню, гнучкості та доступності.

iPaaS призвів до значної цифрової трансформації. Ця цифрова зміна була неминучою, оскільки виникала необхідність співпрацювати на різних платформах. Тому iPaaS відіграла ключову роль у наданні допомоги в інтеграції різних додатків, процесів, програм і даних, тим самим зменшуючи складність об'єднання платформ. Таким чином, iPaaS є однією з найбільш популярних тем хмарних обчислень. З огляду на схильність ринку до застосування інтеграції в реальному часі, мобільності та електронної комерції, можна очікувати, що ринок iPaaS буде тільки зростати в найближчі роки.

Завдяки широкому застосуванню хмарних технологій компаніями iPaaS став невід'ємним компонентом технології хмарних обчислень. iPaaS разом з іншими хмарними програмами дозволив організаціям вирішувати нові проблеми, такі як кросплатформна інтеграція. Платформи iPaaS стандартизовано щодо того, як додатки мають бути інтегровані в організації, що допомогло автоматизувати бізнес-процеси та забезпечило обмін даними між платформами з більшою гнучкістю, ніж застарілі системи. Компанії, які використовують технології iPaaS, можуть скоротити ручну роботу та зменшити залежність від електронних листів і електронних таблиць. iPaaS допомагає організаціям забезпечити повну видимість свого бізнесу, підвищує швидкість їх операцій і забезпечує підвищену точність на всіх розгорнутих платформах [7].

Оскільки зазвичай глибоко володіють бізнес-процесами люди без технічних навичок, low-code та no-code платформи є зручними у використанні для написання інтеграцій. iPaaS з низьким кодом поєднує в собі переваги традиційного iPaaS із доступністю розробки з низьким кодом. Це дозволяє організаціям пришвидшувати інтеграційні проекти, покращувати співпрацю між технічними та нетехнічними командами та швидше реагувати на мінливі потреби бізнесу. Цей підхід є особливо цінним у сучасному швидкоплинному бізнес-середовищі, де швидка інтеграція часто має вирішальне значення для збереження конкурентоспроможності.

Архітектура iPaaS надає комплексне рішення для організацій, які прагнуть оптимізувати свої зусилля з інтеграції даних. Він абстрагує велику частину технічної складності, дозволяючи користувачам зосередитися на розробці, реалізації та керуванні інтеграціями ефективно та результативно. Хмарний характер iPaaS забезпечує гнучкість, масштабованість і доступність, що робить його потужним інструментом для сучасного бізнесу.

РОЗДІЛ 2

ІНТЕГРАЦІЯ ТЕХНОЛОГІЇ ІРААС У ПРОЕКТУВАННЯ АВТОМАТИЗОВАНИХ СИСТЕМ

Хмарні обчислення внесли зміни в парадигми функціонування ІТ-індустрії. Відповідно до прогнозів видання IDC в проекті FutureScape 70% компаній зменшать витрати і складнощі, що пов'язані з розробкою та впровадженням додатків, шляхом більшого впровадження SaaS платформи та додатки до 2024 року [9].

Через різке зростання використання програмного забезпечення як послуги (SaaS) та інших хмарних сервісів розробникам трапляється низка перешкод. Основна потреба в поєднанні локальних систем з хмарними програмними рішеннями ускладнює та робить більш трудомістким процес інтеграції. Такі ускладнення дозволяє вирішити гнучка інтеграційна платформа як послуга (iPaaS). У міру того як ІТ-світ ставав складнішим, підходи iPaaS розвивалися. З'явилися нові сфери, зокрема cloud-to-cloud, IoT-to-on-premises та EDI з підтримкою iPaaS. У результаті великі постачальники програмного забезпечення, такі як SAP, Microsoft або Oracle, почали розширювати своє SaaS портфоліо. Вони почали працювати з постачальниками iPaaS, а згодом викупили їх або створили власні рішення для того, щоб полегшити інтеграцію хмари та SaaS своїх клієнтів з локальними рішеннями.

Перевагою великих постачальників послуг iPaaS є те, що вони здатні легко та гнучко масштабуватись через власні сервіси або своїх партнерів. Однак інтеграція систем сторонніх виробників залишається проблематичною для них. Тому останнім часом з'явилося багато інших постачальників iPaaS, які є більш зосередженими на вимогах бізнесу та є більш гнучкими в цьому плані. Ці передумови створили високу конкуренцію якісних послуг на ринку.

2.1. Загальний огляд технології iPaaS

Оскільки платформа інтеграції як послуга (iPaaS) — це набір автоматизованих засобів та інструментів, які дозволяють інтегрувати програмні додатки, що розгортаються в різних місцях. Великі компанії, які використовують системи різного рівня, часто використовують iPaaS для інтеграції додатків і даних, які

розгортаються локально, а також у публічних і приватних хмарах (зображено на рисунку 2.1).

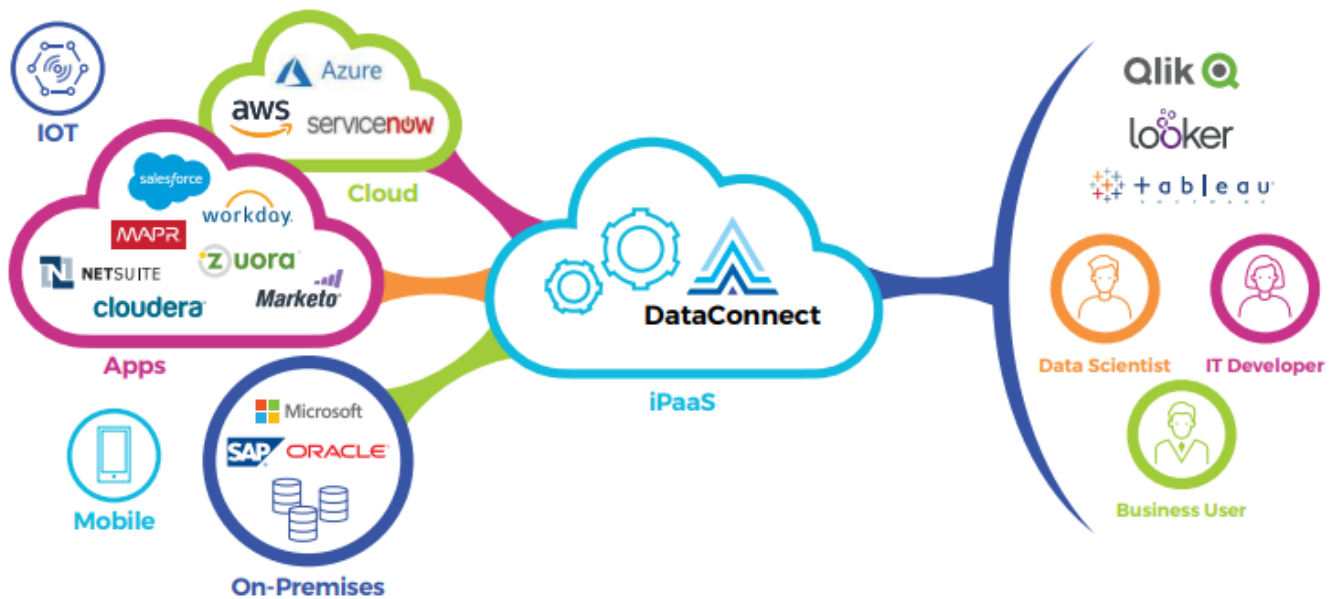


Рис. 2.1. Схема інтеграції найбільш популярних хмарних застосунків та приватних хмар

Зазвичай інструменти iPaaS надають певний набір готових функцій, бізнес-правил, конекторів, що здатні полегшити розробку систем та оркеструють інтеграційні потоки. Зараз дуже поширеною пропозицією деяких постачальників iPaaS є набір спеціальних засобів для розробки, що дозволяють модернізувати застарілі програми, додавати мобільну підтримку, керування бізнес-процесами, інтеграцію з різними системами.

За своєю конструкцією iPaaS не є залежною від платформи та може об'єднувати будь-яке поєднання локальних і хмарних процесів, даних і систем – у межах однієї компанії чи між організаціями.

Платформа iPaaS дозволяє компаніям об'єднувати, захищати та управляти системами, які інакше не мають можливості комунікувати та обмінюватися даними. Ця платформа може допомогти підвищити ефективність, зменшити витрати, а також забезпечити кращий та гнучкіший спосіб роботи з клієнтами. Для застарілих

програмних засобів, що потребують у модернізації архітектури iPaaS виступає ключем до вирішення проблеми.

Залежно від постачальника виділяють різні типи інтеграції iPaaS. Деякі з цих інтеграцій є сфокусованими виключно на певній області, а інші в свою чергу пропонують комбінацію різних послуг. Найбільш поширеними iPaaS інтеграціями є:

- Хмарна інтеграція. Дозволяє поєднати кілька гібридних або хмарних середовищ у єдину систему. Підтримує побудову та виконання різних потоків інтеграції, інтеграції процесів та обміну повідомленнями.

- Електронна комерція та інтеграція B2B. Впроваджує автоматизацію зв'язку та оптимізує процеси між клієнтами, постачальниками та торговими партнерами. Це призводить не лише до ефективніших та масштабованих процесів у сфері електронної комерції, завдяки єдиному розумінню взаємодії з клієнтами, але й відкриває можливості для швидкої та безпечної співпраці, починаючи від виставлення рахунків і закінчуючи відстеженням поставок та іншими етапами.

- Інтеграція бізнесу з урядом (B2G). Підтримує безпечні електронні комунікації та транзакції між державними органами та підприємствами. Допомагає стандартизувати цифрову реєстрацію документів, впровадити електронний документообіг.

- Інтеграція та міграція даних. Завдяки можливостям інтеграції даних інформація з кількох джерел поєднується в єдине структуроване та уніфіковане уявлення. Сервіси міграції даних фактично переносять дані з однієї системи до іншої.

- Інтеграція API. Дозволяє декільком системам обмінюватися даними та взаємодіяти один з одним автоматично через API (прикладний програмний інтерфейс) кожної системи. Після підключення ці системи можуть взаємодіяти в реальному часі.

Типи інтеграції, необхідні компанії, мають враховуватися при виборі постачальника iPaaS.

Платформа iPaaS підтримує графічну розробку інтеграційних програм. Це значно спрощує та прискорює інтеграцію, а також дозволяє уникати написання

додаткового коду для цього. Розробники можуть будувати логіку інтеграції, використовуючи готову бізнес-логіку, віджети та компоненти, що легко налаштовуються до вимог. Тому нову інтеграції, їхнє оновлення може швидко вводиться в експлуатацію, відповідаючи потребам бізнесу, що постійно змінюється.

Архітектурно iPaaS як правило містить наступні компоненти та рівні забезпечення постійного обміну та синхронізації даних:

- інтерфейс користувача: представлений у вигляді веб-консолі, завдяки якій користувачі взаємодіють з платформою, та надає зручний інтерфейс для проектування та керування інтеграціями;
- потоки інтеграції:
 - конструктор потоків інтеграції – інструмент для візуального проектування потоків інтеграції (визначення ланцюга операцій, порівняння та перетворення даних за допомогою drag-and-drop інтерфейсу);
 - конектори – завчасно створені з'єднувачі, що роблять простішим зв'язок програм, баз даних і служб та абстагують складність взаємодії даних через API;
- рівень оркестрування:
 - Workflow Engine – управляє потоками інтеграції, забезпечуючи виконання завдань у визначеному порядку та обробку виключень та помилок;
 - керування бізнес процесами (BPM – Business Process Management) – надає можливість моделювання та виконання складних бізнес-правил, розширюючи можливості оркестрування;
- перетворення та відображення даних:
 - Transformation Engine забезпечує сумісність даних між різними системами шляхом перетворення форматів даних;
 - mapping інструменти – графічні інтерфейси, що визначають зіставлення даних і перетворень для полегшення вирівнювання структур даних;

- рівень підключення: конектори встановлюють підключення до різних систем, а адаптери є більш специфічними з'єднувачами для певних програм;
- рівень безпеки:
 - шифрування – забезпечує цілісність даних під час передачі, зберігає їх та забезпечує конфіденційність;
 - керування ідентифікацією та доступами до платформи;
 - контроль відповідності галузевих норм;
- runtime інтеграція: середовище виконання виконує інтеграційні потоки, керує поведінкою інтеграцій та обробляє дані, забезпечує оптимальний розподіл ресурсів, забезпечує масштабованість для впровадження робочих навантажень;
- керування API: шлюз API надає можливість компаніям використовувати власні інтеграції як API для внутрішнього або зовнішнього використання іншими програмами;
- моніторинг та управління:
 - аудит – запис детальних журналів процесів інтеграції, помилок і виключень для усунення несправностей та ведення аудиту;
 - інформаційна панель, що візуалізує потоки інтеграції, продуктивність системи та журнали помилок;
 - сповіщення, що надсилаються в реальному часі або за розрахованими графіками та правилами;
- сховище метаданих пов'язаних з потоками інтеграції, форматуванням, що надають видимість, керування та аналіз впливу;
- управління ЖЦ:
 - контроль версій – дозволяє управляти версіями потоків інтеграції, порівнювати версії та повертатись до попередніх;
 - контроль змінами – забезпечує кероване планування, тестування та впровадження змін у процесі інтеграції.

Така архітектура iPaaS забезпечує надійне, масштабоване, гнучке середовище для оптимізації інтеграційних процесів та легку адаптацію до мінливих бізнес-вимог. Хмарна сутність iPaaS забезпечує доступність, співпрацю у відповідь на постійні потреби в інтеграції.

2.2. Вплив iPaaS на архітектуру систем

Коли перші постачальники SaaS вийшли на ринок, перед ними постала проблема, що їхні клієнти потребували допомоги з інтеграцією хмарних програмних рішень з уже існуючими локальними системами та між собою. Це стало гарною можливістю для перших постачальників iPaaS послуг. Тому iPaaS та SaaS провайдери почали тісно взаємодіяти. В результаті для постачальників SaaS послуг це було вигідно, адже їм не довелося шукати нових рішень для вирішення проблем з інтегруванням даних, вони могли зменшити або повністю делегувати цю роботу. iPaaS провайдерам в свою чергу не потрібно було шукати клієнтів для їхніх послуг. Тому ця ситуація посприяла зростанню обох сторін.

В цілому, інтеграція стає складним процесом, це пов'язано з стрімким зростанням пропозицій на рішення, включаючи PaaS і SaaS, а також з збільшенням проблеми обробки великих даних. До того ж необхідно вирішити складнощі, що пов'язані з переходом локальних рішень на гібридну чи хмарну архітектуру. Проте використовуючи iPaaS, процес інтеграції можна зробити кращим, плавнішим та економічно ефективнішим. Процес інтеграції повинен включати кілька основних функцій, щоб бути комплексним рішенням iPaaS, наприклад [10]:

1. API Management. Функція керування API слугує основною частиною доступу та обміну даними в режимі реального часу. Компанії, які його використовують, мають повний контроль над створенням, розгортанням і керуванням API.

2. Централізоване управління. Для того щоб компанії працювали ефективніше, інтеграція повинна здійснюватися з централізованого місця. За допомогою рішення iPaaS підприємства можуть переглядати інтеграцію безпосередньо із зручної веб-

консолі. Це дозволить створювати, управляти, контролювати, розширювати та розвивати інтеграції в усій організації.

3. Хмарний шлюз B2B. Електронний обмін даними досить довго розглядався як рішення для обміну документами або даними між системами. Однак завдяки iPaaS процес встановлення зв'язків між хмарою та локальними процесами став швидшим і простішим водночас.

iPaaS — це не тільки переміщення даних між системами та інструментами, хоча обмін даними є основною функцією платформи. Окрім цього, iPaaS також може:

- відображати дані між програмами, щоб забезпечити повний огляд потоків даних;
- перетворювати дані між програмами та системами, щоб інструменти з різними форматами могли обмінюватися даними автоматично, без ручної роботи;
- відстежувати та покращувати якість даних, якщо це необхідно;
- застосовувати політики для захисту даних від несанкціонованого доступу.

Постачальники iPaaS розміщують сервери додатків та дані інфраструктури. При цьому, вони надають інструменти для інтеграції та проміжне ПЗ, що дозволяє розробникам створювати програмне забезпечення, що розміщене в хмарі, розгортати його, тестувати та керувати. Багато пропозицій iPaaS здатне прискорити розвиток потоків інтеграції в бізнесі за допомогою попередньо створених конекторів і бізнес-правил для визначення взаємодії в середовищах із кількома клієнтами.

Процес впровадження iPaaS виглядає наступним чином. Спочатку компанії обирають постачальника iPaaS на основі наявних програм організації та задач інтеграції. Далі бізнес-користувач оформляє підписку, обираючи доступні інструменти та послуги постачальника. Постачальники iPaaS займаються керуванням різноманітними послугами, в той час коли бізнес шукає та запитує підтримку спеціальних функцій додатків і програмного забезпечення.

iPaaS надає можливість розгортати процес інтеграції без втручання зовнішнього програмного забезпечення компанією, яка займається програмним

забезпеченням, або окремим користувачем. Завдяки використанню iPaaS компанії можуть розширити спектр послуг, що надають, при цьому не створюючи додаткових функцій, а інтегруючи уже наявне програмне забезпечення. Використання iPaaS здатне підвищити рівень задоволення клієнтів, адже можуть запропонувати послуги як надійніше рішення на додаток до існуючого набору послуг.

Платформа інтеграції як послуга зазвичай надає ряд готових з'єднувачів і API, які забезпечують безперебійний зв'язок між цими різними системами. Наприклад, він може пропонувати конектори для популярних програм SaaS, таких як Salesforce, Workday і HubSpot, а також для локальних програм, таких як SAP і Oracle. Найчастіше основною задачею iPaaS інтеграцій є трансформація та переміщення даних між несумісними платформами (це зображено на рисунку 2.2). Прикладом такої роботи може бути наступний сценарій. Компанія використовує SAP для ERP (Enterprise Resource Planning – планування ресурсів підприємства), Salesforce для прямого контакту з кінцевими користувачами та інше програмне забезпечення для оцінки задоволеності клієнтів. Усі ці системи працюють з даними та показниками, які оброблюються окремо в кожній системі, проте залежить один від одного. iPaaS об'єднує їх та зберігає у спільне сховище даних, звідки вони потім можуть бути отримані незалежно. Контактні дані та контракти нових клієнтів, які спочатку вводяться в Salesforce, будуть синхронізовані з системою ERP, збережені в одне місце та автоматично включені для створення наступного звіту іншим ПЗ.

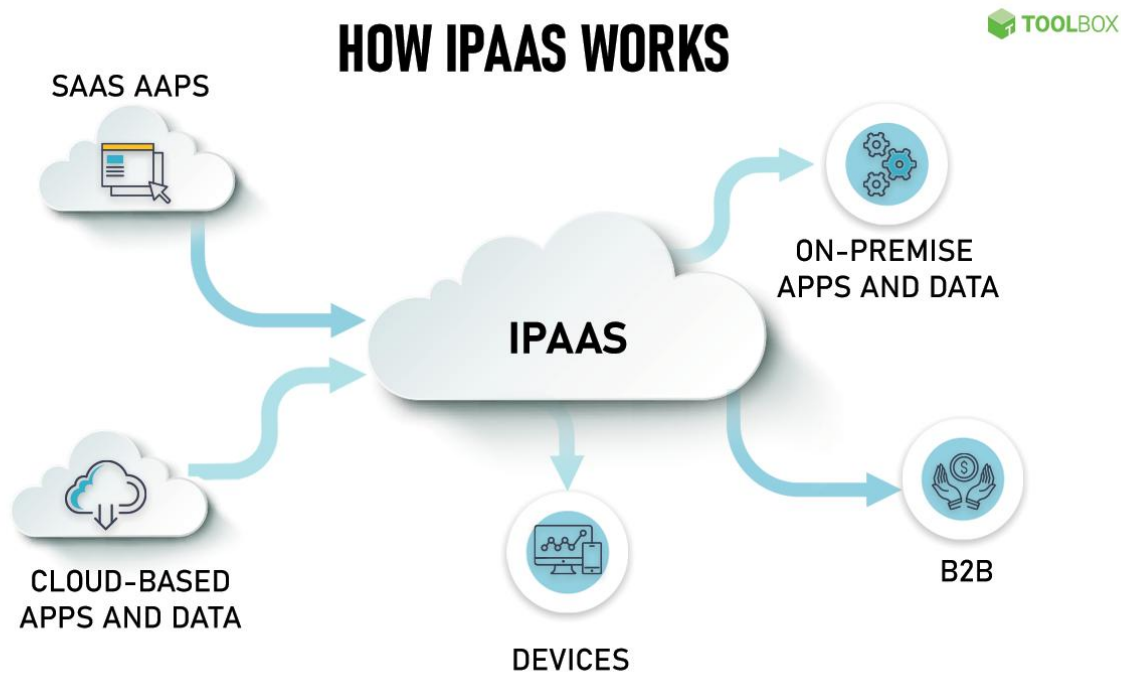


Рис. 2.2. Принцип роботи iPaaS

iPaaS надає корисний вихід для керування API. Це зручно для організацій, адже цей спосіб надає єдину платформу для керування API та конектором даних. Ця платформа здатна забезпечити гнучкість додавання нових послуг та легке масштабування операцій. В свою чергу єдиний центр даних дозволяє отримувати доступ і обмінюватися даними в реальному часі в усій організації. Це надає можливість швидшої розробки та впровадження стратегії на основі даних. Конектори даних або з'єднувачі – це попередньо створений код, що надає поля та сутності з конфігурації кінцевої точки інтерфейсу користувача платформи та створює з'єднання з певними програмами та джерелами даних. Конектори можуть працювати через брандмауери для локальних мереж через захищені з'єднання. Більшість постачальників iPaaS мають в арсеналі бібліотеку конекторів для загальних кінцевих точок, що обов'язково включає загальні з'єднувачі для REST API (або набір інструментів розробки для створення власних) [11].

Важливою особливістю використання iPaaS є те, що вона підходить для multi-tenant систем. Мультитенант – це архітектура, де один екземпляр ПЗ працює на одному сервері та обслуговує кількох клієнтів, створюючи для них власні інстанси.

Тобто до системи з кількома клієнтами окремі клієнти підключаються одночасно до одного апаратного забезпечення та сховища даних, створюючи окремий екземпляр для кожного клієнта. Більшість підходів не підтримує таку архітектуру.

2.3. Переваги та обмеження використання iPaas

Сучасні технології iPaas мають ряд значних переваг для сучасного бізнесу. Серед основних переваг використання iPaas можна визначити:

- кращий менеджмент витрат (iPaas спрощує інтеграцію даних за регулярну щомісячну або річну підписку, значно спрощуючи частину витрат; дозволяє не залучати розробників для написання коду з інтеграції клієнтів; відсутність необхідності в апаратному та програмному забезпеченні);

- посилення безпеки даних (відповідальність за безпеку покладається на постачальника iPaas, який забезпечує вбудовані інструменти для виявлення та спостереження за загрозами);

- захищений шлюз (безпека між локальними програмами та хмарами є однією з основних в розрізі хмарної інтеграції, в свою чергу iPaas здатний передавати дані більш безпечним способом, не потребуючи кодів клієнтів; до того ж постачальники iPaas гарантують постійні оновлення функцій захисту);

- більш швидкий та кращий доступ до даних (iPaas містить хмарний набір інструментів, що розроблений для обробки збільшеної кількості інформації, яка пов'язана із складними транзакціями B2B, роботі в хмарних середовищах, підтримкою мобільних пристроїв; це включає інтеграцію в реальному часі, основною метою чого є мінімізація збоїв, спричинених непідтримуваністю роботи додатків з іншими форматами даних);

- надійніші дані (iPaas платформи надають можливість створювати загальну базу даних та цілісно обробляти дані, замість того, щоб переглядати відокремлені бази даних для кожної програми; тому дані простіше синхронізуються та є більш валідними);

- зміна архітектурної моделі та моделі ліцензування (на відміну від застарілого проміжного програмного забезпечення, що найчастіше працює за

принципом ESB – сервісної шини – iPaaS усуває обмеження центрального провідника, що сполучає різні системи; хоча інтеграція контролюється в одній точці та працює разом; перехід архітектури до iPaaS забезпечує «завжди відкриту» платформу, що об'єднує програми, системи, дані, процеси та людей);

- масштабованість (завдяки iPaaS клієнти легко можуть розширити функції системи, це дозволяє бізнесу розвиватись як з технічної, так і з комерційної сторони);

- підтримка low-code розробки (iPaaS надає можливість використовувати користувацький інтерфейс та готові з'єднувачі для візуального планування та впровадження інтеграцій; завдяки використанню low-code інструментів, інтеграція може здійснюватися з меншими технічними знаннями, адже низький вміст коду створює правильну видимість потоку інтеграцій та спрощує етап проектування);

- Agile Enablement (iPaaS сприяє більшій гнучкості бізнесу, що в свою чергу призводить до здатності швидко та ефективно коригувати свої витрати; керувати інтеграціями, створювати, редагувати та видаляти їх в будь-який момент, не турбуючись про дозвіл постачальників або додаткові витрати)

- оновлення над оновленнями (iPaaS забезпечує регулярні, автоматичні та надійні оновлення, це призводить до можливості бізнесу використовувати останні версії, зниження витрат на обслуговування та підвищення надійності).

Також iPaaS розширює прийняття рішень. В бізнес середовищі, що постійно змінюється, можливість миттєво реагувати на зміни є вкрай важливою для того, щоб запобігти зайвим витратам. Це досягається тим, що iPaaS зменшує розрив між інтеграцією даних та бізнес аналітикою. Тому компаніям надається можливість відслідковувати дані про операцій в режимі реального часу та приймати максимально точні рішення. Також iPaaS технології значно скорочують час для виявлення потрібних деталей, адже вона працює в хмарі та отримує дані з різних систем.

Ще однією проблемою, яку вирішує iPaaS, є data silos (бункер даних або резервуари даних). Бункер даних — це набір інформації, що є ізольована від організації та недоступний для всіх або деяких частин компанії. Такі дані знижують

прозорість, ефективність і довіру всередині організації та перешкоджають продуктивності бізнесу. Резервуари даних також виснажують грошові ресурси, оскільки компанії змушені платити за додаткове зберігання даних і вартість неефективності. Розташування даних створює дорогі та трудомісткі проблеми для компаній, але iPaaS відносно легко вирішує їх. iPaaS допомагає усунути накопичення даних, створюючи зв'язки між системами компанії та програмним забезпеченням. Це особливо важливо для підприємств, які швидко розвиваються, щоб забезпечити безперебійний обмін даними між програмами. За допомогою iPaaS також можна видалити повторювані контакти та оптимізувати потоки даних між командами. Це сприяє кращому узгодженню між різними командами.

Платформа iPaaS здатна автоматизувати та оптимізувати робочі процеси, забезпечуючи розширені функції інтеграції для них. Таким чином значно скорочується людська (ручна) робота та введення даних, оптимізуються та автоматизуються повторювані ручні завдання (наприклад, розсилка маркетингових листів, підтримка клієнтів, виставлення рахунків, тощо). Тому організації стають здатні до певної операційної досконалості, якщо використовують iPaaS як головну платформу для розширення можливостей автоматизації бізнесу.

Переваги та варіанти використання iPaaS переконливі, але не всі рішення iPaaS однакові. Потужність деяких традиційних пропозицій iPaaS полягає в складних системах, які вимагають використання сертифікованих експертів з розробки інтеграції.

Не зважаючи на низку переконливих переваг використання iPaaS платформ, слід зазначити низку викликів та обмежень, що можуть бути пов'язані з впровадженням iPaaS:

- кожна iPaaS платформа створена для задоволення різних вимог і потреб бізнесу, це ускладнює вибір правильного провайдера, адже багато пропозицій працюють ефективніше за певних обставин;
- додаткова складність (iPaaS платформи розроблені для полегшення процесу інтеграції корпоративних даних, проте системам із високою складністю все ще необхідно інвестувати в аналіз та проектування, щоб правильно налаштувати й

підтримувати інструменти для власних робочих процесів; при цьому ця потреба виникатиме завжди при розвитку інтеграцій);

- потреба у власних експертах (постачальники iPaaS покладають відповідальність за більшість управлінських проблем на себе, проте компанії повинні мати власний персонал для нагляду за роботою iPaaS інструментів та оцінки їхнього впливу на бізнес);

- ризик збою обслуговування (оскільки компанії купують послуги у третіх сторін, покладаючись на них, будь-які збої, наприклад, мережеві, призводять до проблем інтеграції та ставлять під ризик всю роботи системи; також у зв'язку з збільшенням кількості провайдерів iPaaS та їхнього спектру послуг, можливе припинення роботи деяких постачальників або послуг, що призводить до вимушеності змінювати постачальника);

- питання безпеки (найчастіше постачальники iPaaS забезпечують безпеку інтеграції даних, проте фінальну відповідальність за порушення чи втрату даних все ж несе організація; ця проблема особливо гостро постає під час інтеграції з сторонніми системами, хмарами та SaaS);

- певні обмеження масштабованості (iPaaS платформи є легко масштабованими, проте мають певні межі, наприклад, багато iPaaS інструментів потребують втручання сторонніх осіб для розширення платформи; автоматизація вимагає оркестровки обіцяє масштабованість, але масштабованість має межі; багато систем iPaaS потребують втручання користувача для масштабування платформи; автоматизація вимагає оркестровки; компаніям слід враховувати швидкість транзакцій та максимально допустимий розмір);

- обмеження налаштування (iPaaS рішення зазвичай пропонують готові з'єднувачі та шаблони для популярних програм, але можливості налаштування можуть бути обмеженими; існує ймовірність, що компаніям потрібно буде створювати власні конектори або використовувати інших провайдерів для задоволення конкретних вимог).

Не зважаючи на те, що iPaaS інтеграції є достатньо простими в налаштуванні, все ж повною мірою платформи можуть використовувати лише відповідні

спеціалісти. Тому це створює складність інтеграції в середині компанії. Прикладом такої проблеми може бути випадок, коли бізнес використовує програмне забезпечення для інтеграції як послугу самостійно без втручання ІТ-спеціалістів, створюючи інтеграції програм та автоматичних задач, що може призвести до ризику стороннього втручання.

Ще однією проблемою використання платформи для інтеграції є необхідність повторного підключення потоків iPaaS кожного разу, коли встановлюється нова точка інтеграції, щоб адаптувати нову точку в системі. Коли трапляється така ситуація, продуктивність нового підключення потрапляє до існуючих потоків, що може спричинити затримки. В результаті, така інтеграція може зменшити маркетинговий хід, бо впровадження нових функцій впливає на систему в цілому та робить її менш надійною. Ці проблеми можна вирішити, обравши рішення iPaaS, у якого є функція автоматизованої сегментації даних та їхнім керуванням в масштабованій локальній, хмарній або гібридній архітектурі. Такий підхід забезпечує архітектурну гнучкість, незважаючи на зміни в налаштуваннях інтеграцій[12]

Отже, для того, щоб визначити, вирішить iPaaS задачі повністю чи частково, перш за все треба визначити чи є платформа рішенням. Залежно від потреб бізнесу може бути обраний найбільш підходящий спосіб інтеграцій. На відміну від традиційних готових інтеграційних рішень, платформи iPaaS можуть гарантувати, що весь стек програм працює синхронно, при цьому потреба в ручному втручанні мінімальна. Цей підхід може зробити організацію менш схильною до помилок, одночасно підвищуючи гнучкість і прозорість у бізнес-середовищі, що постійно змінюється.

Загалом iPaaS може застосовуватися для вирішення багатьох потреб в інтеграції на різних платформах та з різними джерелами даних. Поширеними способами використання iPaaS платформ є [13]:

- інтеграція від програми до програми (application-to-application): підключення систем та даних, автоматичне впровадження інтерфейсів та потоків даних для розгортання та керування кожною інтеграцією;

- інтеграція мікросервісів: iPaaS здатен автоматично створювати та публікувати API для підтримки мікросервісів, це зменшує необхідність власної розробки;
- B2B інтеграція: iPaaS керує інтеграціями сторонніх систем з B2B без необхідності в спільному API чи клієнтських інтеграціях за стандартами безпеки;
- інтеграція даних: керування приведенням різних форматів даних відбувається інструментами iPaaS, які також здійснюють синхронізацію даних;
- інтеграція платформ, що розгортаються в різних локаціях;
- інтеграція пристроїв IoT (Internet of Things – колективна мережа підключених пристроїв і технологій, яка забезпечує зв'язок між пристроями та хмарою): iPaaS контролює конектори між пристроями IoT, інструментами та API для збору, обробки та передачі даних;
- інтеграція великих даних;
- декілька хмарних інтеграцій: iPaaS може впроваджувати та забезпечувати складні інтеграції загальнодоступних хмарних ресурсів та інших хмар, наприклад, постачальниками SaaS;
- автоматизовані бізнес-процеси: iPaaS ефективно використовується для узгодження з бізнес-задачами, адже здатна усунути ручні процеси.

Висновок

iPaaS — це спеціальний тип пропозиції програмного забезпечення як послуги (SaaS), що є конектором між хмарою та локальними джерелами. Платформа здатна полегшити процеси інтеграції, адже використовує low-code та no-code інструменти, маючи легкий інтерфейс користувача. iPaaS здатен легко масштабуватись, оскільки є хмарною службою, та забезпечувати необхідну кількість ресурсів для обчислення, що беруть участь в інтеграційних процесах.

Платформи iPaaS є корисними фреймворками для оптимізування процесів та введення інновацій.

Оскільки процес переходу до хмарних рішень є поступовим, попит на рішення, які можуть допомогти в інтеграції старих систем із новою архітектурою

постійно зростає. Завдяки популярності iPaaS останнім часом, компанії мають великий вибір конкурентних пропозицій для подолання проблем пов'язаних з інтеграцією даних.

Однією з ключових переваг інтеграції iPaaS є те, що вона дозволяє організаціям отримувати доступ до даних із різноманітних джерел. Використовуючи iPaaS для інтеграції систем, що знаходяться в різних джерелах, компанії синхронізовано отримують дані в реальному часі. Це створює більш повне розуміння власного бізнесу та операцій, а також допомагає у визначенні закономірностей та тенденцій, повну картину яких не можна було б відслідкувати з одного джерела. Тому бізнес може приймати більш обгрунтовані рішення.

Порівняно з програмними рішеннями для інтеграції даних, iPaaS дозволяє створювати автоматичні інтеграції без або з мінімальним залученням коду, тому інтеграції можуть легше та дешевше впроваджуватись бізнесом.

Платформи iPaaS зазвичай є основою стратегії інтеграції та може задовольнити більшість інтеграційних потреб. Проте компаніям складно зробити їх центром інтеграційної стратегії одразу. Це пов'язано з використанням застарілих технологій та заплутаною мережею даних.

Слід зазначити, що неможливо досягти вирішення всіх питань тільки використанням платформ iPaaS, адже вони інколи потребують доповнень, конекторів та інших SaaS.

Оскільки цифрова трансформація продовжує розвиватись, iPaaS відіграватиме все більш значну роль для багатьох компаній, кожна з яких використовує все більше і більше систем та ресурсів для ефективної роботи. Кожен програмний продукт зберігає дані бізнесу, які часто повторюються в різних модулях компанії в різному форматі і потребують постійної синхронізації.

РОЗДІЛ 3

МЕТОДОЛОГІЯ ПРОЕКТУВАННЯ АВТОМАТИЗОВАНИХ СИСТЕМ НА БАЗІ ІРААС

Незважаючи на те, що платформи іPaaS мають низку переваг для оптимізації складних інтеграційних процесів, для досягнення найкращих результатів забезпечення бізнес вимог слід використовувати рекомендовані іPaaS практики, наприклад:

- **Покриття стеку:** компаніям варто сформулювати комплексний план інтеграції для впровадження іPaaS, що включатиме всі програми для інтегрування (найкращим підходом буде забезпечення інтеграції для усіх систем та типів даних, в протилежній ситуації бізнес може натрапити на необхідність в додаткових механізмах інтеграції, коли їх знову доведеться аналізувати та налаштовувати, це зменшить цінність іPaaS).

- **Поєднання різних іPaaS платформ:** подеколи різні іPaaS платформи виглядають дублюючими, проте не кожна з них пропонує однаковий перелік конкретних інтеграцій. Компанії мають можливість розгортати декілька іPaaS платформ одночасно, так як вони є хмарними продуктами. Це дозволить покрити усі потреби, хоча в такому випадку необхідно ретельніше керувати інтеграціями.

- **Моніторинг збоїв:** провайдери іPaaS здебільшого несуть відповідальність лише за інфраструктуру та ПЗ, як і будь-яка третя сторона вони можуть постраждати від збоїв. Тому слід ретельно ознайомитись з переліком послуг, що надає постачальник, перед оформленням підписки, щоб чітко визначити зони відповідальності кожної із сторін та бути попередженими про можливі простой служби, що впливатимуть на бізнес.

- **Повне розуміння конфігурації:** іPaaS забезпечує автоматизацію процесів компаній, але лише організації несуть остаточну відповідальність за якість послуг, що вони надають. Тому сторона компанії має мати внутрішній персонал, що забезпечує процес нагляду та контролю діяльності платформи. Це допоможе переконатися в тому, що надане меню послуг дійсно відповідає необхідним замовнику.

- Чистота даних: незважаючи на те, що здебільшого iPaaS платформи виконують управлінську роботу, варто регулярно перевіряти дані вручну. Слідкування за своїми базами даних, що включає видалення дублікатів, зміна та оновлення інформації, усунення непотрібних даних, дозволить переконатись, що процеси інтеграції працюють правильно.

- Оптимізація та зміни: оскільки хмарні технології стрімко розвиваються, iPaaS може потребувати систематичних оновлень, переналаштувань інтеграцій або робочих процесів відповідно до змін середовища. Періодичний перегляд активних інтеграцій дасть змогу визначати нові плагіни чи модулі обраного провайдера, що призведе до підвищення продуктивності та спрощення бізнес-процесів.

- Використання Agile підходу: розбиття великих задач інтеграції на маленькі ітерації, що адаптують процес інтеграції до змін та пришвидшують його.

Використання цих практик та рекомендацій дозволяє оптимізувати використання iPaaS для забезпечення ефективнішого процесу інтеграції.

3.1. Етапи проектування автоматизованих систем

Проектування автоматизованих систем з залученням iPaaS платформ для інтеграції – це комплексний процес, який розгортається через окремі етапи для ефективних взаємопов'язаних рішень. iPaaS платформи пропонують динамічну структуру для простого підключення сервісів.

Від початкової концептуальної моделі до впровадження ПЗ, етапи проектування систем, що використовують iPaaS рішення, застосовують комплексну методологію, що стосується різних аспектів. Оскільки компанії дедалі частіше використовують iPaaS інструменти для задоволення бізнес потреб та підвищення своїх операційних можливостей, володіння специфікою кожного етапу дизайну системи стає вимогою для отримання ефективних результатів. Розробка автоматизованих систем із використанням платформ iPaaS передбачає цілісний підхід, що включає стратегічне планування, ретельне відображення даних, оптимізацію робочого процесу та дотримання стандартів безпеки та відповідності.

Нижче представлена методологія розкриває основні етапи та розсудливі рішення, які враховуються під час цього процесу:

1) Оцінка потреб:

- визначення мети та бізнес-вимог, на вирішення яких буде спрямована інтеграція даних та її автоматизація;
- аналіз предметної області, порівняння існуючих систем, погляд процесів та потенційно ризикових місць;
- приєднання зацікавлених осіб для визначення вимог та очікувань від автоматизації процесу інтеграції.

2) Проектування архітектури програмної системи:

- визначення частин, де буде залучена автоматизація, адаптація існуючої архітектури системи під використання iPaaS модулів, включаючи масштабованість, гнучкість та модульність;
- оцінка сумісності існуючих програм з iPaaS платформами;
- визначення точок інтеграції існуючої системи та iPaaS;
- розробка архітектури даних, що буде основана на безперебійному потоку інформації між різними компонентами та середовищами.

3) Вибір iPaaS інструментів:

- аналіз існуючих iPaaS рішень, що здатні задовольнити поставлені вимоги та відповідають спроектованій архітектурі;
- оцінка простоти використання, рівня автоматизації, необхідності ресурсів, залучених в процес інтеграції, масштабованості, доступних конекторів та рівня безпеки;
- оцінка довгостроковості інтеграції;

4) Стратегія інтеграції:

- розробка стратегії інтеграції, де буде визначено способи передачі даних між системами;
- вибір шаблону інтеграції (point-to-point, middleware, ETL, ELT та ін. – детальніше описані в розділі 1.1);
- оцінка рівня узгодженості стратегії інтеграції з бізнес-вимогами,

ефективності обміну даних.

5) Розробка правил синхронізації, форматування та виведення даних:

- визначення умов відображення даних для забезпечення послідовної та чіткої передачі даних між інтегрованими частинами системи;
- розробка логіки перетворення даних інструментами iPaaS для уникнення проблем з невідповідності форматів та структур даних у різних системах;
- визначення процесів валідації даних, видалення даних для забезпечення їхньої цілісності в процесі інтеграції.

6) Впровадження заходів безпеки:

- забезпечення захисту зберігання та передачі конфіденційних даних;
- забезпечення відповідності галузевим нормам та внутрішнім і зовнішнім стандартам організації;
- впровадження шифрування, контролю доступу до системи;
- впровадження аудитів та журналів логування помилок та виключень для підвищення загальної безпеки системи.

7) Рівень взаємодії користувача:

- розробка інтерфейсів, що будуть спроектовані з дотриманням кращих практик, інтуїтивно зрозумілі технічним та нетехнічним користувачам, залученим до інтеграції;
- створення інформаційних віджетів та додатків для моніторингу продуктивності роботи системи, якості передачі даних та визначення потенційно вразливих процесів в реальному часу;
- тестування зручності використання.

8) Забезпечення якості інтеграції:

- розробка комплексного плану тестування інтеграції даних;
- впровадження автоматизованого тестування з потенційним залученням сторонніх сервісів;
- performance та load тестування для оцінки продуктивності процесів інтеграції.

9) Розгортання:

- поступове розгортання систем, що залучені в інтеграцію, для мінімізування збоїв та виключень в операціях;
- налаштування процесів моніторингу продуктивності інтегрованих систем;
- адаптація до мінливих вимог бізнесу та можливості масштабування.

10) Документування:

- створення детальної документації архітектури процесів інтеграції, потенційно слабких місць інтеграції, рекомендацій для виправлень несправностей;
- оновлення документації в разі появи нових інтеграцій;
- навчальні сесії для команд підтримки, користувачів та відповідальних осіб.

11) Зворотній зв'язок:

- налаштування механізмів зворотнього зв'язку від кінцевих користувачів для визначення вразливих областей, вдосконалення та оптимізації процесів інтеграції;

12) Оновлення та оцінка ринку iPaaS технологій для вдосконалення функцій, впровадження нових можливостей.

Дотримання подібного комплексу заходів дозволить організаціям створювати та розгортати системи, що будуть задовольняти їхні бізнес-плани, адаптуватися до динамічного розвитку сучасних рішень та підвищувати ефективність власних процесів.

Загалом наведені етапи для впровадження інтеграції здатні забезпечити чітко керовану та організовану структуру для легшого проектування автоматизованих систем та оптимізування інтеграційних процесів.

3.2. Постачальники та інструменти iPaaS

iPaaS платформи є зручним рішенням для інтеграції програм та систем в хмарному або локальному середовищі. Вони мають більше можливостей, ніж,

наприклад, корпоративна шина, але мають ряд перешкод у великих проектах в порівнянні з гібридною інтеграцією.

Стрімкий розвиток цифрової інтеграції призвів до широкого вибору iPaaS інструментів та вендорів, а їхній вибір безпосередньо впливає на успіх та ефективність бізнесу. Провайдери від ІТ гігантів до успішних стартапів пропонують рішення для гнучкої та масштабованої основи для з'єднання систем, надаючи велику варіацію функцій та моделей розгортання. Вибір найбільш підходящих iPaaS інструментів має стратегічне значення в успіху інтеграції та дозволяє компаніям досягати стійкості, гнучкості та адаптивності своїх стратегій. Глибокий аналіз iPaaS постачальників та списку їхніх послуг допомагають бізнесу обрати вірне рішення. Слід зазначити, що дослідження ринку цифрових інтеграцій є повторюваним процесом, незважаючи на те, чи вже функціонує інтеграція системи, чи це створення нової інфраструктури.

Вибір iPaaS рішення, що працюватиме для поставлених вимог до інтеграції найкраще, непростий. Популярні постачальники Mulesoft's Anypoint, Prismatic, Microsoft, Zapier та багато інших представляють широкий вибір функцій інтеграції. Нижче буде наведено порівняння двох груп альтернативних послуг від найбільш відомих провайдерів за певними критеріями, такими як складність інсталяції, ціна, безпека, дата менеджмент тощо, – Microsoft Azure Logic Apps, SAP CPI з Alumio й Prismatic, Tray.io та Paragon.

3.2.1. Порівняння Microsoft Azure Logic Apps, SAP CPI та Alumio

Azure Logic Apps – це хмарна iPaaS служба, яка дозволяє створювати та запускати автоматизовані робочі процеси без коду або з невеликим залученням. Платформа є продуктом корпорації Microsoft як частина набору служб інтеграції Azure. Програми Logic Apps налаштовують робочі процеси, використовуючи низку API, доступних як конектори. Ці з'єднувачі виконують послідовність дій, визначених в робочому процесі, щоразу, коли запускається тригер. Налаштовувати робочі процеси можна за допомогою візуального дизайнера без написання коду.

SAP Cloud Platform Integration (CPI) початково була створена для допомоги з'єднання SAP програм з сторонніми програмами. Згодом CPI було перейменовано в SAP BTP (Business Technology Platform) та поділено на різні продукти – Integration Suite і Extension Suite. Для цього порівняння буде використано Integration Suite.

Alumio iPaaS — це хмарна платформа, що дозволяє об'єднувати системи, SaaS, хмарні додатки у локальних і хмарних середовищах. Створення інтеграцій, керування та моніторинг потоками даних можливі без коду через веб-інтерфейс (хоча все ж платформа є low-code).

Загальний огляд платформ представлено в таблиці 3.1 [14]. Показник складності характеризує рішення за рівнем залученості коду. Low-code та no-code інструменти знижують технічний поріг для створення інтеграцій, low-code рішення забезпечують розширені функції для створення складних інтеграцій та гнучких конфігурацій, в той час як no-code спрощують інтеграцію для нетехнічних осіб. Деякі інтеграційні рішення дозволяють підключати лише хмарні програми, інші також інтегрують локальні системи. Веб-інтерфейс iPaaS платформ може налаштовувати інтеграцію за принципом «drag-and-drop» (просто перетягуючи параметри разом) або «click-and-configure» (вибір інтеграцій з випадального списку з кількома параметрами). Показник ціни оцінює рішення на основі їхньої економічної релевантності в контексті запропонованих функцій.

Таблиця 3.1.

Загальний огляд iPaaS платформ
Microsoft Azure Logic Apps, SAP CPI та Alumio

Ознака	Microsoft Azure Logic Apps	SAP CPI	Alumio
Складність	No-code	Low-code	Low-code/ No-code
Інтеграція	Хмара	Хмара	Хмара

Продовження таблиці 3.1.

Інтерфейс	Click-and-configure з drag-and-drop опцій в робочих процесах	Drag-and-drop	Click-and-configure
Ціна	Ціна на основі функцій і використання. Від 140 євро на місяць.	Ціна на основі використання. Корпоративна угода про інтеграцію в хмару. Від 800 євро на місяць	Ціна на основі використання та завдань. Від 1200 євро на місяць

Інформаційна панель iPaas є важливою частиною платформи інтеграції, за допомогою яких користувачі можуть додавати, розгортати і управляти інтеграціями. SAP integration suite має сучасний загальний дизайн (зображений на рисунку 3.1). Overview сторінка включає моніторинг помилок, про які користувачі можуть отримувати сповіщення електронною поштою. Застосунок містить приховані меню, що дещо ускладнює інтуїтивне розуміння програми. Функція перетягування зручна у використанні.

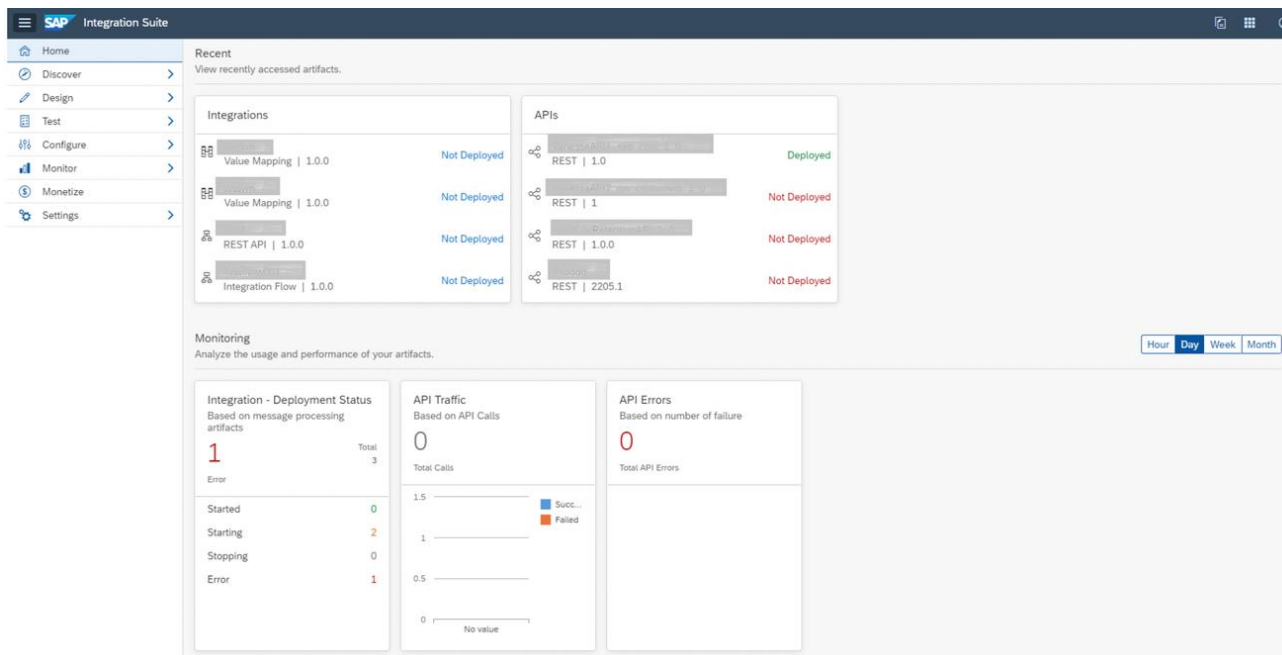


Рис. 3.1. UI Overview сторінки SAP integration suite

Завдяки готовим операціям візуальному конструктору Logic App, користувачі можуть швидко створювати робочі процеси, що інтегрують та керують їхніми програмами. Інтерфейс Microsoft Azure Logic Apps (представлений на рисунку 3.2) інтуїтивно більш зрозумілий, ніж SAP CPI. UI використовує сучасний макет з використанням контрастних кольорів, що гарно розмежовує елементи та робить візуально простішими no-code інтеграції. Робоча область має назву «Logic app designer» і складається з початкового елементу – триггеру та послідовності дій. Кожен блок виконується послідовно, що відображено стрілками. Якщо дія, що використовується, містить додаткові налаштування, то їх можливо задати у відповідних текстових полях розгорнувши блок. Також підтримується функціонал випадаючих списків, що можуть містити статичні або динамічні варіанти. Динамічні випадаючі списки завантажують можливі варіанти з відповідної системи, що дозволяє суттєво полегшити налаштування, оскільки можливо бачити реальні дані що є в системі користувача. Також для створення робочих процесів можна використовувати drag-and-drop функцію для додавання та зміни порядку виконання дій.

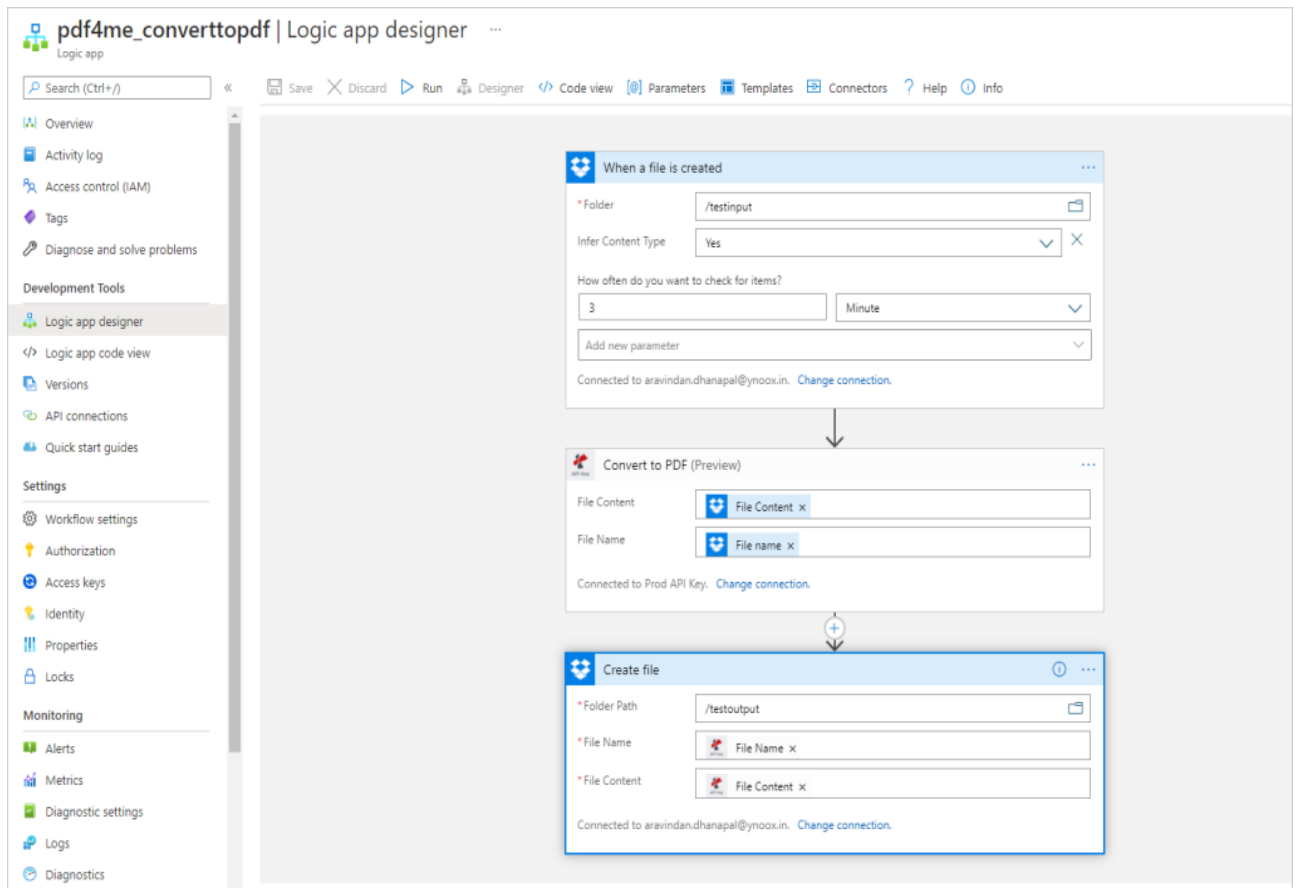


Рис. 3.2. UI Logic App Designer сторінки Microsoft Azure Logic Apps

Alumio має сучасний дизайн інтерфейсу (рисунок 3.3), де розташовані усті інтеграції, потоки даних, вікно моніторингу останніх процесів для виявлення помилок та конфліктів. Налаштування інтеграцій обирається натисканням на необхідну опцію, що доволі зручно.

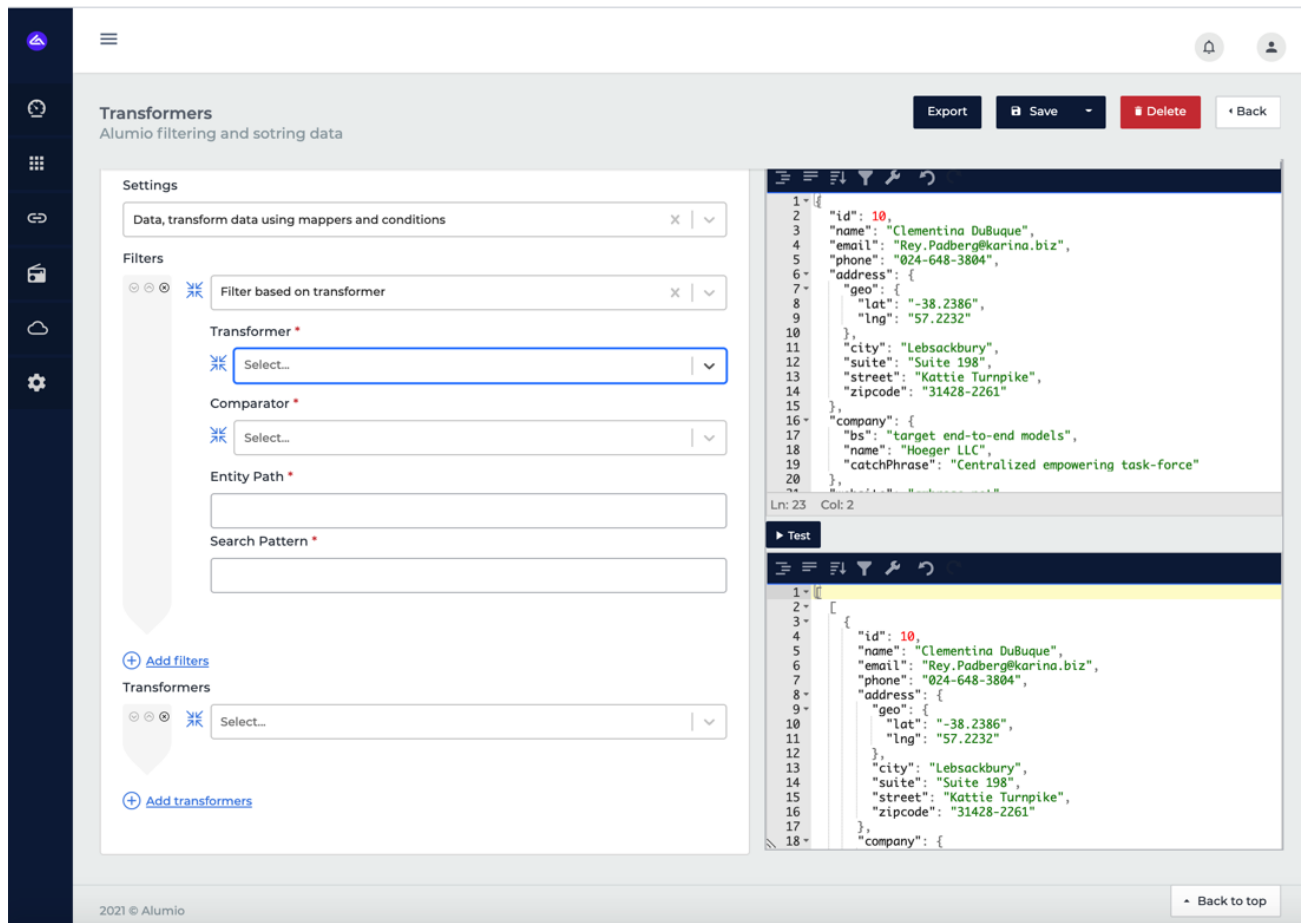


Рис. 3.3. UI сторінки налаштування робочих процесів Aluminio

Ключові функції рішення для інтеграції Microsoft Azure Logic Apps ДОЗВОЛЯЮТЬ:

- управління API;
- XML веб-служби;
- портал розробника;
- перевикористання міграцій;
- міграція даних;
- моніторинг стану мережі;
- контроль продуктивності;
- підтримка корпоративної інтеграції та сценаріїв B2B;
- візуалізація робочих процесів.

Основні функції інтеграційного рішення SAP CPI включають:

- управління API;
- інтеграція business-to-business;
- керована подіями архітектура.

До основних функцій Alumio належать:

- багаторазові інтеграції API;
- інтеграція хмарних даних;
- веб-хуки;
- керування документообігом.

Потужне проміжне ПЗ сучасних iPaaS рішень значно перевершують інтеграційну шину даних. Простота адаптації, розробки та навчання є основними факторами швидкості пристосування користувачів до iPaaS інструментів. У таблиці 3.2 представлено перелік основних можливостей, що може стати вирішальним фактором при виборі платформи.

Таблиця 3.2.

Ключові функції iPaaS платформ
Microsoft Azure Logic Apps, SAP CPI та Alumio

Функція	Microsoft Azure Logic Apps	SAP CPI	Alumio	Показник
Попередньо готові конектори	✓	✓	✓	Попередньо налаштовані з'єднання між двома програмами чи системами мінімізує зусилля необхідні для конфігурації з'єднання

Продовження таблиці 3.2.

Веб-сервіси	✓	✓	✓	Для створення зв'язку між ПЗ та системами потрібно інтегрувати веб-сервіси, що містять усі послуги, які пропонує ПЗ, та дозволяє комунікувати між собою через мережу
Управління даними	✓	×	✓	Керування даними – це процес зберігання, упорядкування або підтримки даних, створених організацією
EDI	✓	✓	✓	Електронний обмін даними – це затверджений формат, який забезпечує обмін бізнес-інформацією
ETL	✓	×	✓	Інструмент ETL (вилучення, передача та завантаження) забезпечує ефективну міграцію даних, постійний доступ до них і їх використання
Multiple data sources	✓	✓	✓	Функція кількох джерел даних забезпечує централізацію інформації з різних баз даних в одному

				вихідному сховищі
--	--	--	--	-------------------

Продовження таблиці 3.2.

Безпека даних	✓	✓	✓	Ключовою необхідністю іPaaS інтеграцій є беззупинна передача даних між інтегрованими системами, тому платформи мають гарантувати безпеку інформації
Compliance	✓	✓	✓	Дотримання основних нормативних актів щодо конфіденційності даних (GDPR і CCPA) є показником достатньої безпеки даних та відповідних заходів відповідальності
Управління помилками	✓	×	✓	Помилки та виключення інтеграції впливають на ефективність бізнес процесів. Тому можливість виявлення та моніторинг помилок інтеграції або конфліктів API скорочує час та вартість усунення несправностей

Продовження таблиці 3.2

Довідкова служба	✓	✗	✓	У випадку виникнення питань з впровадження чи застосування рішень провайдери пропонують функції електронної пошти/довідкової служби для того, щоб мати змогу комунікації з персоналом, що обслуговує клієнтів
База знань	✓	✓	✓	Бібліотека самообслуговування з різною інформацією про послуги, що надаються (блоги, відео-портал, форуми, навчальні посібники)

3.2.2. Порівняння Prismatic, Tray.io та Paragon

Prismatic, Tray.io та Paragon – альтернативні інтеграційні платформи для підключення різних програм та автоматизації робочих процесів.

Prismatic – це перспективний вбудований iPaaS, який в основному обслуговує SaaS-бізнес.

Ключовими характеристиками Prismatic є:

- Prismatic зосереджений на створенні єдиного середовища для створення, розгортання та керування інтеграціями.
- Prismatic має візуальний інтерфейс для проектування та автоматизації

робочих процесів з низьким залученням коду.

- Підтримує готові конектори для популярних служб та програм.
- Пропонує простий підхід для інтеграції даних, що робить його доступним для нетехнічних користувачів також.
- Користувачі можуть створювати індивідуальні інтеграції для забезпечення своїх вимог завдяки гнучкості платформи.
- Пропонує розгортання та підтримку інтеграції для легкого керування обліковими даними, налаштуваннями та підтримкою.
- Вбудований клієнтський досвід із великою кількістю пропозицій для самостійної інтеграції та інструментами самообслуговування.

Prismatic пропонує широкий набір вже готових конекторів зі своїми діями, проте зобов'язує дотримання контракту даних. Зазвичай цей контракт сильно відрізняється від системи до системи і тому зіставлення відповідних даних є унікальним для кожних двох систем і часто призводить до помилок у логіці інтеграції. На рисунку 3.4 видно, що попередньо створений компонент Salesforce має дію «Створити запис», але він дозволяє лише довільне введення тексту для типів записів і не пропонує спосіб посилання на поля, що містяться в цих типах. Тому розробники повинні заглиблюватись у документацію Salesforce API. Це робить абстракцію від API менш ефективною.

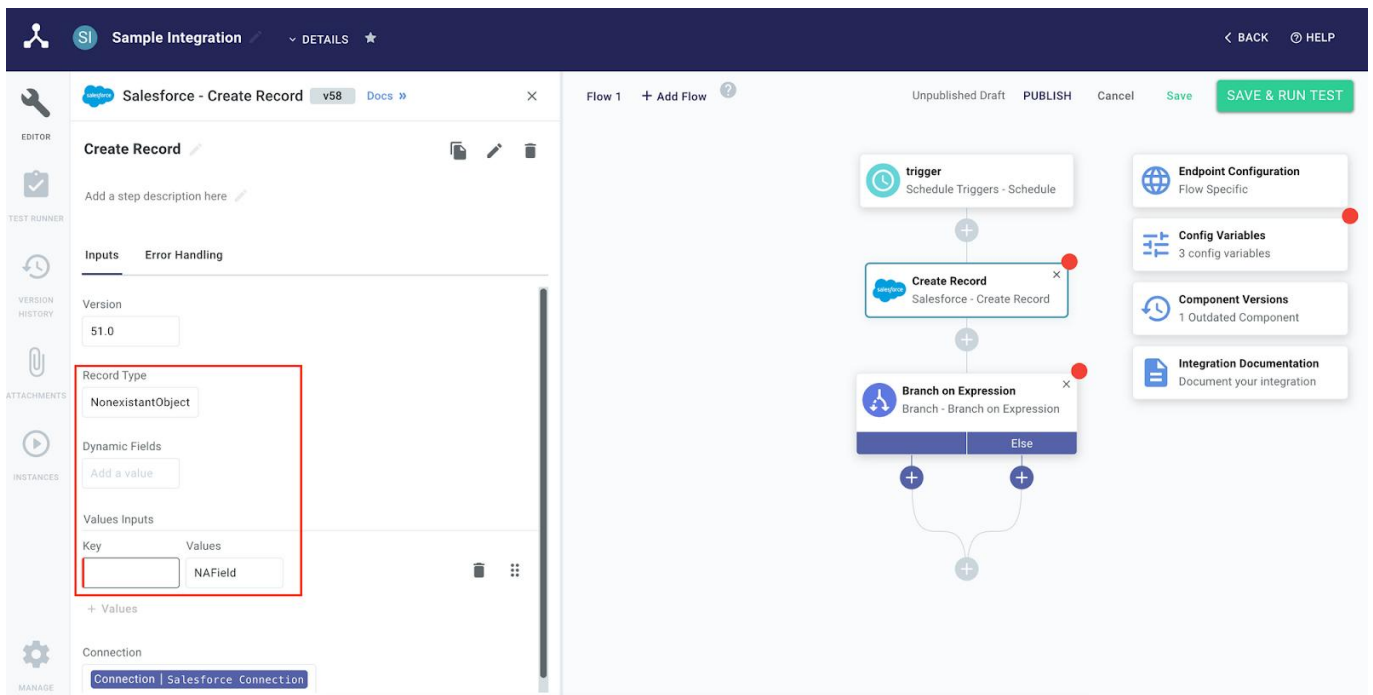


Рис. 3.4. Конструктор налаштування інтеграцій в Prismatic

Відповідно до документації Prismatic, користувачі не можуть напряму посилатися на вихідні дані з попередніх кроків, а мають використовувати власно розроблені обхідні шляхи. Це створює певну проблему масштабованості інфраструктури Prismatic. Можливість зіставляти конкретні фрагменти даних із полями на наступних кроках робочого циклу має вирішальне значення для економії часу розробки, незалежно від того, чи це дані з поточної системи, чи надійшли з стороннього webhook.

Однак загалом досвід розробників у Prismatic неоднозначний. Є деякі функції, що зручні для розробників, та на які варто звернути увагу. Розробники можуть створювати власні компоненти, що дає змогу їм інтегруватись із сторонніми програмами через http. Також можна встановити змінні налаштування, які є специфічними для конкретного клієнта, таким чином надаючи їм унікальний екземпляр інтеграції.

Та все ж найслабшою частиною Prismatic є його конструктор робочих процесів, який не дуже інтуїтивно зрозумілий та бездоганий.

Tray – ще одна провідна платформа автоматизації робочого процесу, що має інструменти на основі RPA (роботизовані процеси) для вирішення ситуації використання вбудованого iPaaS.

Основними характеристиками Tray.io є:

- Гнучка інтеграційна платформа з акцентом на автоматизацію та проектування складних робочих процесів, що включають велику кількість дій та використання умовних операторів.
- Великий вибір конекторів для різних програм та служб.
- Візуальний конструктор для налаштування процесів з низьким залученням коду.
- Забезпечення легкої масштабованості за допомогою гнучкої інфраструктури та
- Можливість використання власних стратегій інтеграції, що будуть забезпечені надійним керуванням, безпекою та засобами контролю.
- Наявність великої кількості готових умовних операторів та циклів.

Найбільша проблема Tray полягає в тому, що план розвитку продукту зосереджений на їхньому основному внутрішньому продукті автоматизації, а не на їхніх вбудованих iPaaS рішеннях. Крім того, платформа не підтримує спеціальні конструктори для користувацьких інтеграцій, які б дали можливість розгорнути невідтримувані інтеграції.

Незважаючи на те, що Tray забезпечує базове ведення журналів і можливість легко повторити робочі процеси, існує недостатня увага до деталей щодо обробки помилок і функцій стійкості, таких як автоматичний повтор. Коли зовнішні служби видають помилку, як-от досягнення обмежень частоти під час завдання масового оновлення, механізм робочого циклу Tray часто припиняє повторювати спроби та завершує робочий процес.

Крім того, відсутність керування версіями на їхній платформі також створює проблеми під час усунення проблем, ускладнюючи відкат оновлення/випуску робочого циклу.

Tray Embedded не надає можливості локального розгортання. Тому клієнтські дані завжди потрапляють в їхню хмарну інфраструктуру перед подальшою інтеграцією. Якщо інфраструктура платформи вийде з ладу, це може спричинити витік, втрату чи пошкодження клієнтських даних.

Paragon – це вбудована інтеграційна платформа, яка дає змогу розробникам B2B SaaS продуктів випускати інтеграції, орієнтовані на клієнтів.

Ключові характеристики та можливості платформи Paragon:

- Сучасний та інтуїтивний візуальний інтерфейс для проектування та автоматизації інтеграцій та робочих процесів.
- Широкий вибір готових конекторів до популярних програм та баз даних.
- Можливість одночасної співпраці різних команд над інтеграційними проектами.
- Наявність можливості форматування даних та їхня синхронізація для забезпечення безперебійного потоку інформації між системами.

Власний інтерфейс кінцевого користувача платформи Paragon надає своїм клієнтам повністю безпечний та нативний користувацький досвід. Це забезпечується тим, що платформа немає жодних переадресацій та iFrame-ів. Користувачі мають змогу використовувати настроюваний готовий компонент Javascript (що називається Connect Portal), який стане вбудованим для програм. Paragon постачає розширюваний конструктор для робочих процесів, який є інтуїтивно зрозумілим (інтерфейс зображено на рисунку 3.5) та має багато функцій, щоб забезпечити можливість створення будь-якого сценарію використання. Платформа достатньо проста у використанні попередньо створених абстракцій через сторонні API, а також надає можливість надсилати запити до будь-якої кінцевої точки. Крім того, попередньо створений інтерфейс користувача дає змогу легко надавати конфігурацію, визначену клієнтом, наприклад, вибір списку вибору або динамічне (спеціальне) відображення об'єктів і полів.

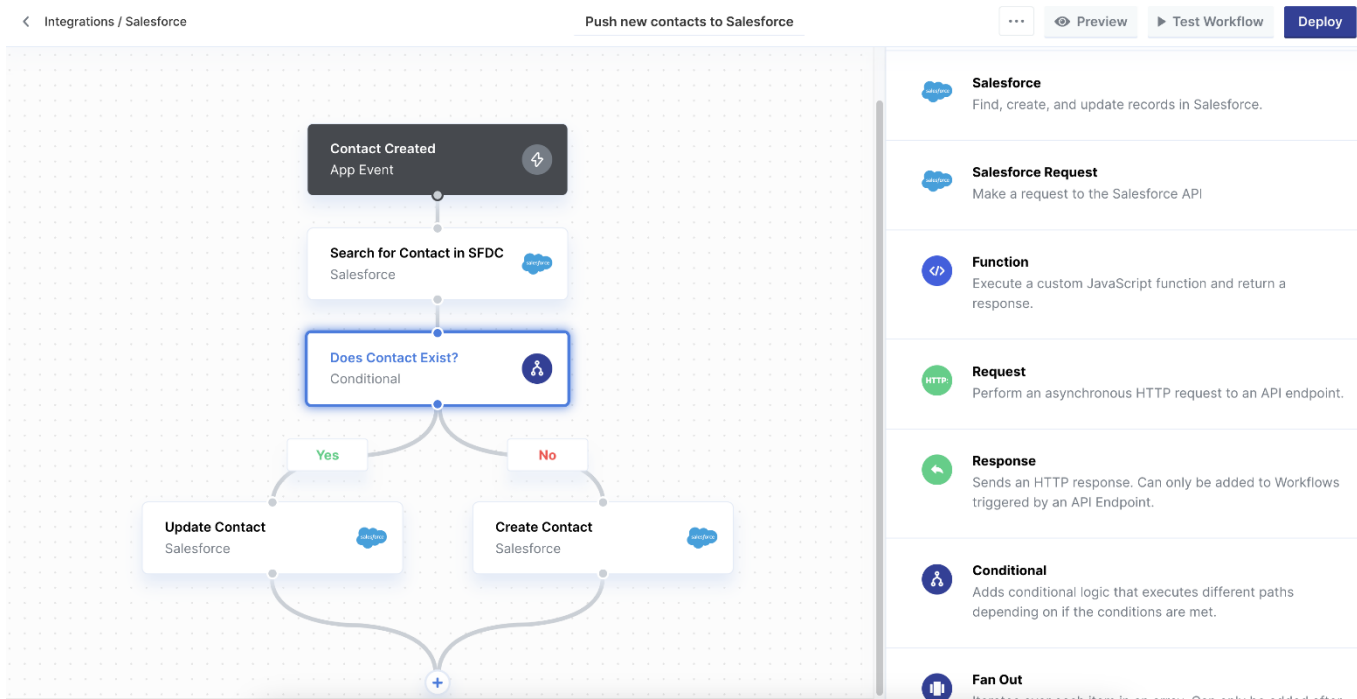


Рис. 3.5. Конструктор налаштування інтеграцій в Paragon

Paragon має гарний інструмент для відслідковування історії завдань, що представлений у вигляді журналу виконання кожного кроку робочого процесу, статусу виконання, а також має можливість детально відтворити будь-яке виконання окремого робочого процесу. Тому платформа здатна значно спростити тестування робочих процесів під час створення інтеграцій та процес усунення помилок завдяки гарній візуалізації процесів.

З технічної точки зору основна інфраструктура Paragon є надійною та масштабованою. Платформа здатна обробляти мільйонів запитів одночасно, а це означає, що рішення буде легко адаптуватись до масштабування та додавання нових інтеграцій. До значних переваг цього iPaaS також можна віднести можливість самообслуговування та локального розгортання, що допомагає впроваджувати власні стандарти безпеки.

У таблиці 3.3 перелічено основні функції, які притаманні цим платформам.

Таблиця 3.3.

Ключові функції iPaaS платформ
Prismatic, Tray.io та Paragon

Функція	Prismatic	Paragon	Tray.io
Конструктор інтеграції з низьким кодом	✓	✓	✓
Вбудований інтеграційний marketplace	✓	✓	✓
Попередньо зібрана бібліотека конекторів	✓	✓	✓
Користувацькі конектори	Повний компонент SDK	Легкі впаери для REST API	Легкі впаери для REST API
Вбудований конструктор інтеграції для кінцевих користувачів	✓	×	×
Обробка складних сценаріїв інтеграції	✓	×	✓

Продовження таблиці 3.3.

Можливість інтеграції з існуючими робочі процеси CI/CD	✓	×	×
Наявність зовнішніх журналів аудиту	✓	×	✓
Можливість використання Third Party сервісів	×	✓	×
SaaS / Web сервіси	✓	✓	✓

3.3. Правила забезпечення безпеки та надійності використання iPaaS рішень

Із збільшенням використання платформи iPaaS та API для обміну даними між різними платформами підвищився рівень кіберзагроз. Забезпечення безпеки та надійності є першочерговими вимогами при використанні iPaaS рішень для автоматизації бізнес процесів. В питаннях безпеки компанії не можуть покладатися лише на провайдерів інтеграційних послуг, оскільки вони є кінцевою відповідальною стороною за якісне та безпечне надання послуг. Тому організаціям потрібно впроваджувати рекомендовані практики кібербезпеки своїх систем та даних.

Нижче наведено основні правила безпеки та кращі практики по їх досягненню:

- Шифрування даних: необхідно шифрувати дані під час передачі та під час збереження, щоб захистити конфіденційну інформацію. Для цього необхідно використовувати протоколи шифрування SSL/TLS для передачі даних. Для зберігання даних в системах або базах даних платформи можна використовувати механізми шифрування, що надають iPaaS провайдери. Загальними механізмами шифрування даних, що використовують iPaaS є:

- Transparent Data Encryption (TDE) шифрує всю БД на рівні файлів. Технологія забезпечує шифрування даних під час запису та розшифрування під час читання. TDE зазвичай використовуються в базах на стороні iPaaS.

- Advanced Encryption Standard (AES) – алгоритм симетричного шифрування, що захищає дані в БД та файлових системах.

- Key Management сервіси, що пропонують платформи iPaaS, створюють, зберігають та керують ключами шифрування.

- Hardware Security Modules – це спеціальні апаратні пристрої, що гарантують, що ключі зберігаються в захищеному від втручання середовищі.

- Шифрування на рівні файлу передбачає шифрування окремих об'єктів даних, а не всього сховища.

- Шифрування БД часто використовують для захисту інформації, що зберігається в iPaaS базах даних.

- Маскування даних передбачає заміну конфіденційних даних фіктивними або спеціальними символами. Метод використовується іPaaS платформами під час попереднього перегляду або обміну даними з окремими користувачами.

- Identity and Access Management (IAM): для контролю доступу необхідно використовувати надійну аутентифікацію та регулярно переглядати та оновлювати дозволи доступу за принципом найменших привілеїв. Прикладом такого механізму може бути керування на основі ролей (RBAC – Role-Based Access Control), який гарантує, що лише авторизовані користувачі можуть переглядати або змінювати конкретні дані.

- Дотримання стандартів та правил галузі: необхідно обирати провайдера іPaaS з сертифікатами відповідності галузі вашої компанії та налаштовувати функції відповідно до вимог відповідності.

- Безпека API: для інтеграції потрібно використовувати безпечні API з безпечними механізмами аутентифікації, наприклад, такими як ключі API, OAuth, SAML, MFA, JWT, тощо. Для запобігання зловживанням рекомендується реєлювати швидкість API.

- Моніторинг та аудит: для швидкого виявлення порушень безпеки слід використовувати журнали аудиту функцій іPaaS в режимі реального часу та регулярно переглядати їх.

- Механізми обробки помилок та функції повтору: слід запровадити механізми автоматичної обробки тимчасових виключень, а також сконфігурувати політику повторних спроб для обробки помилок.

- Оновлення іPaaS платформ та їхніх компонентів: постачальники іPaaS постійно змінюють механізми безпеки та додають нові рівні, тому потрібно тримати свою системи в тому стані, що пропонують провайдери.

- Кінцеві точки інтеграції: для безпечної інтеграції рекомендують використання захищених протоколів, сертифікатів для комунікації та правильної авторизації/аутентифікації до кінцевих точок інтеграції.

До ключових правил надійності можна віднести наступні механізми та

правила для їхнього забезпечення:

- Угоди про рівень обслуговування (SLA – Service Level Agreements): вибір постачальника має безпосередньо залежати від ваших у потреб у SLA щодо продуктивності, підтримки та часу безвідмовної роботи.

- Суверенітет даних: компанії мають чітко розуміти та контролювати процес збереження даних та обирати провайдерів з центрами обробки даних, які відповідають їхнім вимогам резистентності даних.

- Відновлення даних та резервне копіювання: слід регулярно створювати резервні копії важливих даних та налаштувань для можливості їх відновлення у випадку втрати чи пошкодження. Можна використовувати процедури відновлення, що дозволить піднімати дані автоматично для безперервної роботи у разі помилок.

- Резервування та висока доступність: для забезпечення безперервної роботи потоки інтеграції варто зберігати та виконувати на різних серверах або регіонах, використовуючи функції високої доступності.

- План реагування на збої: необхідно створити план, який би визначав обов'язки та канали комунікації у разі інцидентів у роботі служби.

- Тестування: перед впровадженням інтеграції слід провести інтеграційне та наскрізне тестування.

Як було зазначено раніше правильна аутентифікація є основою безпеки під час іPaaS інтеграцій. Правила аутентифікації в платформах визначають спосіб аутентифікації користувачів, служб та систем, і визначають надання доступу до функцій чи ресурсів платформи. Завдяки цьому можна забезпечити цілісність даних, що використовуються для інтеграції. Платформи іPaaS вимагають авторизації з дійсними обліковими даними, інколи використовуючи додаткові фактори автентифікації – багатофакторну (MFA) або OAuth чи LDAP. Нижче представлено основні способи аутентифікації та правила їхнього використання, а також загальні правила аутентифікації, що використовують іPaaS платформи:

- Аутентифікація API ключами: під час взаємодії з API платформи програми повинні передавати дійсний API ключ. іPaaS платформи генерують API ключі, що є унікальними, вони є параметрами запитів API.

- Аутентифікація на основі токенів: для безпечної авторизації можна використовувати JSON токени (JWT) або Bearer токени, які повертають iPaaS платформи під час успішної авторизації. В них закодована інформація про користувача (або програму), його ролі та доступи. Потім наступні запити в платформі вимагають ці токени.

- Аутентифікація на основі OAuth: платформи iPaaS мають функцію підтримки OAuth 2.0 для делегованої авторизації. Тобто спочатку програми або користувачі аутентифікуються через сервер для авторизації, який повертає їм токени для доступу до захищених ресурсів в запитах API.

- Single Sign-On (SSO): користувачам надається можливість пройти автентифікацію один раз та безперешкодно отримати доступ до кількох інтегрованих та підключених систем без потреби в повторному вході, адже iPaaS платформи інтегруються з SSO рішеннями, наприклад SAML або OpenID Connect.

- Аутентифікація сертифікатів: платформи підтримують автентифікацію клієнтів за допомогою цифрових сертифікатів як підтвердження особи для авторизації.

- Мультифакторна аутентифікація (MFA – Multi-Factor Authentication): iPaaS використовує MFA для додаткового рівню безпеки за допомогою паролів, повідомлень або біометричної перевірки особливо для користувачів, що мають підвищені доступи.

- Списки білих/чорних IP-адрес: iPaaS платформи надають можливість клієнтам обмежувати доступ на основі IP-адрес. Списки з білими або чорними IP-адресами контролюють, яким IP дозволений або заборонений доступ.

- Керування сесіями: iPaaS платформи запобігають несанкціонованим втручанням, використовуючи методи безпечного керування сесіями. Вони забезпечують механізми контролю часу тривання сеансу, безпечну обробку токенів та захист від викрадення сесій.

Впровадження та дотримання цих правил може забезпечити безпечний, контрольований доступ до ресурсів та інтеграцій iPaaS, відповідно до політик безпеки організацій. Проте компаніям необхідно регулярно переглядати та

актуалізувати механізми авторизації, щоб унеможливити запобігання пристосування до загроз.

Висновки

Методологія проектування автоматизованих систем на основі iPaaS представляє собою структурований підхід до розробки інтегрованих рішень, який здатний надати можливість використовувати iPaaS рішення найбільш ефективно та залучати максимальну кількість функцій для автоматизації процесів. Використання принципів дизайну для використання iPaaS рішень в процесах інтеграції здатне закласти фундамент для розробки/автоматизації системи, що здатна функціонувати в мінливому середовищі та забезпечувати надійний та ефективний потік даних між різними програмами чи системами.

Проектування систем для залучення інтеграцій iPaaS дозволяє врахувати такі фактори, як безпека, ефективність, гнучкість, масштабованість. Етапи проектування автоматизованих систем на базі iPaaS платформ утворюють ітеративний та комплексний підхід до розгортання та підтримки таких систем. Ітеративна природа цього процесу дозволяє постійно вдосконалювати та адаптувати свої системи до сучасних технологій. Це стосується також факторів безпеки, адже рівень кіберзагроз постійно зростає.

Правила для безпечної інтеграції з використанням iPaaS рішень гарантують захист від потенційних загроз, правильну обробку помилок та безперебійну роботу систем та інтеграцій. Тому перш за все потрібно дотримуватись правил шифрування даних, використовувати безпечні механізми аутентифікації, закласти конфігурацію для моніторингу помилок та збоїв. Дотримання наведених заходів безпеки дозволить використовувати увесь потенціал iPaaS, забезпечуючи надійність своїх інтегрованих систем та створюючи стійке та високоефективне середовище.

Компанії не можуть повністю покладатися на провайдерів iPaaS послуг. Важливими задачами для них є контроль усіх процесів інтеграції для управління ризиками та забезпечення безперебійності роботи бізнесу. Проведення регулярних перевірок та оновлень рішень мають бути стратегією роботи автоматизованих систем.

Огляд інтеграційної платформи як послуги висвітлює ключові можливості та функції, що вони надають. На основі цих знань компанії здатні обрати постачальника iPaaS послуг серед їхнього широкого різноманіття. Динамічний розвиток цих рішень підкреслює необхідність в постійному аналізі ринку та оновленні рішень. В свою чергу методологія проектування з використанням iPaaS, дозволяє легше адаптуватися до нових тенденцій і викликів., забезпечуючи надійну структуру автоматизованих систем.

Успіх автоматизованих систем залежить від ретельного застосування цих методологій проектування, сприяння інноваціям, ефективності та бездоганній інтеграції.

РОЗДІЛ 4

ІНТЕГРАЦІЯ СИСТЕМ З ВИКОРИСТАННЯМ MS POWER AUTOMATE

Оскільки теперішній бізнес диктує умови необхідності в зменшення ручної роботи та оптимізації багатьох процесів, інтеграція стає незамінною складовою функціонування сучасних систем. Служба Microsoft Power Automate є гарним рішенням для автоматизації бізнес-процесів та впровадження інтеграцій через широкий спектр запропонованих динамічних рішень. До того ж може вважатися вдалим інструментом для початку роботи з автоматизацією інтеграцій через простий та інтуїтивний інтерфейс та велику добірку посібників та документації.

У наступних підрозділах буде запропоновано рішення системної інтеграції нескладних процесів використовуючи можливості Microsoft Power Automate та детальний огляд цієї платформи.

4.1. Загальний огляд Microsoft Power Automate платформи

Power Automate – це служба від Microsoft, яку використовують для автоматизації повторюваних процесів між різними системами та програмами. MS Power Automate в своїй основі використовує Microsoft Azure Logic App з додаванням певних надбудов та функцій. Корпорація Майкрософт створила цю службу, щоб допомогти автоматизувати рутинні ручні процеси в Microsoft Office 365 та Microsoft Azure та підвищити продуктивність. Програма за замовчуванням увімкнена в усіх застосунках Office 365.

Технологія здатна автоматизувати велику кількість систем, програм та сайтів за допомогою штучного інтелекту, а також автоматизації цифрових і роботизованих процесів (RPA – Robotic Process Automation).

Доступ до робочих процесів, що створені у Power Automate, та керування ними можна отримати з мобільних пристроїв через мобільну версію Power Automate, це корисна функція для забезпечення гнучкості управління.

Платформа є повністю хмарною програмою для автоматизації програм та БД, використовуючи вбудовані з'єднувачі [15]. Конектори можуть підключатися до

внутрішніх Microsoft служб або зовнішніх служб. Такі з'єднувачі, як SharePoint або Excel, завжди підключаються до внутрішніх служб. Однак щоразу коли з'єднувач залишає інфраструктуру Microsoft, платформа запускає серію перевірок, щоб забезпечити захист даних. Важливим фактом є те, що якщо дані системи зберігаються в інфраструктурі Microsoft, Microsoft контролює їх, але коли він переходить до зовнішньої інфраструктури, контроль делегується перевіреному видавцю.

Power Automate постачає близько 150 конекторів та пропонує приблизно таку ж кількість преміум з'єднувачів. До найпопулярніших конекторів платформи належать:

- Outlook.com;
- OneDrive;
- Salesforce;
- Google Drive;
- FTP;
- GitHub;
- Slack;
- Azure AD;
- Azure Event Grid;
- Trello;
- Google Calendar;
- SFTP - SSH;
- Approvals;
- Microsoft Transactions;
- Amazon SQS.

Power Automate можна порівняти з PowerShell, що працює в хмарі. Платформа теж є інструментом для створення сценаріїв, які виконують інструкції користувачів для автоматизації процесів.

Для успішного користування платформою слід запровадити потужні безпекові заходи, адже, по суті, якщо у користувача є доступ до середовища, він скоріше за

все має доступ до Power Automate. Робочі процеси Power Automate дозволяють створювати спеціальні інтеграції та автоматизовані процеси в Office 365 та сторонніх системах без втручання девелоперів. Хоча це рішення надає широкі можливості, в той же час воно має багато потенційно вразливих місць, оскільки потоки інтеграції можуть обходити політики безпеки та втрачати дані. Широка доступність і простота використання Power Automate робить його частково корисним інструментом для зловмисників. Виявлення підозрілого сценарію є складною задачею, адже реальні адміністратори системи створюють схожі інструкції. Прикладом такого кейсу може бути наступний сценарій: хакер входить в мережу та пише інструкції для пересилання всіх електронних листів від якось користувача на визначену електронну адресу. Ідентифікувати такий сценарій як зловмисний непросто, адже потрібно перевірити усі інструкції. Тому важливо правильно налаштувати механізми аутентифікації та моніторингу підозрілих дій.

Події виявлення, пов'язані зі зловживанням Power Automate, можна швидко виправити завдяки використанню функції Advanced Investigations від Vectra. Надбудова може сповіщати про підозріле створення потоку Power Automate.

Power Automate – корисний інструмент для автоматизації бізнес процесів, що забезпечити краще використання ресурсів та підвищення продуктивності. Прикладом бізнес-задач, які платформа дозволяє автоматизувати є:

- виставлення рахунків;
- керування документами;
- керування контрактами;
- навчання користувачів.

В контексті інтеграції даних Microsoft Power Automate пропонує компаніям універсальну та зручну платформу для автоматизації та оптимізації робочих процесів інтеграції. Нижче представлено основні переваги використання Microsoft Power Automate для інтеграції даних:

- low-code рішення для створення робочих процесів інтеграції даних та управління ними нетехнічним персоналом;
- широкий вибір конекторів для популярних служб та баз даних спрощує

процес інтеграції, дозволяючи підключатись до різних джерел даних та місць призначення;

- легка інтеграція з Microsoft 365 та Azure, що полегшує потік даних між Microsoft програмами та службами Azure з сторонніми системами, підвищуючи ефективність робочих процесів;

- можливість автоматизації робочих процесів для синхронізації та оновлення даних на основі попередньо визначених умов або тригерів;

- функції для форматування та маніпулювання даними, що дозволяє користувачам перетворювати різні представлення даних, застосовувати фільтри та виконувати операції, що є підтвердженням того, що дані мають потрібну структуру для інтеграції;

- підтримка інтеграції даних в режимі реального часу та за розкладом, що забезпечує гнучкість у керуванні оновленням даних;

- Power Automate забезпечує кросплатформну сумісність, здатен працювати на різних пристроях та операційних системах; тому користувачі можуть створювати інтеграції між системами незалежно від платформ, де вони розгортаються;

- платформа містить інструменти моніторингу та аналізу для ведення аналітики ефективності робочих процесів, виявлення потенційно небезпечних місць інтеграції та формування звітності по процесам інтеграції даних;

- технологія повністю відповідає стандартам безпеки Microsoft, що забезпечує безпечну обробку та зберігання конфіденційних даних;

- платформа надає можливості масштабування відповідно до зростаючих вимог компаній та може працювати з різними рівнями складності та кількістю інформації, незалежно від розмірів інтеграції;

- Power Automate пропонує гнучкі моделі ціноутворення, що дозволяє організаціям вибирати плани на основі своїх конкретних потреб.

Отже, компанії можуть оптимізувати процеси, покращити передачу та забезпечити безперервний потік даних між різними системами, використовуючи Microsoft Power Automate для інтеграції даних.

Як і будь-яке автоматизоване інтеграційне рішення, що має ряд переваг, Microsoft Power Automate несе потенційні ризики та недоліки, пов'язані з інтеграцією даних, наприклад:

- Power Automate дозволяє виконання простих та достатньо складних інструкцій для обробки та перетворення даних, але має обмеження для надто складних перетворень чи сценаріїв у можливостях платформи.

- Оскільки платформа є хмарною службою, користувачі мають змогу тільки обмеженого контролю над середовищем виконання, що може створювати проблеми для виконання інтеграцій з локальними системами.

- Незважаючи на те, що платформа має різні типи підписок, масштабна або ресурсомістка інтеграція може додати витрат, адже в ці тарифні плани не входить вартість за кількість пройдених циклів та додаткові функції.

- Використання Power Automate може вплинути на продуктивність інтеграції, особливо для великих наборів даних, оскільки платформа обмежує швидкість передачі даних.

- Платформа має обмеження щодо розміру даних, яку користувачі можуть інтегрувати. Для потреб у великих інтеграціях краще розглянути альтернативних провайдерів для керування інтеграцій з великими обсягами даних.

- Як було зазначено в перевагах Power Automate дотримується стандартів безпеки Microsoft, але для організацій, що інтегрують регульовані або надто конфіденційні дані, потрібно задовольняти більш специфічні вимоги до безпеки, що може стати проблемою через розгортання в хмарі. В такому випадку можливе використання надбудов від інших служб, які керують безпекою іншими шляхами.

- Сервіс не може задовольнити офлайн-інтеграції, адже повністю покладається на підключення до Інтернету.

- Користувачі можуть зіткнутися з проблемами в інтеграції з певними службами, якщо відсутні конектори до них. Певні конектори можуть мати обмежений набір функцій, що можуть бути використані.

- Створення складних сценаріїв потребує технічної команди, яка глибоко розуміє платформу.

- Користувачі можуть бути обмежені в параметрах налаштування локальних систем порівняно з хмарною інтеграцією.

- Налаштування керування версіями та змінами в Power Automate потребує ретельного планування та тестування змін робочих процесів, щоб уникнути втрати даних.

- Керування версіями та повернення/відміна змін у Power Automate може бути складним завданням, адже станом на зараз платформа підтримує систему контролю версіями. Користувачі повинні ретельно планувати та тестувати зміни робочих процесів, щоб уникнути небажаних наслідків. Відслідковувати зміни можна у звітах налаштувань робочих процесів, що не є надто зручним рішенням.

Після оцінки вимог до інтеграції компанії повинні врахувати такі фактори, як розмір даних, складність інтеграції, безпекові стандарти та вартість, перед вибором провайдера. Хоча Power Automate має багато інструментів для автоматизації процесів, можуть траплятися випадки, коли для вирішення більш вузьких проблем можуть знадобитися більш широкі рішення.

4.2. Побудова інтеграції Trello та Gmail за допомогою Power Automate

Незважаючи на те, що Power Automate є невдалим інструментом для вирішення складних процесів, він гарно підходить для демонстрації простої інтеграції. Налаштування інтеграцій в більшості iPaaS платформ використовує схожі сценарії, тому для розуміння основ процесу була обрана служба Power Automate.

Платформа Power Automate є достатньо ефективною для автоматичної інтеграції таких сервісів, як Trello та Gmail, адже має низку готових конекторів та функцій для цього.

4.2.1. Налаштування робочого процесу інтеграції в Power Automate

Робочий процес інтеграції в Microsoft Power Automate – це послідовність автоматизованих дій, які з'єднують різні програми, служби або системи, щоб забезпечити потік інформації між ними. Power Automate надає зручне середовище з низьким кодом для розробки та впровадження робочих процесів інтеграції. Цей

процес також полегшується існуванням широкого ряду готових конекторів та компонентів задач.

Користувачі можуть створювати два типи робочих процесів:

- авторизований RPA, який запускається, коли користувач увійшов у систему (гарно підходить для процесів, які все ще потребують втручання людини);
- автоматичний RPA, який запускається, коли користувач не увійшов у систему, а визначений тригер запускає робочий процес.

Робочі потоки Power Automate базуються на тригерах і діях. Потоки робочого столу використовуються для автоматизації завдань в Інтернеті або на робочому столі за допомогою Power Automate Desktop. Потоки бізнес-процесів базуються на визначеному наборі дій, які допомагають людям виконувати завдання швидше та з меншою кількістю помилок.

Кроки створення процесу інтеграції в Power Automate:

1. Виберіть тригер (створення тригера зображено на рисунку 4.1), який буде починати робочий процес. Тригером може бути якась подія (наприклад, створення якось об'єкта або електронний лист), розклад або ручний запуск.

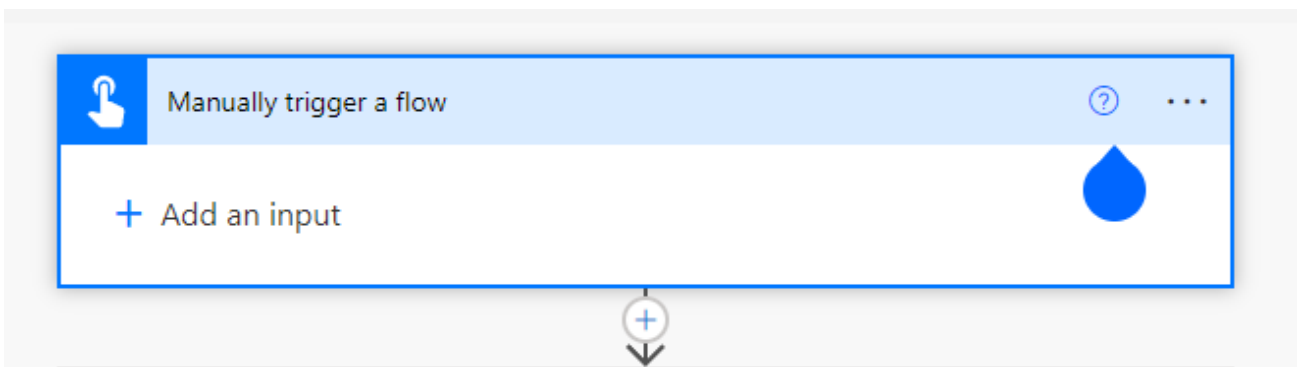


Рис. 4.1. Приклад створення ручного тригера

2. Додайте дію або набір дій, які слід виконати під час інтеграції (рисунок 4.2). Для цього спочатку необхідно обрати конектор, що є мостом між Power Automate та різними службами. Потім необхідно налаштувати параметри дій в списку параметрів можливих опцій. Ці параметри залежать від вимог інтеграції.

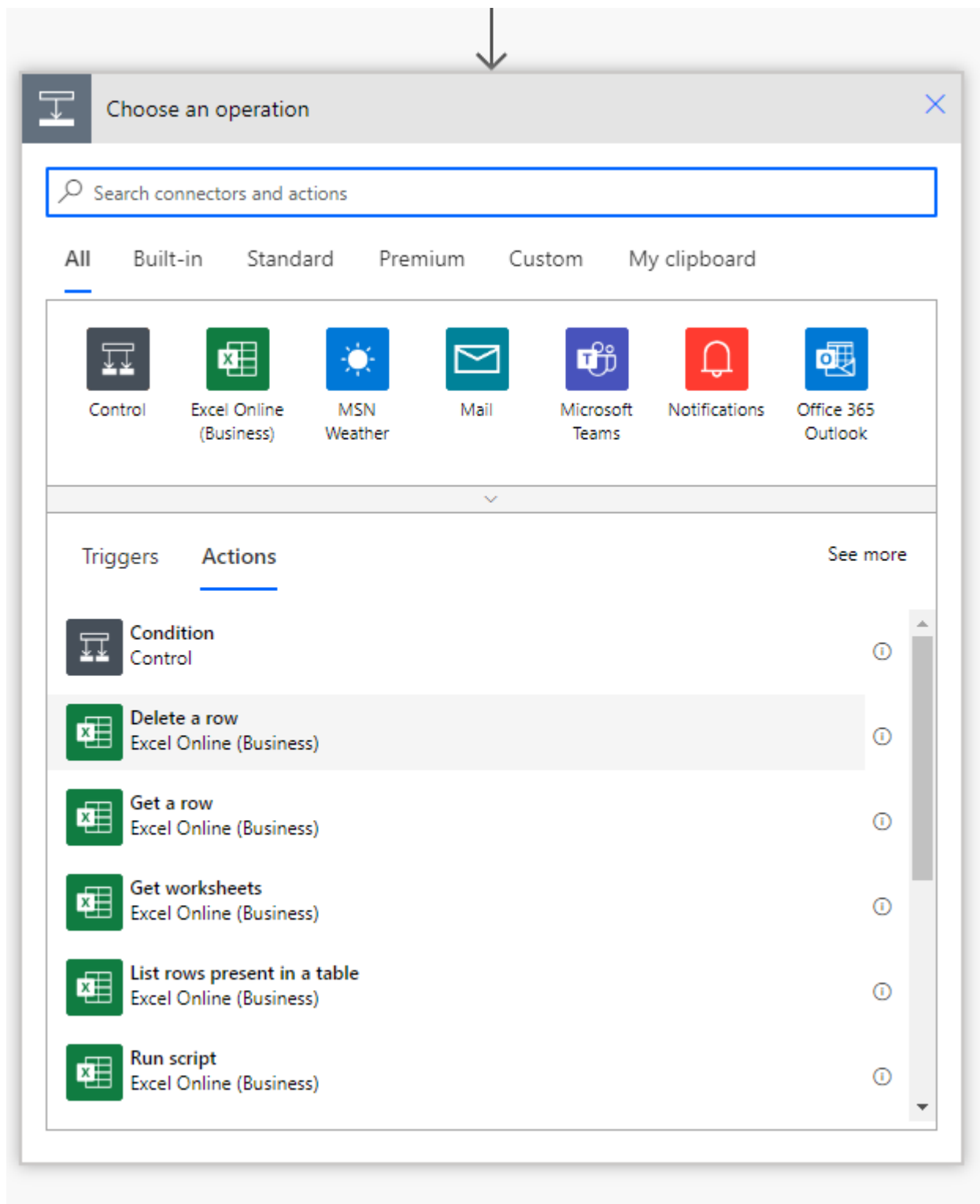


Рис. 4.2. Створення та налаштування дії

3. Реалізуйте механізмів для обробки помилок шляхом додавання дій, які запускаються у разі помилкових ситуацій.

4. Протестуйте інтеграцію перед розгортанням за допомогою. Функцій тестування, що імітують роботу описаного процесу та результат його виконання.

Після успішного тестування можна розгортати інтеграцію.

5. Налаштуйте процеси моніторингу для продуктивності робочого процесу, журнал виконання (зображений на рисунку 4.3) тощо.

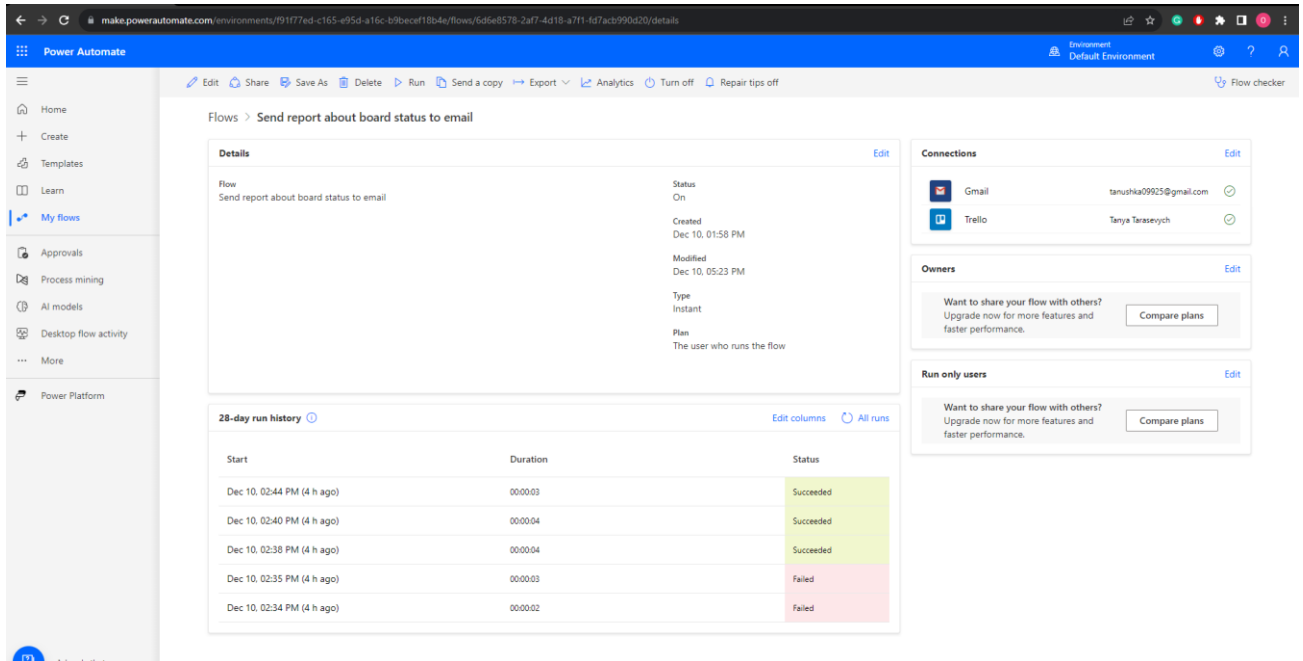


Рис. 4.3. Overview сторінка робочого процесу

До ключових елементів робочих процесів Power Automate належать наступні поняття:

- Змінні та вирази: використовуються для зберігання та обробки даних у робочому процесі. З їхньою допомогою можна динамічно виконувати обчислення та перетворення даних.

- Умовні оператори та цикли: використовуються для логіки розгалуження робочого процесу. Умовні оператори дозволяють будувати робочий процес за правилом «зробити [дія], якщо [умова]». Тобто можна налаштовувати виконання різних дій на основі конкретних критеріїв. Для керування потоком також можна використовувати цикли, щоб перебирати набори даних.

- Паралелізм: для оптимізації робочих процесів та підвищення ефективності можна використовувати функцію паралельного виконання, дозволяючи виконання кількох дій одночасно.

- Зони видимості: налаштувати дії варто в межах областей для інкапсуляції пов'язаних задач, це дозволить легше розуміти та обслуговувати робочий процес.

- Користувацькі конектори: коли стандартні конектори недоступні або не задовольняють усі вимоги можна використовувати користувацькі конектори, де можна налаштувати АРІ з іншими службами.

Виконуючи ці кроки та враховуючи ключові концепції, ви можете розробити ефективні робочі процеси інтеграції в Microsoft Power Automate для автоматизації бізнес-процесів і оптимізації потоків даних у різних програмах і службах.

4.2.2. Опис інтеграції Trello та Gmail в Microsoft Power Automate

Trello – це інструмент, який дає змогу командам керувати проектом та його робочим процесом, відстеженням завдань. Приклад дошки Trello зображена на рисунку 4.4.

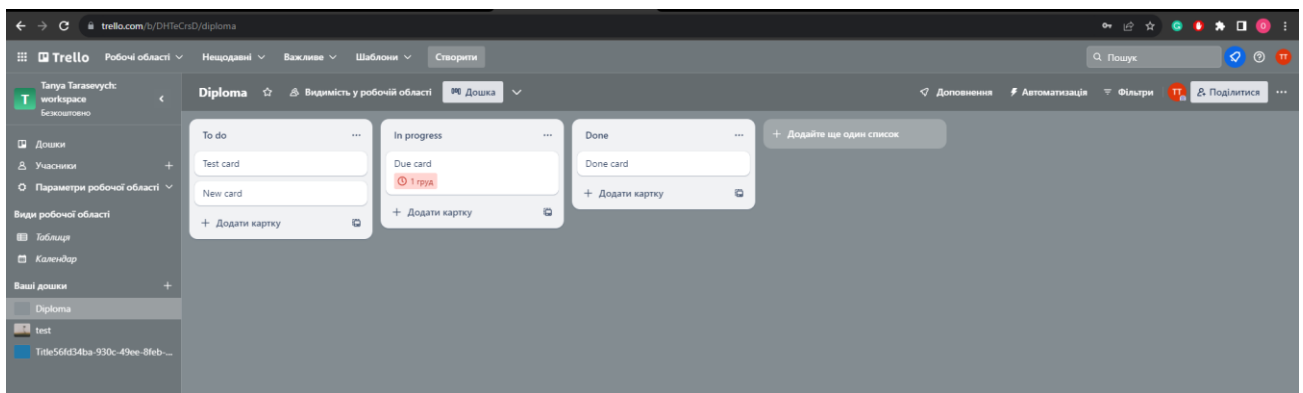


Рис. 4.4. Trello дошка

Для прикладу було автоматизовано два процеси інтеграції вибраних служб – «Notify email when new card is created» та «Send report about board status to email».

Авторизація в Trello здійснюється з використанням OAuth 2.0. При додаванні з'єднувачів до Trello Power Automate запитує авторизацію до служби (рисунок 4.5). Переглянути або додати нове підключення можна викликавши контекстне меню

процесу. Також всі використання робочим процесом підключення зображенні у вікні Connections (рисунок 4.6).

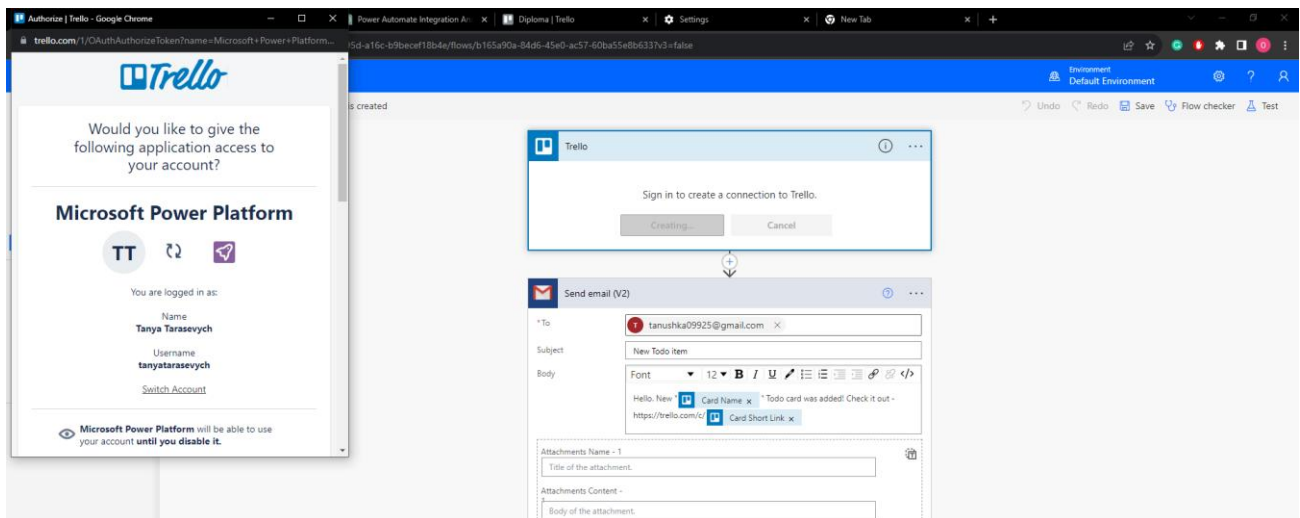


Рис. 4.5. OAuth 2.0 авторизація до конекторів

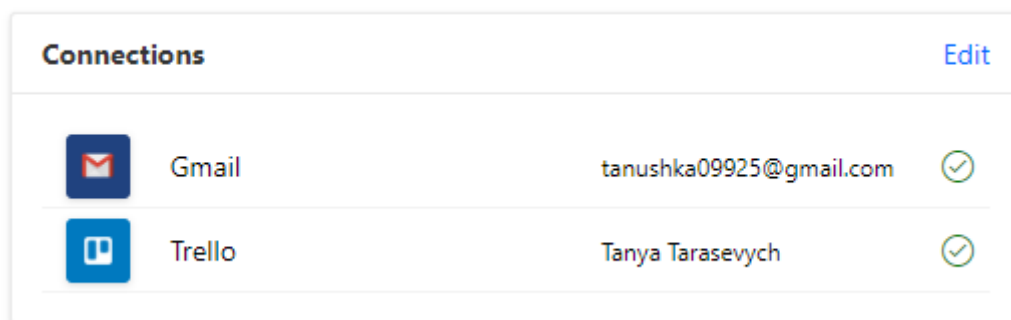


Рис. 4.6. Вікно підключень робочого процесу

Робочий процес «Notify email when new card is created» інтеграції складається з двох елементів – триггеру та дії (зображені на рисунку 4.7). Логіка інтеграції – відправляти повідомлення, коли на дошці Trello з'являється нова картка. Для цього обираємо тригер «When a new card is added to a board» та вибираємо робочу дошку Trello. Далі додаємо дію «Send email», в якій налаштовуємо текст листа, кому він буде відправлений.

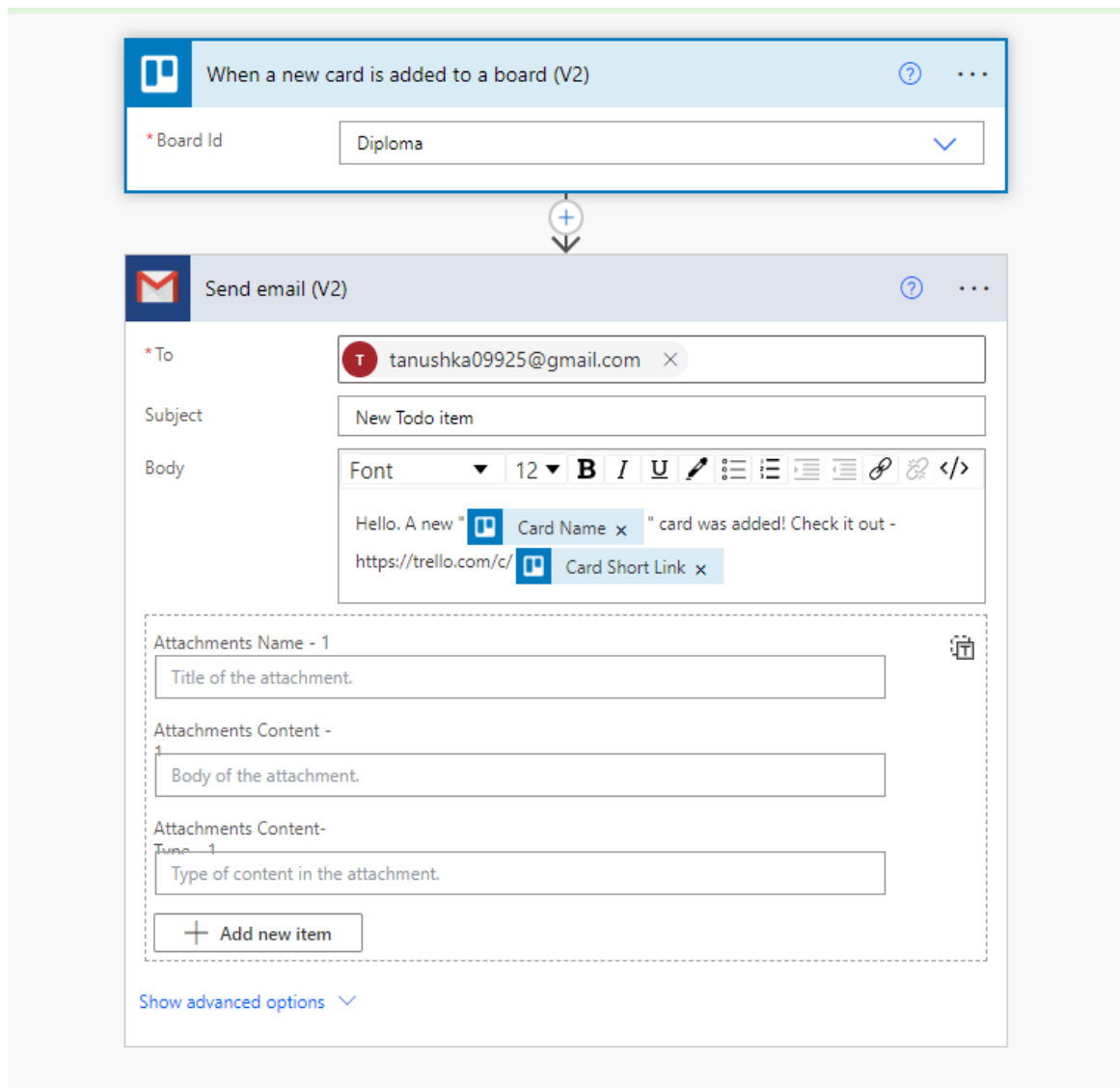


Рис. 4.7. Інтеграція «Notify email when new card is created»

Після того, як на дошку додана нова картка, на вказану електронну адресу надходитиме нотифікація про це (текст повідомлення зображено на рисунку 4.8). Відправником повідомлення буде користувач, який вказаний в підключенні.



Рис. 4.8. Приклад електронного листа

Робочий процес «Send report about board status to email» є дещо складнішим та використовує більше компонентів.

Загальний сценарій інтеграції працює наступним чином (налаштування процесу зображено на рисунку 4.9).

- Precondition: На дошці Trello є три колонки/списки – ToDo, In Progress, Done, в кожній з яких є картки-задачі. Задачі мають очікуваний термін виконання.
- Condition: Користувачу необхідно отримати звіт про стану дошки Trello, де буде вказано загальну кількість карток кожного з списків (ToDo, In Progress, Done), відсоткове співвідношення закритих карток до усіх задач та протерміновані картки.
- Action: Користувач вручну запускає процес формування та відправки звіту на пошту.
- Result: Формується звіт, що відправляється на пошту.

Для налаштування цієї логіки використовувалися такі елементи:

- ручний тригер;
- локальна змінна ToDoCount;
- локальна змінна InProgressCount;
- локальна змінна DoneCount;
- локальна змінна DueCardsLinks (зображена на рисунку 4.10);
- задача отримати список всіх карток дошки;
- цикл з умовами;
- задача відправити згенерований звіт на пошту.

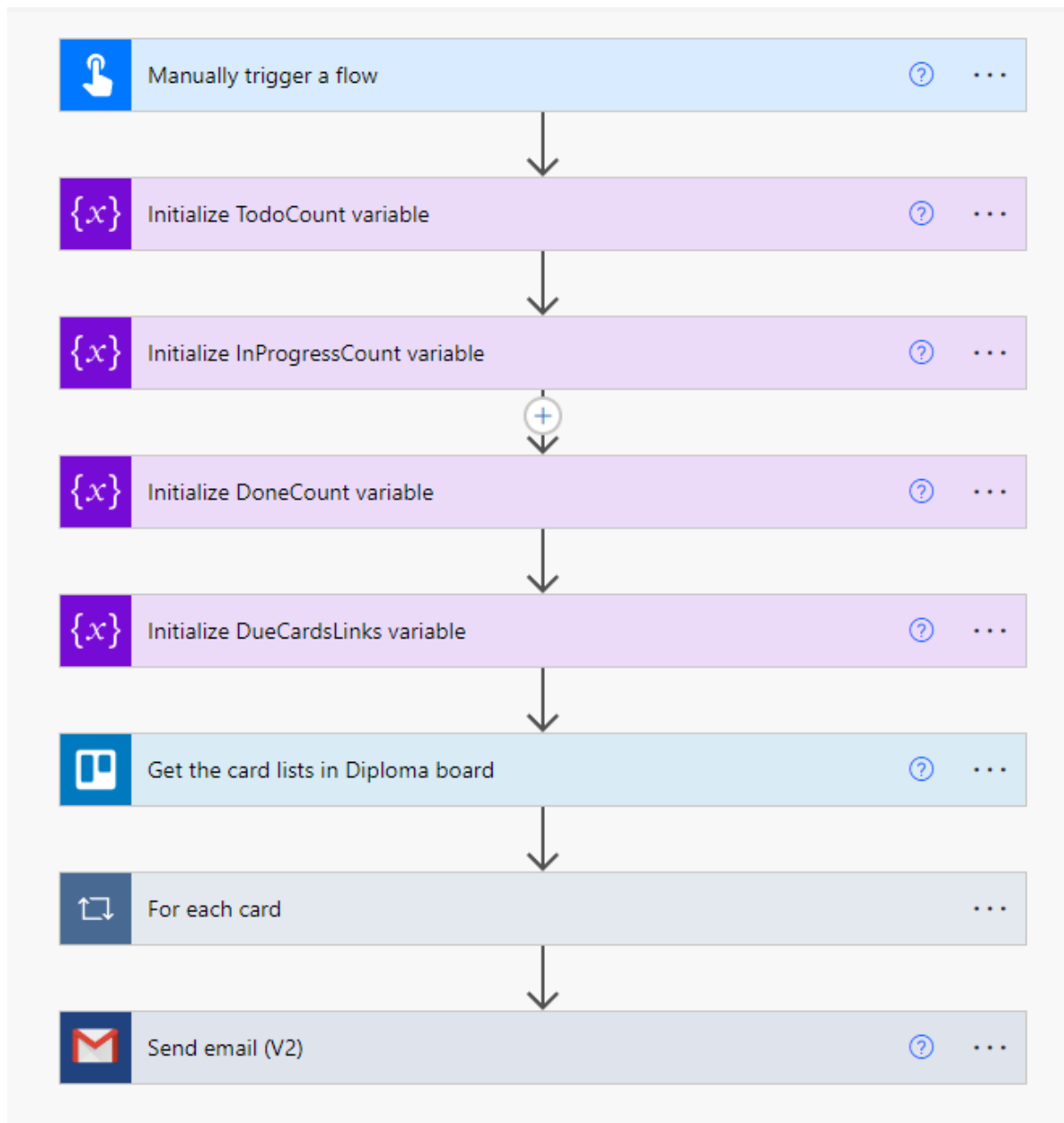


Рис. 4.9. Інтеграція «Send report about board status to email»

*Name		DueCardsLinks
*Type		Array
Value		[]

Рис. 4.10. Оголошення локальної змінної DueCardsLinks

Підготовка даних відбувається в циклу (його налаштування зображено на рисунку 4.11) відбувається з використанням умов для кожного списку карток та працює наступним чином. Для кожного списку перевіряється його ім'я і якщо воно відповідає "To do", "In progress" або "Done", виконуються відповідні наступні кроки. Для кожної картки зі списку "To do" інкрементується змінна TodoCount, яка зберігає загальну кількість карток в цьому списку, а також перевіряється очікувана дата виконання "DueDate". Якщо очікувана дата виконання вже пройшла, то посилання на поточну картку додається до змінної-масиву "DueCardsLinks". Процес роботи циклу на прикладі одного з списків зображено на рисунку 4.12.

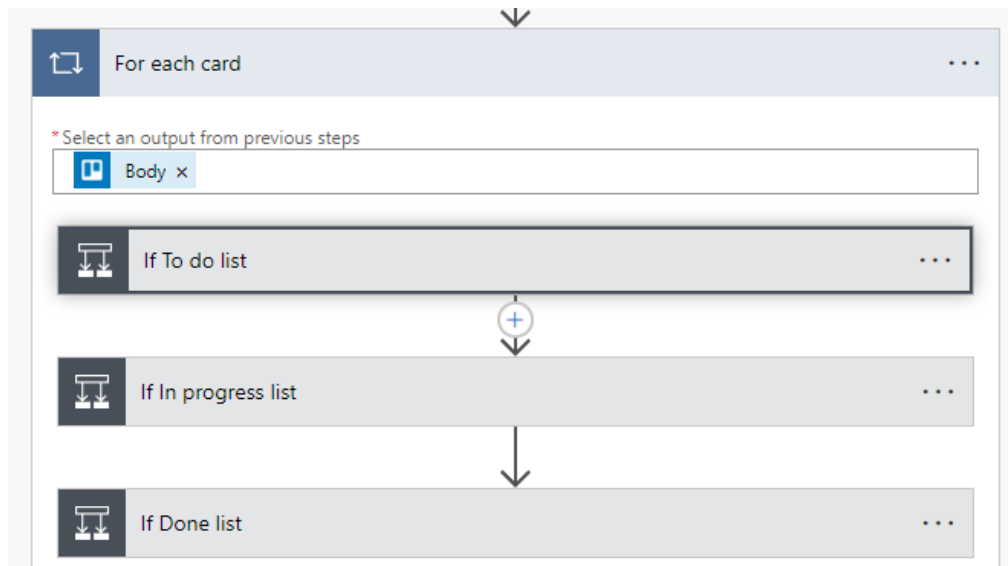


Рис. 4.11. Загальна конфігурація циклу

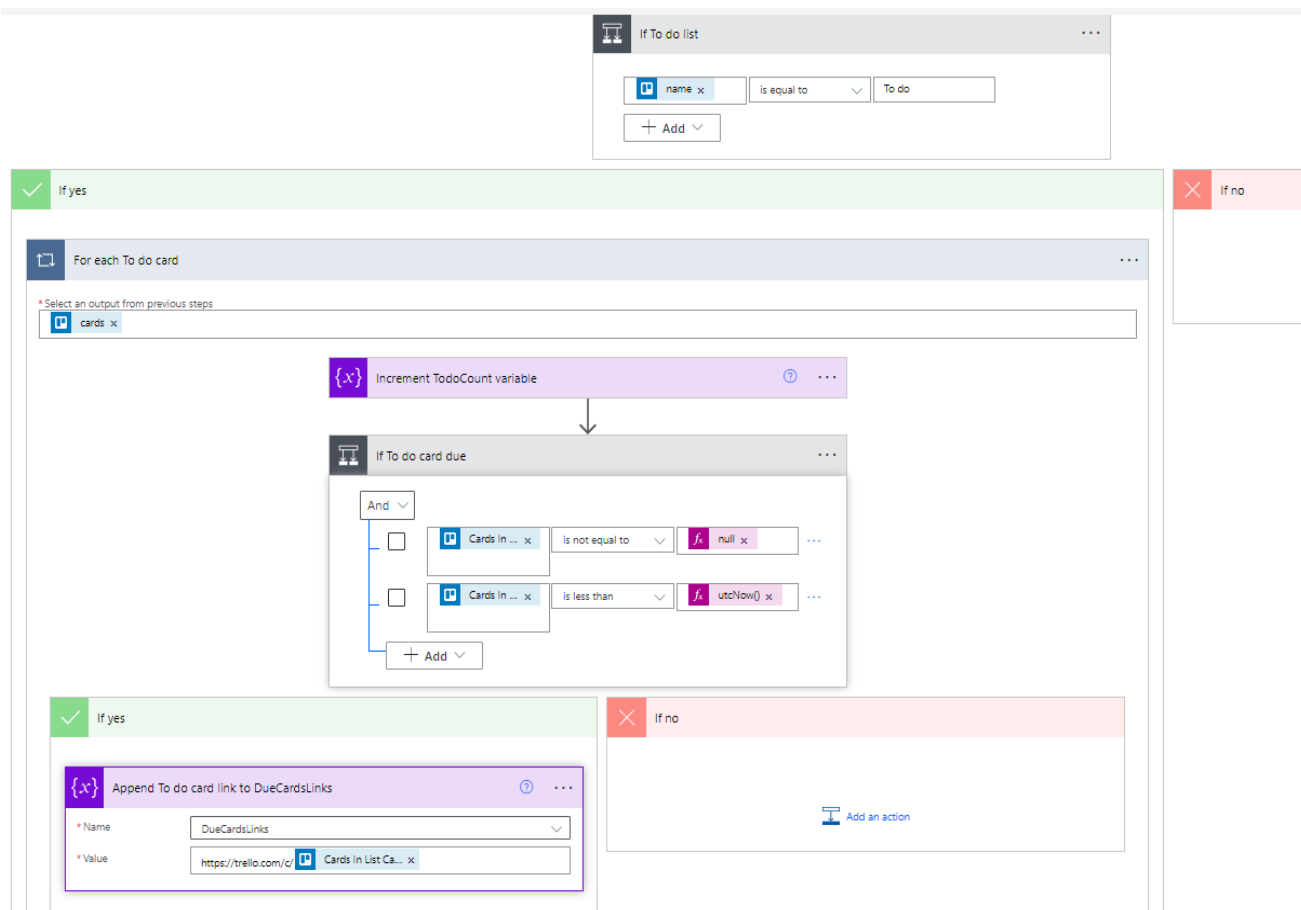


Рис. 4.12. Налаштування умов для обчислень в списку

Після підготування даних наступною дією є відправка листа з деталями звіту на вказану електронну пошту. Лист формується динамічно з використанням змінних, а також додатково обчислюється відсоток виконаних карток та додаються посилання на всі картки, що прострочили очікувану дату виконання. Задача для надсилання листа зображена на рисунку 4.13.

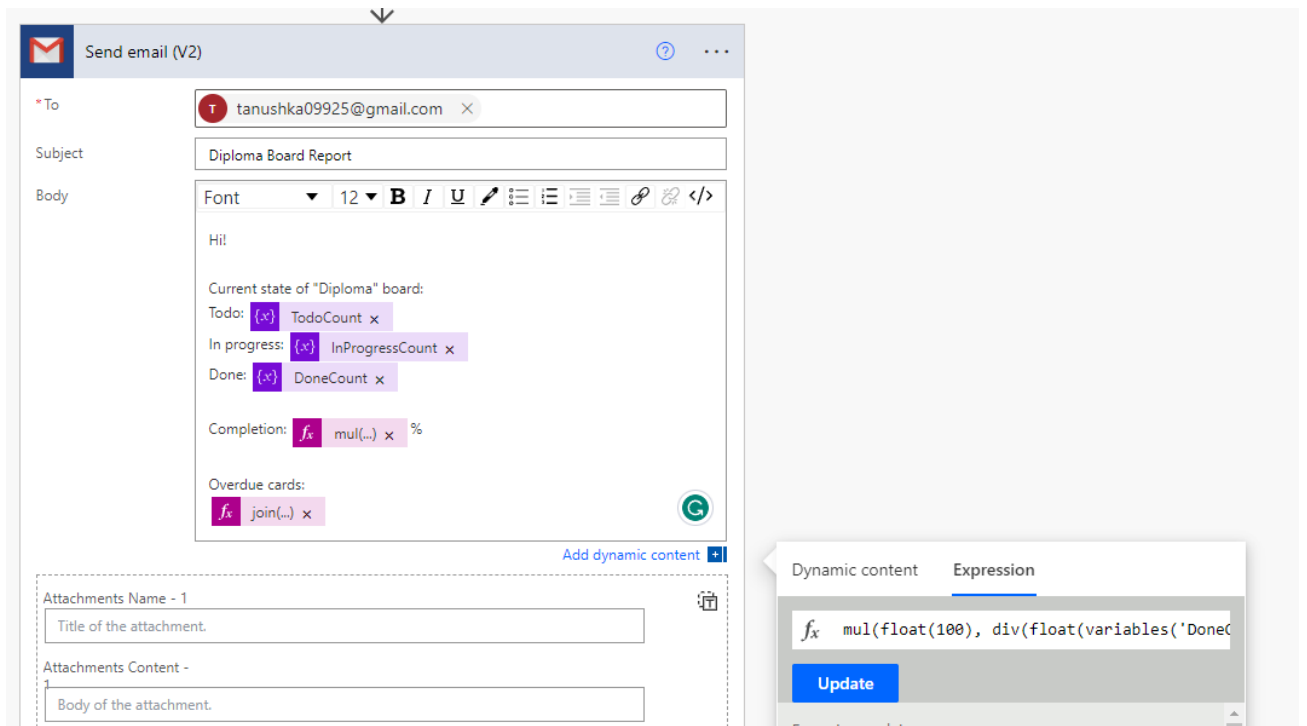


Рис. 4.13. Sent email задача

Відсоток виконаних карток обчислюється наступною функцією `mul(float(100), div(float(variables('DoneCount')), add(float(variables('ToDoCount')), add(float(variables('InProgressCount')), float(variables('DoneCount'))))))`.

Функція для генерування посилань на протермінованих карток – `join(variables('DueCardsLinks'), '
')`.

В результаті на визначену електрону адресу надсилається лист із згенерованим за цими правилами звітом (рисунок 4.14).

Diploma Board Report Вхідні x



tanushka09925@gmail.com

кому мені ▾

Hi!

Current state of "Diploma" board:

Todo: 5

In progress: 2

Done: 1

Completion: 12.5%

Overdue cards:

<https://trello.com/c/ValLblmc>

Рис. 4.14. Приклад електронного листа

Висновки

Power Automate – це популярний та надійний інструмент з використанням функцій Azure Logic App з широким вибором безкоштовних та платних конекторів, які дозволяють легко виконувати інтеграції різноманітних систем. Він є простим у використанні через зручний інтерфейс та low-code принципи. Перевагою є те, що у разі відсутності готових конекторів, є можливість написання користувацьких з'єднувачів з використанням API.

Слід враховувати, що ця платформа не є найкращим варіантом для інтеграції з локальними системами та інтеграцій складних та великих обсягів даних. Безпекові протоколи також можуть бути вирішальним фактором у виборі якої-небудь іншої iPaaS платформи, адже в Power Automate складно налаштувати власні правила надійності.

Для прикладу було реалізовано інтеграцію служб Gmail та Trello. Power Automate показав себе як підходящий варіант для визначених вимог та задач такої інтеграції. До того ж через велику варіативність навчальних ресурсів інструмент було просто використовувати для ознайомлення з роботою інтеграційних платформ

в цілому. Для демонстрації процесу створення інтеграції було детально розписано елементи робочих процесів та їхню конфігурацію.

Power Automate надає можливість створення розгалуженого виконання робочих задач, що було продемонстровано на прикладі розробленої інтеграції.

Запуск інтеграцій може бути налаштований за різних умов – ручний запуск, за розкладом та за виконання передумов. Це надає гнучкості конфігурацій.

ВИСНОВОК

В сучасному світі проблема інтеграції даних займає одну з провідних ролей. Це пов'язано із збільшенням обсягів даних, якими оперують системи через розвиток галузі в цілому. Тому для вирішення задач з інтеграції даних існує багато підходів, основні з яких було розглянуто в першому розділі кваліфікаційної роботи, – REST, SOAP, GraphQL, WebSocket, ETL, ESB. Всі ці технології та підходи мають низку переваг і недоліків та цілком можуть бути використанні для певних задач інтеграцій даних. Проте всі вони потребують висококваліфікованого персоналу, який міг би запровадити інтеграцію.

Дедалі частіше для спрощення та автоматизації ресурсозатратного процесу інтеграції використовують відносно новий альтернативний підхід, що базується на використанні рішень інтеграційних платформ як послуг (iPaaS).

Під час виконання кваліфікаційної роботи було детально розглянуто можливості, що надає використання iPaaS платформ для автоматизації інтеграційних процесів. Незважаючи на те, що iPaaS платформи – це хмарні рішення, багато провайдерів надають можливість інтегрувати локальні, хмарні та гібридні системи, це значно підвищує популярність використання таких платформ. Через збільшення попиту зростає і кількість провайдерів iPaaS, які надають широкий спектр функцій. Це дещо ускладнює процес вибору постачальника. У дипломному проекті було порівняно декілька популярних iPaaS платформ, таких як Paragon, Microsoft Azure Logic Apps, Prismatic, SAP CPI, Alumio та Tray.io. Однією з переваг використання iPaaS є те, що вони є low-code або no-code рішеннями, це спрощує процес інтеграцій, а також гарантує, що ознайомившись з процесом впровадження автоматизацій одного провайдера, не повинно виникнути складнощів у переході на використання іншого.

Використання iPaaS рішень для інтеграції систем вимагає дотримання певних правил при проектуванні програм. Слід дотримуватися виконання цих рекомендацій для того, щоб зробити інтеграції не тільки простими у впровадженні, а також легко масштабованими та адаптивними під вимоги бізнесу в майбутньому.

Після проведення досліджень про особливості дизайну систем з використанням iPaaS платформ було продемонстровано інтеграцію кількох служб з використанням рішення від Microsoft – Power Automate. Цей засіб в своїй основі використовує Azure Logic Apps з додатковим функціоналом та покращеним користувацьким досвідом. Для прикладу було реалізовано інтеграцію даних між Trello та Gmail, в якій було автоматизовано процес надсилання інформації про стан дошки в Trello на електронну пошту. Розробка даного рішення була значно спрощена, тому що Power Automate має готові конектори для здійснення цих операцій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Data Integration Approaches – Which One is Right for Business? [Електронний ресурс]. – 2020. – Режим доступу : <https://dataintegrationinfo.com/data-integration-approaches/> (дата звернення 20.09.2023). – Назва з екрану.
2. Інтеграційні платформи (iPaaS): у чому фішка [Електронний ресурс]. – 2019. – Режим доступу : <https://dou.ua/lenta/articles/integration-platforms-unboxing/> (дата звернення 16.09.2023). – Назва з екрану.
3. What is an enterprise service bus (ESB)? [Електронний ресурс]. – Режим доступу : <https://www.ibm.com/topics/esb> (дата звернення 20.09.2023). – Назва з екрану.
4. ETL и ELT: ключевые различия, о которых должен знать каждый [Електронний ресурс]. – 2022. – Режим доступу : <https://habr.com/ru/articles/695546/> (дата звернення 19.09.2023). – Назва з екрану.
5. Advanced Message Queueing Protocol (AMQP) 1.0 support in Service Bus [Електронний ресурс]. – 2023. – Режим доступу : <https://learn.microsoft.com/en-us/azure/service-bus-messaging/service-bus-amqp-overview> (дата звернення 19.09.2023). – Назва з екрану.
6. Middleware V/S Point-to-point [Електронний ресурс]. – Режим доступу : <https://integrtr.com/middleware-vs-point-to-point/> (дата звернення 20.09.2023). – Назва з екрану.
7. What Is Integration Platform as a Service (iPaaS)? Definition, Components, Providers, and Trends [Електронний ресурс]. – 2022. – Режим доступу : <https://www.spiceworks.com/tech/cloud/articles/what-is-ipaas/> (дата звернення 19.09.2023). – Назва з екрану.
8. What is iPaaS? [Електронний ресурс]. – Режим доступу : <https://www.techtarget.com/searchcloudcomputing/definition/iPaaS-integration-platform-as-a-service> (дата звернення 20.09.2023). – Назва з екрану.
9. 10 прогнозів хмарної індустрії на 2020-2024 [Електронний ресурс]. – Режим доступу : <https://denovo.ua/blog/10-prognoziv-hmarnoyi-industriyi-na-2020-2024->

- 69 (дата звернення 25.11.2023). – Назва з екрану.
10. iPaaS Solution: Definition, Features, and Impact [Електронний ресурс]. – Режим доступу : <https://medium.com/@adeptia/ipaas-solution-definition-features-and-impact-533055c93b91> (дата звернення 11.11.2023). – Назва з екрану.
 11. What is iPaaS? [Електронний ресурс]. – Режим доступу : <https://www.tibco.com/reference-center/what-is-ipaas> (дата звернення 05.12.2023). – Назва з екрану.
 12. Key challenges with iPaaS and how to resolve them [Електронний ресурс]. – Режим доступу : <https://www.cio.com/article/303605/key-challenges-with-ipaas-and-how-to-resolve-them.html> (дата звернення 28.10.2023). – Назва з екрану.
 13. Top 10 Use Cases for Integration-Platform as-a-Service (iPaaS) [Електронний ресурс]. – Режим доступу : <https://www.actian.com/wp-content/uploads/2020/01/Actian-IpaaS-Usecases-eBook.pdf> (дата звернення 02.12.2023). – Назва з екрану.
 14. Explore a detailed comparison of top iPaaS vendors SAP CPI vs. MS Azure vs. Alumio iPaaS [Електронний ресурс]. – Режим доступу : <https://www.alumio.com/compare/sap-cpi-vs-ms-azure-vs-alumio-ipaas> (дата звернення 27.11.2023). – Назва з екрану.
 15. What is Power Automate and Who is It For? [Електронний ресурс]. – Режим доступу : <https://www.vectra.ai/learning/power-automate> (дата звернення 21.11.2023). – Назва з екрану.