ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Архітектура хмарної системи для обміну даними електронної комерції з державними службами»

на здобуття освітнього ступеня магістр
за спеціальності 126 Інформаційні системи та технології
(код,найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Роман ВОРОБЙОВ
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав:
здобувач вищої освіти
група ІСДМ-62 Роман ВОРОБЙОВ
(ім'я, ПРІЗВИЩЕ)
Керівник Віктор САГАЙДАК
PhD. *(ім'я, ПРІЗВИЩЕ)*
Рецензент:
(ім'я, ПРІЗВИЩЕ)

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти магістр
Спеціальність 126 Інформаційні системи та технології
Освітньо-професійна прорама Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ
_____ Каміла СТОРЧАК

“____” _____2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Воробйову Роману Руслановичу
(прізвище, ім’я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Архітектура хмарної системи для обміну даними електронної комерції з державними службами
керівник кваліфікаційної роботи: Віктор САГАЙДАК PhD
(ім’я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від “30 ” жовтня 2025 р. № 467

2. Строк подання кваліфікаційної роботи «26» грудня 2025 р.

3. Вихідні дані кваліфікаційної роботи:
1. Технології хмарних обчислень та e-commerce.
2. Архітектура хмарних рішень.
3. Методи інтеграції з державними системами.
4. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1. Аналіз предметної області та вимог.
2. Проектування ахрітектури хмарної системи .
3. Реалізація та аналіз результатів.

5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «30» жовтня 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури		
2.	Аналіз предметної області та вимог		
3.	Проектування ахрітектури хмарної системи		
4.	Реалізація та аналіз результатів		
5.	Висновки по роботі		
6.	Розробка демонстраційних матеріалів, доповідь.		
7.	Оформлення магістерської роботи		

Здобувач вищої освіти _____ Роман ВОРОБЙОВ
(підпис) *(ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Віктор САГАЙДАК
(підпис) *(ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступня магістр: 95 стор., 4 рис., 16 джерел.

Мета роботи – дослідження та практична реалізація інтеграції e-commerce платформи з державною системою еАкциз на базі хмарної інфраструктури AWS з використанням технологій машинного навчання для оптимізації бізнес-процесів.

Об'єкт дослідження – процес автоматизації податкової звітності та управління e-commerce системами в хмарному середовищі.

Предмет дослідження – методи побудови високодоступних хмарних архітектур, технології інтеграції з державними системами та застосування машинного навчання в e-commerce.

Короткий зміст роботи. У першому розділі магістерської роботи виконано аналіз сучасного стану e-commerce ринку в Україні та важливості автоматизації податкової звітності для підприємств, що торгують підакцизними товарами. Проаналізовано архітектурні підходи до побудови хмарних систем та методи забезпечення високої доступності.

У другому розділі виконано огляд технологій інтеграції з державною системою еАкциз. Проаналізовано підходи до роботи з кваліфікованим електронним підписом, SOAP протоколом та асинхронною обробкою запитів. Досліджено методи забезпечення безпеки та відповідності нормативним вимогам.

У третьому розділі описується практична реалізація системи на базі AWS інфраструктури з використанням managed services. Представлено результати впровадження системи в production середовище та проаналізовано досягнуті технічні та бізнес-результати.

КЛЮЧОВІ СЛОВА**:** ХМАРНІ ОБЧИСЛЕННЯ, E-COMMERCE, AWS, ІНТЕГРАЦІЯ З ЕАКЦИЗ, АВТОМАТИЗАЦІЯ ЗВІТНОСТІ, МІКРОСЕРВІСНА АРХІТЕКТУРА, DEVOPS, CI/CD, INFRASTRUCTURE AS CODE, МАШИННЕ НАВЧАННЯ, КОНТЕЙНЕРИЗАЦІЯ, ВИСОКА ДОСТУПНІСТЬ.

ABSTRACT

The text part of the qualifying work for obtaining a master's degree: 95 pp., 4 fig., 16 sources.

The purpose of the work is to research and implement the integration of an e-commerce platform with the state e-Excise system based on AWS cloud infrastructure using machine learning technologies to optimize business processes.

The object of research is the process of automating tax reporting and managing e-commerce systems in a cloud environment.

The subject of the study is methods for building highly available cloud architectures, technologies for integration with state systems, and the application of machine learning in e-commerce.

Summary of the work. The first chapter of the master's thesis analyzes the current state of the e-commerce market in Ukraine and the importance of automating tax reporting for companies that trade in excisable goods. Architectural approaches to building cloud systems and methods for ensuring high availability are analyzed.

The second chapter provides an overview of technologies for integration with the state eAkcyz system. Approaches to working with qualified electronic signatures, the SOAP protocol, and asynchronous request processing are analyzed. Methods for ensuring security and regulatory compliance are investigated.

The third chapter describes the practical implementation of the system based on AWS infrastructure using managed services. The results of the system's implementation in a production environment are presented, and the technical and business results achieved are analyzed.

KEYWORDS: CLOUD COMPUTING, E-COMMERCE, AWS, INTEGRATION WITH EAKTSIZ, REPORTING AUTOMATION, MICROSERVICE ARCHITECTURE, DEVOPS, CI/CD, INFRASTRUCTURE AS CODE, MACHINE LEARNING, CONTAINERIZATION, HIGH AVAILABILITY.

ЗМІСТ

ВСТУП

Актуальність теми. Електронна комерція в Україні переживає період стрімкого зростання, що супроводжується значним ускладненням бізнес-процесів та підвищенням вимог до автоматизації. Підприємства, що торгують підакцизними товарами, стикаються з необхідністю ведення складної податкової звітності через державну систему еАкциз, що потребує значних трудових ресурсів та створює ризики помилок при ручній обробці документів.

Традиційні on-premise рішення для e-commerce платформ мають обмежену масштабованість, високі витрати на підтримку інфраструктури та складність забезпечення високої доступності. Водночас хмарні технології, зокрема платформа Amazon Web Services (AWS), надають можливості для побудови гнучких, масштабованих та економічно ефективних систем, які здатні автоматично адаптуватися до змінного навантаження та забезпечувати високу доступність сервісів.

Інтеграція з державними системами, такими як еАкциз, вимагає застосування спеціалізованих технологій роботи з кваліфікованим електронним підписом, дотримання форматів XML документів та забезпечення повної трейсабельності операцій. Ручний процес формування акцизних марок займає десятки хвилин на одне замовлення, що при значних обсягах продажу призводить до суттєвих витрат робочого часу та підвищує ймовірність помилок у документації.

Мета роботи – дослідження та практична реалізація інтеграції e-commerce платформи з державною системою еАкциз на базі хмарної інфраструктури AWS з використанням технологій машинного навчання для оптимізації бізнес-процесів.

Для досягнення мети, у магістерській роботі успішно виконано наступні завдання:

- Дослідження сучасного стану e-commerce ринку в Україні та аналіз вимог до автоматизації податкової звітності;
- Проектування архітектури хмарної системи на базі AWS з використанням мікросервісного підходу;

- Розробка методів інтеграції з державною системою еАкциз з використанням асинхронної обробки та криптографічних операцій;
- Впровадження системи в production середовище та аналіз результатів експлуатації.

Об'єкт дослідження – процес автоматизації податкової звітності та управління e-commerce системами в хмарному середовищі.

Предмет дослідження – методи побудови високодоступних хмарних архітектур, технології інтеграції з державними системами та застосування машинного навчання в e-commerce.

Методи дослідження. Під час написання магістерської кваліфікаційної роботи були використані методи системного аналізу для дослідження e-commerce процесів, методи проектування розподілених систем, практики Infrastructure as Code для автоматизації розгортання інфраструктури, методи забезпечення високої доступності через Multi-AZ deployment, а також методи моніторингу та аналізу продуктивності систем.

Наукова новизна одержаних результатів. У ході дослідження розроблено комплексний підхід до побудови e-commerce системи на базі хмарної платформи AWS з інтеграцією державної системи еАкциз. Запропоновано архітектурне рішення, що забезпечує високу доступність понад 99.9% через Multi-AZ deployment, автоматичне масштабування під змінне навантаження та асинхронну обробку запитів до зовнішніх систем. Розроблено методику інтеграції з legacy державними системами з використанням черг повідомлень та retry механізмів для забезпечення надійності.

Практична значущість одержаних результатів. Запропонована система забезпечує ефективне рішення для автоматизації e-commerce процесів та податкової звітності. Впровадження системи продемонструвало значну економічну ефективність через зниження витрат на інфраструктуру понад 70%, автоматизацію формування акцизних марок з десятків хвилин до кількох секунд та повне усунення штрафів за несвоєчасну податкову звітність. Система може бути адаптована для інших e-commerce компаній в Україні, що потребують інтеграції з державними

системами.

Апробація результатів магістерської роботи. Основні положення і результати магістерської роботи впроваджено в production середовище та здійснювались у формі участі в XVIII міжнародної науково-практичної конференції «Інформаційні технологіх і автоматизація», а також в III Всеукраїнську науково-технічну конференці «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу».

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ

1.1 Огляд систем електронної комерції та взаємодії з державними службами

Електронна комерція (e-commerce) в Україні демонструє стійке зростання протягом останніх років. За даними 2024-2025 років, обсяг онлайн-торгівлі в Україні перевищує 150 млрд грн на рік, що становить близько 7-8% від загального обсягу роздрібної торгівлі. Це зростання обумовлено кількома ключовими факторами: активною цифровізацією традиційного бізнесу, стрімким розвитком безконтактних платіжних систем (Apple Pay, Google Pay, Google Wallet), розширенням логістичної інфраструктури завдяки зростанню мережі служб доставки (Нова Пошта, Укрпошта, Meest Express), а також кардинальною зміною споживчої поведінки, особливо після пандемії COVID-19 та в умовах воєнного стану.

Сучасний ринок e-commerce в Україні характеризується високою конкуренцією та активним впровадженням інновацій. Провідні гравці ринку, такі як Rozetka, Prom.ua, Comfy, Moyo, постійно розширюють асортимент та покращують користувацький досвід. Особливу увагу приділяється розвитку мобільної комерції (m-commerce), адже за статистикою понад 60% онлайн-покупок здійснюється через мобільні пристрої. Платформи впроваджують персоналізацію пропозицій на основі машинного навчання, розширюють способи оплати та доставки, інтегрують програми лояльності та впроваджують омніканальний підхід до продажів.

Класифікація систем електронної комерції

Сучасні платформи електронної комерції можна класифікувати за різними критеріями. За моделлю взаємодії розрізняють B2C (Business-to-Consumer) – коли компанії продають товари безпосередньо кінцевим споживачам. Яскравими прикладами є Rozetka, Prom.ua, Comfy, Moyo, які забезпечують масові продажі та обробляють велику кількість транзакцій щоденно. Модель B2B (Business-to-

Business) передбачає торгівлю між компаніями, зазвичай великими партіями з оптовими цінами та спеціальними контрактними умовами. Такі платформи часто інтегруються з ERP-системами постачальників для автоматизації замовлень та управління запасами.

C2C (Consumer-to-Consumer) модель реалізована через маркетплейси типу OLX та Shafa, де приватні особи продають товари один одному. Платформа виступає посередником, забезпечуючи безпеку угод, систему рейтингів та комунікацію між продавцем і покупцем. B2G (Business-to-Government) модель стосується продажу товарів та послуг державним установам, найяскравішим прикладом чого є система ProZorro для державних закупівель, яка забезпечує прозорість та конкурентність тендерів.

За типом товарів та послуг розрізняють платформи для продажу фізичних товарів (одяг, електроніка, продукти харчування, меблі, побутова техніка), цифрових продуктів (програмне забезпечення, музика, відео, електронні книги, онлайн-курси) та послуг (бронювання готелів та квитків, доставка їжі через Glovo або Bolt Food, таксі через Uber або Uklon, фрілансерські послуги).

Державна податкова служба України (ДПС)

Взаємодія з Державною податковою службою є критично важливою для всіх e-commerce платформ. Кожен суб'єкт господарювання, що здійснює роздрібну торгівлю, зобов'язаний зареєструвати реалізатори розрахункових операцій (РРО) або використовувати програмні РРО. Після кожної продажі система повинна автоматично генерувати фіскальний чек та передавати його до ДПС через оператора фіскальних даних – ТОВ «ІНФОТЕКС». Цей процес повинен відбуватися в режимі реального часу або з мінімальною затримкою [11].

Електронний кабінет платника податків надає можливість автоматичної передачі даних про транзакції, формування та подання електронних декларацій з ПДВ та податку на прибуток, проведення електронних актів звірки з податковими органами, а також перевірки контрагентів на предмет їх благонадійності. Для великих e-commerce платформ критично важливою є можливість інтеграції через API для автоматизації подання звітності та мінімізації ручної роботи бухгалтерів.

Система еАкциз

Система еАкциз є централізованою електронною системою обліку та контролю обігу підакцизних товарів на території України. Вона є обов'язковою для компаній, що здійснюють виробництво, імпорт, оптовий або роздрібний продаж алкогольних напоїв (горілка, лікеро-горілчані вироби, вино, шампанське, коньяк, пиво), тютюнових виробів (цигарки, сигари, тютюн для куріння) та пального (бензин, дизельне пальне, скраплений газ).

Кожна одиниця підакцизного товару повинна бути промаркована акцизною маркою, яка містить унікальний штрих-код, QR-код з криптографічним захистом, серію та номер марки, код виду товару та код виробника. При надходженні товару на склад підприємство зобов'язане зареєструвати факт отримання марок у системі еАкциз, вказавши кількість, дату, постачальника та місце зберігання. При продажі кожна марка повинна бути списана в системі шляхом формування електронного акту списання марок акцизного збору.

Інтеграція з системою еАкциз здійснюється через SOAP API, що вимагає використання кваліфікованого електронного підпису (КЕП) для автентифікації та підпису всіх електронних документів. Для e-commerce платформ це означає необхідність автоматизації процесу списання марок при кожному онлайн-замовленні, що містить підакцизні товари. Система повинна перевіряти наявність марок перед оформленням замовлення, резервувати їх на час обробки та автоматично списувати після підтвердження оплати. При поверненні товару необхідно реєструвати повернення марки в систему.

Державна митна служба

Для e-commerce компаній, що здійснюють міжнародну торгівлю, критично важливою є інтеграція з Державною митною службою України. Система електронного митного оформлення дозволяє подавати митні декларації в електронному вигляді, автоматично розраховувати митні платежі, мита та збори, відстежувати статус митного оформлення вантажів, отримувати електронні дозволи на випуск товарів.

Інтеграція з системою «Єдине вікно подання електронних документів»

спрощує процес оформлення, дозволяючи подавати всі необхідні документи (інвойси, сертифікати походження, фітосанітарні сертифікати, дозволи) через єдиний інтерфейс. Для великих e-commerce платформ, що працюють з міжнародними постачальниками, автоматизація митного оформлення дозволяє суттєво прискорити процес доставки товарів та зменшити операційні витрати.

Державна служба з безпеки харчових продуктів та Держпродспоживслужба

E-commerce платформи, що торгують продуктами харчування, зобов'язані реєструватися як оператори ринку в Державній службі з безпеки харчових продуктів. Вони повинні забезпечувати дотримання температурних режимів при зберіганні та транспортуванні, контролювати терміни придатності, мати сертифікати на всю продукцію. Держпродспоживслужба здійснює контроль якості товарів, перевіряє відповідність характеристик заявленим, розглядає скарги споживачів та може накладати штрафи за порушення прав споживачів.

Технічні виклики інтеграції

Одним з найбільших технічних викликів є використання застарілих протоколів та стандартів у державних системах. Багато API працюють через SOAP замість сучасного REST, використовують застарілі версії SSL/TLS протоколів, не мають належної документації у форматі OpenAPI або Swagger. Це ускладнює розробку та підтримку інтеграційних модулів, вимагає залучення спеціалістів зі специфічними знаннями.

Низька доступність державних систем створює додаткові проблеми. Планові та позапланові технічні роботи часто відбуваються без попередження, система може бути недоступною під час пікових навантажень (кінець місяця, кінець кварталу), відсутність Service Level Agreement (SLA) не дає гарантій щодо часу відновлення сервісу. Це вимагає від e-commerce платформ впровадження механізмів черг для відкладених операцій, retry логіки з експоненційним backoff, моніторингу доступності та автоматичних алертів.

Складність автентифікації пов'язана з обов'язковим використанням кваліфікованого електронного підпису (КЕП). Необхідно працювати з різними стандартами ЕЦП (ДСТУ 4145-2002), використовувати апаратні криптомодулі або

програмні бібліотеки криптопровайдерів (ІІТ, Алмаз-1К), забезпечувати безпечне зберігання сертифікатів та приватних ключів, контролювати терміни дії сертифікатів та своєчасно їх оновлювати.

Відсутність sandbox або тестових середовищ у більшості державних систем обмежує можливості тестування перед виходом у production. Розробники змушені тестувати інтеграцію на реальних даних, що створює ризики помилок, які можуть призвести до фінансових втрат або штрафних санкцій. Деякі системи надають тестові середовища з обмеженою функціональністю, які не повністю відтворюють поведінку production системи.

Регуляторні виклики

Часті зміни податкового законодавства вимагають оперативного оновлення бізнес-логіки системи. Зміни можуть стосуватися ставок ПДВ, правил оподаткування цифрових товарів, вимог до фіскалізації, звітності. Невідповідність новим вимогам може призвести до штрафних санкцій від податкових органів, блокування рахунків, кримінальної відповідальності керівництва.

Складність нормативної бази полягає в тому, що правила встановлюються не тільки законами, але й численними постановами Кабінету Міністрів, наказами міністерств, методичними рекомендаціями податкової служби. Різні органи можуть по-різному трактувати одні й ті ж норми, що створює правову невизначеність. E-commerce компанії змушені постійно моніторити зміни законодавства, консультуватися з юристами та податковими консультантами.

Вимоги до захисту персональних даних регулюються Законом України «Про захист персональних даних». E-commerce платформи повинні отримувати явну згоду користувачів на обробку їх персональних даних, забезпечувати право на доступ, виправлення та видалення даних, обмежувати строк зберігання персональних даних, реєструвати базу персональних даних в Уповноваженого Верховної Ради України. Для компаній, що працюють з громадянами ЄС, додатково застосовуються вимоги GDPR (General Data Protection Regulation), які є ще більш суворими та передбачають значні штрафи за порушення.

Операційні виклики

Синхронізація даних між e-commerce платформою та державними системами повинна відбуватися в режимі реального часу або з мінімальною затримкою. Необхідно забезпечити узгодження складських залишків між внутрішньою системою обліку та системою еАкциз, синхронізацію статусів замовлень, обробку помилок при невдалих спробах передачі даних. Для цього використовуються черги повідомлень (message queues), retry механізми з експоненційним backoff, компенсуючі транзакції для відкату змін у разі помилок.

Масштабування системи під пікові навантаження є критично важливим для e-commerce. Під час розпродажів (Чорна п'ятниця, Кіберпонеділок), святкових сезонів (Новий рік, 8 березня) кількість транзакцій може зростати в десятки разів. Система повинна автоматично масштабуватися, додаючи додаткові обчислювальні ресурси, підтримувати горизонтальне масштабування (додавання нових серверів), використовувати кешування для зменшення навантаження на бази даних та зовнішні API.

Моніторинг та аудит всіх транзакцій необхідні для відповідності регуляторним вимогам. Система повинна логувати всі звернення до зовнішніх API (запити та відповіді), зберігати історію всіх змін критичних даних, забезпечувати можливість швидкого пошуку транзакцій за різними критеріями. Дані повинні зберігатися протягом 3-5 років відповідно до вимог Податкового кодексу України, що вимагає значних обсягів дискового простору та ефективних механізмів архівування.

Переваги автоматизації інтеграції

Автоматизована інтеграція e-commerce платформ з державними службами приносить суттєві переваги. Зниження операційних витрат досягається через мінімізацію ручної роботи бухгалтерів та операторів, які раніше повинні були вручну заповнювати форми, подавати документи. Автоматизація дозволяє одному спеціалісту контролювати процеси, які раніше вимагали роботи кількох осіб.

Зменшення помилок відбувається завдяки автоматичній валідації даних перед відправкою до державних систем. Система може перевіряти правильність заповнення обов'язкових полів, відповідність форматів, коректність розрахунків

податків. Це знижує ризик отримання відмов від державних систем та необхідності повторного подання документів.

Прискорення процесів досягається через миттєву передачу даних електронними каналами замість паперової документації, яка може йти днями. Автоматичне формування звітів та декларацій економить години роботи, швидке отримання підтверджень та статусів операцій дозволяє оперативно реагувати на проблеми.

Автоматичне дотримання законодавчих вимог (комплаєнс) забезпечується через вбудовані перевірки відповідності регуляторним нормам, автоматичне оновлення правил при зміні законодавства, попередження про наближення дедлайнів подання звітності. Повний аудит-лог всіх операцій забезпечує прозорість для внутрішніх та зовнішніх аудиторів, можливість швидко надати дані для податкових перевірок.

Масштабованість дозволяє обробляти великі обсяги транзакцій без пропорційного зростання штату співробітників, автоматично адаптуватися до зростання бізнесу, легко додавати нові інтеграції з іншими державними системами.

1.2 Аналіз системи еакциз та вимог до інтеграції

Система еАкциз є централізованою електронною системою Державної податкової служби України для автоматизованого обліку, контролю виробництва та обігу підакцизних товарів. Впровадження обумовлено необхідністю посилення контролю за обігом підакцизних товарів, боротьби з нелегальним виробництвом та контрабандою, забезпечення повноти сплати акцизного податку. Нормативна база включає Податковий кодекс України (статті 212-229), Постанову КМУ від 11.07.2020 № 597 про порядок маркування, накази Мінфіну та ДПС щодо технічних специфікацій API.

Підакцизні товари під контролем

Система контролює алкогольні напої (горілка понад 8.5%, вино, шампанське, коньяк, пиво понад 0.5%), тютюнові вироби (цигарки, сигари, тютюн для куріння),

пальне (бензин, дизель, скраплений газ, авіаційне паливо). Кожна одиниця маркується акцизною маркою з унікальним номером (AA1234567890), штрих-кодом Code-128/Data Matrix, QR-кодом з криптографічним захистом, голографічними елементами.

Життєвий цикл марки

Замовлення марок через електронний кабінет системи з вказанням кількості, виду товару, реквізитів підприємства. Після схвалення та оплати марки виготовляються та передаються. Отримання супроводжується електронним актом приймання-передачі, підписаним ЕЦП обох сторін. Марки переходять у статус "На складі". Маркування товару передбачає фізичне нанесення марки на упаковку з реєстрацією факту в системі. Оприбуткування включає внесення інформації про складські залишки з розбивкою по номерах марок та місцю зберігання [12].

Реалізація формує електронний акт списання марок з вказанням номера, дати, місця продажу. Марка переходить у статус "Списана". При поверненні товару формується документ про повернення, марка отримує статус "Повернута" для повторного продажу. При боє або псуванні формується акт списання з причиною, марка отримує статус "Знищена".

Технічна архітектура API

Система надає SOAP веб-сервіси. Формат даних XML з строгою типізацією та XSD валідацією. Транспорт HTTPS з TLS 1.2+, порт 443.

Автентифікація через КЕП

Обов'язкове використання кваліфікованого електронного підпису відповідно до ДСТУ 4145-2002. КЕП створюється акредитованими АЦСК, зберігається на захищених носіях (HSM або апаратні токени). SOAP запит містить Header з блоком Authentication, що включає сертифікат (Base64) та цифровий підпис (XMLDSig). Body містить бізнес-дані запиту.

Криптопровайдери включають ІІТ (JavaScript, Java, .NET бібліотеки), Алмаз-1К (апаратні токени), Криптон. Для e-commerce необхідна інтеграція однієї з цих бібліотек для автоматичного підпису запитів.

Основні API методи

Робота з марками: GetStampBalance (отримання залишків з фільтрацією), RegisterStampReceipt (реєстрація отримання від постачальника), WriteOffStamps (списання при продажу - найчастіший в e-commerce), ReturnStamps (повернення при поверненні товару). Звітність: SubmitMonthlyReport (місячний звіт про обіг), GetReportStatus (статус звіту), GetReportErrors (помилки валідації). Довідники: GetProductTypes (типи підакцизних товарів), GetStampInfo (інформація про марку), ValidateStamp (перевірка дійсності).

Функціональні вимоги до інтеграції

Автоматична реєстрація продажу при оформленні онлайн-замовлення з викликом WriteOffStamps без ручного втручання, формування електронного акту списання. Синхронізація залишків щоденно в автоматичному режимі (нічний час), перевірка доступності марок перед оформленням замовлення, оновлення інвентарю після кожної операції. Обробка повернень з автоматичною реєстрацією в еАкциз, перевіркою можливості повернення. Автоматизована звітність з формуванням місячних звітів на основі операцій, архівуванням мінімум 3 роки. Валідація марок при прийманні товару, виявлення підроблених або списаних марок.

Нефункціональні вимоги

Надійність: доступність 99.5% (max 3.65 годин downtime/місяць), MTTR < 1 година, автоматичний retry з exponential backoff до 5 спроб. Безпека: КЕП для всіх операцій, TLS 1.2+ з strong ciphers, зберігання сертифікатів у AWS Secrets Manager/KMS, логування всіх операцій. Масштабованість: горизонтальне масштабування при навантаженні, multi-tenant для кількох брендів, обробка піків через черги.

Обробка виключних ситуацій

При недоступності API еАкциз (503, timeout): додавання в чергу відкладених операцій (SQS/RabbitMQ), автоматичний retry, алерти DevOps команді (CloudWatch/PagerDuty), повідомлення користувачам про тимчасову недоступність. При помилках списання (марка списана, не знайдена, expired сертифікат): rollback транзакції замовлення, повернення коштів, детальний error

log, повідомлення менеджерів складу. При невідповідності залишків: щоденна автоматична звірка (нічний час), виявлення розбіжностей понад 5 марок, нотифікація менеджерів про інвентаризацію, блокування продажу до з'ясування.

Специфіка онлайн-продажу підакцизних товарів

Перевірка віку обов'язкова (18+): верифікація при реєстрації через паспорт/ІПН, повторна перевірка при оформленні замовлення, підтвердження кур'єром при доставці. Географічні обмеження: заборона в межах 200м від шкіл/лікарень, заборона продажу 23:00-08:00, обмеження на доставку в певні регіони.

Типовий флоу продажу: додавання товару в кошик → GetStampBalance для перевірки наявності → резервування марки (15-30 хв) → оплата → WriteOffStamps після підтвердження → формування документів з серійним номером → доставка з перевіркою віку → підтвердження в системі.

1.3. Огляд хмарних технологій AWS для побудови інтеграційних рішень

Amazon Web Services становить найбільшу хмарну платформу у світі, що пропонує понад двісті повнофункціональних сервісів для підтримки різноманітних бізнес-потреб підприємств різного масштабу. Компанії використовують AWS для побудови масштабованих, надійних та безпечних систем, включаючи складні інтеграційні рішення, що поєднують різні додатки, бази даних та зовнішні служби в єдину екосистему автоматизованих бізнес-процесів.

Платформа AWS базується на моделях Infrastructure as a Service, Platform as a Service та Software as a Service, що дозволяє підприємствам обирати рівень абстракції відповідно до специфічних потреб конкретних проектів. Ключові переваги платформи включають еластичність ресурсів з можливістю автоматичного масштабування відповідно до поточного навантаження без необхідності передбачення піків активності, глобальну інфраструктуру з більш ніж тридцятьма регіонами та дев'яноста зонами доступності по всьому світу для забезпечення низької мережевої затримки та високої доступності сервісів, модель

оплати за фактичне використання ресурсів без капітальних витрат на придбання обладнання, гарантії доступності на рівні високих показників угоди про рівень обслуговування для критично важливих сервісів та багаторівневу систему безпеки з відповідністю міжнародним стандартам ISO 27001, SOC 2 та PCI DSS, що особливо важливо для систем обробки фінансових транзакцій та персональних даних.

Сервіс Elastic Compute Cloud становить основу обчислювальної потужності платформи AWS, надаючи віртуальні сервери для запуску застосунків у хмарному середовищі. Платформа підтримує широкий спектр типів інстансів, оптимізованих для різних характерів навантаження та вимог до ресурсів. Інстанси загального призначення забезпечують збалансоване співвідношення процесорної потужності, оперативної пам'яті та мережевих ресурсів, що робить їх оптимальним вибором для веб-серверів та застосунків малого і середнього масштабу з типовими вимогами до продуктивності. Інстанси, оптимізовані для обчислень, оснащені високопродуктивними процесорами останніх поколінь для виконання інтенсивних обчислювальних задач, таких як наукові розрахунки, обробка відео та високопродуктивні веб-сервери. Інстанси з оптимізацією для пам'яті надають великі обсяги оперативної пам'яті для баз даних, що зберігають дані в пам'яті, систем реального часу та аналітичних платформ, які потребують швидкого доступу до великих наборів даних. Інстанси, оптимізовані для зберігання, комплектуються високошвидкісним локальним сховищем на твердотільних накопичувачах для систем Big Data, розподілених файлових систем та додатків, що вимагають високої пропускної здатності операцій введення-виведення.

Забезпечення високої доступності застосунків досягається через розміщення інстансів Elastic Compute Cloud у різних зонах доступності в межах одного географічного регіону, що захищає від відмов окремих дата-центрів та забезпечує безперервність бізнес-процесів. Механізм автоматичного масштабування Auto Scaling коригує кількість працюючих інстансів залежно від поточного навантаження згідно з визначеними метриками, такими як використання процесора, кількість запитів або спеціалізовані бізнес-метрики. Під час зростання

навантаження система автоматично запускає додаткові інстанси для обробки підвищеного трафіку, тоді як у періоди низької активності зайві інстанси зупиняються для оптимізації витрат на обчислювальні ресурси [1].

Сервіс AWS Lambda забезпечує виконання програмного коду без необхідності керування серверною інфраструктурою, автоматично масштабуючись у відповідь на події різного характеру. Платформа Lambda реагує на HTTP-запити через інтеграцію з API Gateway, зміни в базах даних через потоки подій DynamoDB Streams, повідомлення в чергах SQS, завантаження файлів до об'єктного сховища S3 та багато інших типів подій у екосистемі AWS. Розробники зосереджуються виключно на бізнес-логіці додатку, тоді як платформа автоматично управляє всією інфраструктурою виконання, включаючи розподіл обчислювальних ресурсів, масштабування під навантаженням, моніторинг працездатності та забезпечення відмовостійкості через автоматичне перезапускання функцій при виникненні помилок [2].

Сервіси Amazon Elastic Container Service забезпечує оркестрацію контейнеризованих додатків у хмарному середовищі з автоматизованим керуванням життєвим циклом контейнерів. Сервіс ECS становить власну розробку AWS для оркестрації контейнерів з глибокою інтеграцією з іншими сервісами платформи, включаючи балансувальники навантаження Application Load Balancer та Network Load Balancer, систему моніторингу CloudWatch, управління доступом через IAM ролі та систему секретів Secrets Manager. Платформа ECS підтримує два режими роботи: запуск контейнерів на попередньо виділених віртуальних машинах EC2 для повного контролю над обчислювальним середовищем або використання безсерверної платформи AWS Fargate, де AWS автоматично керує всією інфраструктурою виконання контейнерів.

Сервіс Simple Storage Service становить об'єктне сховище з надзвичайно високою надійністю збереження даних та практично необмеженою масштабованістю для зберігання та отримання довільного обсягу інформації. Платформа пропонує різні класи зберігання для оптимізації витрат залежно від частоти доступу до даних та вимог до швидкості отримання інформації.

Стандартний клас зберігання S3 Standard призначений для даних з частим доступом та забезпечує мілісекундну затримку отримання об'єктів. Клас інтелектуального рівневого зберігання S3 Intelligent-Tiering автоматично переміщує дані між рівнями з частим та рідким доступом залежно від фактичних патернів використання, оптимізуючи вартість без впливу на продуктивність або доступність. Клас S3 Glacier призначений для довгострокового архівування даних з рідким доступом, пропонуючи значно нижчу вартість зберігання за рахунок збільшеного часу отримання даних від кількох хвилин до кількох годин залежно від обраної опції відновлення.

Механізм версіонування об'єктів забезпечує захист від випадкового видалення або перезапису файлів шляхом збереження всіх історичних версій кожного об'єкта у сховищі. Політики життєвого циклу дозволяють визначити правила автоматичного переміщення даних між класами зберігання залежно від віку об'єкта, наприклад автоматичне переміщення файлів старших тридцяти днів до класу з рідким доступом для економії витрат. Шифрування даних реалізовано як у стані спокою через серверне шифрування з ключами, керованими AWS або клієнтом через сервіс Key Management Service, так і під час передачі через обов'язкове використання захищених протоколів SSL/TLS для всіх операцій завантаження та отримання об'єктів.

Сервіс S3 часто застосовується для зберігання журналів роботи застосунків та інфраструктурних компонентів з можливістю подальшого аналізу через інструменти обробки великих даних, резервних копій баз даних з автоматичним шифруванням та можливістю відновлення до довільного моменту часу, статичних файлів веб-сайтів з можливістю прямого обслуговування через CloudFront CDN для зменшення навантаження на сервери застосунків, медіа-контенту високої роздільності з підтримкою потокового відтворення та даних для аналітики з інтеграцією у системи обробки великих даних через сервіси як Athena, EMR та Redshift Spectrum.

Сервіс Amazon Relational Database Service надає керовані реляційні бази даних з підтримкою популярних двигунів, включаючи MySQL для веб-застосунків

з відкритим кодом, PostgreSQL для застосунків з складними запитами та потребою у розширеній функціональності, MariaDB як сумісну альтернативу MySQL, Oracle Database для корпоративних застосунків з ліцензійною підтримкою та Microsoft SQL Server для застосунків на технологіях Microsoft. Платформа RDS автоматизує рутинні адміністративні задачі, які традиційно вимагали значних затрат часу адміністраторів баз даних, включаючи автоматичне резервне копіювання з можливістю відновлення до будь-якого моменту часу в межах встановленого періоду утримання резервних копій, застосування патчів безпеки та оновлень бази даних під час визначених вікон обслуговування для мінімізації впливу на роботу застосунків, безперервний моніторинг продуктивності з автоматичним сповіщенням про аномалії та вузькі місця, а також вертикальне масштабування обчислювальних ресурсів шляхом зміни класу інстансу бази даних з мінімальним часом недоступності [5].

Amazon Aurora становить власну реляційну базу даних AWS, розроблену з нуля для хмарного середовища та повністю сумісну з MySQL та PostgreSQL на рівні протоколів та SQL синтаксису. Архітектура Aurora забезпечує до п'яти разів вищу продуктивність порівняно зі стандартним MySQL та до трьох разів вищу продуктивність порівняно з PostgreSQL при роботі на ідентичному обладнанні завдяки оптимізованому двигуну зберігання та розподіленій архітектурі. Платформа Aurora автоматично реплікує дані між трьома зонами доступності в межах регіону, забезпечуючи шість копій даних для максимальної надійності та можливості швидкого відновлення після відмови окремих компонентів інфраструктури. Механізм швидкого відновлення після збоїв дозволяє перемкнутися на репліку протягом менше тридцяти секунд без втрати даних у разі відмови первинного інстансу бази даних [6].

Сервіс Amazon DynamoDB становить повністю керовану NoSQL базу даних, що забезпечує стабільну мілісекундну затримку відповіді при будь-якому масштабі навантаження від кількох запитів на секунду до мільйонів операцій читання та запису. Гнучка модель даних без фіксованої схеми дозволяє зберігати документи з різною структурою в одній таблиці, що прискорює розробку застосунків через

відсутність необхідності визначення схеми бази даних на ранніх стадіях проекту та спрощує еволюцію структури даних у процесі розвитку продукту. Автоматичне масштабування DynamoDB обробляє практично необмежену кількість запитів шляхом автоматичного розподілу даних між множиною серверів без необхідності ручного шардування та перебалансування навантаження [4].

Сервіс API Gateway становить повністю керовану платформу для створення, публікації, підтримки, моніторингу та захисту програмних інтерфейсів RESTful та WebSocket на будь-якому масштабі від кількох запитів на день до мільйонів запитів на секунду. Платформа забезпечує маршрутизацію вхідних запитів до різних серверних компонентів, включаючи функції Lambda для безсерверної обробки логіки, інстанси EC2 або контейнери ECS для традиційних застосунків на серверах, HTTP ендпоінти сторонніх сервісів для проксування запитів до зовнішніх API та сервісів AWS, таких як DynamoDB, S3 або Step Functions для прямої інтеграції без необхідності написання коду обробників.

Система авторизації та автентифікації підтримує різні механізми перевірки легітимності запитів, включаючи IAM політики для контролю доступу користувачів та сервісів AWS, інтеграцію з Cognito для керування користувачами застосунків з підтримкою реєстрації, входу та соціальної автентифікації, Lambda Authorizers для реалізації власної логіки авторизації з можливістю інтеграції з будь-якими системами керування ідентичністю та перевірку JWT токенів для stateless автентифікації на основі стандартних токенів безпеки. Механізм обмеження швидкості запитів та квот захищає серверні системи від перевантаження та DDoS атак шляхом обмеження кількості запитів від окремих клієнтів або API ключів протягом визначеного періоду часу.

Функціональність трансформації запитів та відповідей дозволяє змінювати формат даних на льоту без необхідності модифікації коду серверних застосунків, наприклад конвертувати JSON у XML для сумісності з legacy системами або додавати, видаляти чи перейменовувати поля у структурі даних. Вбудований механізм кешування зменшує навантаження на серверні компоненти шляхом зберігання відповідей на типові запити з можливістю інвалідації кешу при зміні

даних або після закінчення визначеного терміну життя записів. Інтеграція з системою моніторингу CloudWatch забезпечує відстеження ключових метрик продуктивності API, включаючи кількість запитів, час відповіді, частоту помилок та розмір переданих даних, а також детальне логування всіх запитів для діагностики проблем та аналізу патернів використання API.

Сервіс Simple Queue Service становить повністю керовану систему черг повідомлень для асинхронного обміну даними між розподіленими компонентами програмних систем без необхідності прямого з'єднання між відправниками та отримувачами повідомлень. Платформа SQS забезпечує надійність зберігання повідомлень через автоматичну реплікацію у декількох зонах доступності, гарантуючи збереження даних навіть при відмові окремих дата-центрів або компонентів інфраструктури AWS. Масштабованість системи дозволяє обробляти практично необмежену кількість повідомлень з автоматичним розподілом навантаження між множиною серверів черги без необхідності ручного втручання або попереднього планування потужностей.

Сервіс Simple Notification Service реалізує модель публікації та підписки для доставки повідомлень одночасно множині отримувачів через різні канали комунікації. Платформа SNS дозволяє надсилати повідомлення через HTTP або HTTPS протоколи до вебхуків для інтеграції з зовнішніми системами та сервісами, електронною поштою для повідомлення користувачів та адміністраторів про важливі події, SMS повідомленнями для критичних сповіщень, що вимагають негайної уваги, інтеграцією з чергами SQS для забезпечення надійності доставки повідомлень з можливістю асинхронної обробки та тригеруванням функцій Lambda для автоматичної обробки подій у реальному часі без необхідності постійно працюючих серверів.

Платформа SNS оптимально підходить для розсилки сповіщень великій кількості підписників одночасно з мінімальною затримкою, моніторингу стану системи з автоматичним повідомленням відповідальних осіб при виникненні проблем або аномалій у роботі компонентів, координації дій між різними мікросервісами через event-driven архітектуру без необхідності прямих викликів

між сервісами та реалізації патерну fan-out для розподілу обробки одного повідомлення між множиною незалежних обробників, кожен з яких виконує свою специфічну функцію над отриманими даними.

Сервіс EventBridge становить serverless платформу для побудови event-driven систем, що дозволяє маршрутизувати події з різних джерел до відповідних обробників без необхідності написання коду інтеграції. Платформа підтримує прийом подій з сервісів AWS, таких як зміни стану інстансів EC2, завершення виконання Lambda функцій або модифікації об'єктів у S3, інтеграцію з понад тридцятьма SaaS партнерами, включаючи Shopify для електронної комерції, Zendesk для підтримки клієнтів та Auth0 для керування автентифікацією, а також власні застосунки через прямий API для відправки користувацьких подій про зміни стану бізнес-процесів [7].

Система правил маршрутизації дозволяє фільтрувати події на основі їх змісту та автоматично направляти відповідні події до призначених обробників, таких як функції Lambda для виконання бізнес-логіки, машини станів Step Functions для оркестрації складних робочих процесів, черги SQS для асинхронної обробки та теми SNS для розповсюдження подій множині підписників. Реєстр схем автоматично виявляє структуру подій, що проходять через шину, та зберігає їх схеми для спрощення розробки обробників подій з можливістю генерації коду на основі збережених схем. Функціональність архівування та повторного відтворення дозволяє зберігати всі події протягом визначеного періоду та відтворювати історичні події для тестування нових обробників або відновлення після збоїв шляхом повторної обробки подій за вказаний проміжок часу.

Система Identity and Access Management забезпечує централізоване керування доступом до всіх ресурсів AWS через гранульовані політики безпеки, ролі з тимчасовими обліковими даними та користувачів з довгостроковими ключами доступу. Принцип найменших привілеїв реалізується через надання кожному компоненту системи мінімального набору дозволів, необхідних для виконання його специфічних функцій, без можливості доступу до інших ресурсів або виконання непотрібних операцій. Механізм IAM ролей дозволяє сервісам AWS

тимчасово отримувати права доступу до інших ресурсів без необхідності зберігання статичних ключів доступу в конфігураційних файлах, що значно підвищує безпеку системи через автоматичну ротацію тимчасових облікових даних [8].

Багатофакторна автентифікація забезпечує додатковий рівень захисту для критичних операцій шляхом вимоги підтвердження особи користувача через фізичний пристрій або мобільний додаток на додачу до звичайного паролю. Федерація ідентичності дозволяє інтегрувати AWS з корпоративними системами керування користувачами, такими як Active Directory або SAML провайдери, дозволяючи співробітникам використовувати існуючі корпоративні облікові дані для доступу до ресурсів AWS без необхідності створення та підтримки окремого набору облікових записів.

Сервіс Secrets Manager забезпечує централізоване безпечне зберігання конфіденційних даних, включаючи паролі баз даних, API ключі для інтеграції з зовнішніми сервісами та токени доступу до різних систем. Функціональність автоматичної ротації періодично змінює паролі до керованих баз даних RDS та Redshift без необхідності ручного втручання та з нульовим часом недоступності застосунків завдяки координації між Secrets Manager та самими базами даних. Всі секрети шифруються за допомогою сервісу Key Management Service з можливістю використання як ключів, керованих AWS, так і власних ключів шифрування для додаткового контролю. Інтеграція з CloudTrail забезпечує повний аудит всіх операцій доступу до секретів з фіксацією часу, користувача та ресурсу, що отримував конфіденційні дані [9].

AWS Systems Manager Parameter Store становить альтернативний сервіс для зберігання конфігураційних параметрів та секретів з базовими можливостями, пропонуючи безкоштовне зберігання параметрів з обмеженим розміром значень та частотою запитів. Сервіс підходить для зберігання несекретних конфігураційних параметрів, таких як адреси ендпоінтів, прапорці функціональності або налаштування середовища, де не потрібна автоматична ротація та розширені можливості керування життєвим циклом секретів.

Сервіс Web Application Firewall захищає веб-застосунки від поширених атак шляхом аналізу HTTP та HTTPS запитів перед їх досягненням серверних застосунків. Правила фільтрації дозволяють блокувати запити на основі IP-адрес джерела для захисту від відомих зловмисників, географічного розташування клієнта для обмеження доступу тільки з дозволених країн або регіонів, аналізу HTTP заголовків для виявлення підозрілих патернів та змісту тіла запиту для запобігання ін'єкціям шкідливого коду. Керовані набори правил від AWS та партнерів надають готові конфігурації для захисту від вразливостей OWASP Top 10 без необхідності глибоких знань у галузі безпеки веб-застосунків.

Правила на основі частоти запитів обмежують кількість запитів з одного IP-адреси протягом визначеного інтервалу часу, захищаючи від брутфорс-атак на форми автентифікації та DDoS атак на рівні застосунку. AWS Shield забезпечує додатковий рівень захисту від розподілених атак типу відмова в обслуговуванні через автоматичне виявлення та пом'якшення наслідків атак на мережевому рівні. Базовий рівень Shield Standard автоматично активний для всіх клієнтів AWS без додаткової плати, захищаючи від найпоширеніших атак на мережевому та транспортному рівнях. Розширений рівень Shield Advanced надає додаткові можливості, включаючи цілодобову підтримку команди реагування на DDoS атаки, фінансову компенсацію витрат на масштабування ресурсів під час атак, розширену телеметрію та звіти про атаки, а також глибшу інтеграцію з WAF для комплексного захисту на всіх рівнях.

Платформа CloudWatch становить централізовану систему моніторингу ресурсів AWS та застосунків з можливістю збору метрик продуктивності від всіх сервісів. Система автоматично збирає стандартні метрики, такі як використання процесора та пам'яті інстансами EC2, кількість запитів до балансувальників навантаження, використання з'єднань до баз даних RDS та затримки відповіді API Gateway, а також підтримує користувацькі метрики для специфічних бізнес-показників, таких як кількість оброблених замовлень, розмір середнього чека або час виконання бізнес-процесів [10].

Функціональність збору та аналізу журналів агрегує логи з різних джерел у

централізоване сховище з можливістю пошуку за ключовими словами, фільтрації за часовими діапазонами та полями структурованих логів, а також кореляції подій між різними компонентами системи. Механізм сповіщень дозволяє налаштувати автоматичні алерти при перевищенні порогових значень метрик або виявленні специфічних патернів у журналах з доставкою повідомлень через SNS до відповідальних осіб або систем автоматичного реагування. Інформаційні панелі забезпечують візуалізацію ключових метрик у реальному часі через налаштовувані віджети графіків, числових показників та таблиць для оперативного контролю стану системи.

Сервіс CloudTrail забезпечує повний аудит всіх дій у AWS аккаунті через логування кожного API виклику з фіксацією користувача або ролі, що виконала операцію, часу виконання, джерела запиту та результату операції. Дані аудиту використовуються для відстеження змін у конфігурації ресурсів для розуміння еволюції інфраструктури та виявлення несанкціонованих модифікацій, забезпечення відповідності регуляторним вимогам різних галузей та юрисдикцій, а також інтеграції з системами керування інформацією та подіями безпеки для комплексного аналізу загроз безпеці організації.

Сервіс Virtual Private Cloud дозволяє створювати ізольовані віртуальні мережі в AWS з повним контролем над топологією мережі, IP адресацією, таблицями маршрутизації та налаштуваннями безпеки на мережевому рівні. Підмережі в межах VPC можуть бути налаштовані як публічні з прямим доступом до Інтернету через Internet Gateway або приватні без прямої маршрутизації до зовнішньої мережі для розміщення внутрішніх сервісів, баз даних та інших компонентів, що не повинні бути доступні ззовні. Групи безпеки реалізують stateful firewall на рівні окремих ресурсів, автоматично дозволяючи вихідний трафік для встановлених з'єднань без необхідності явного правила для зворотного напрямку, тоді як мережеві списки контролю доступу надають stateless фільтрацію на рівні цілих підмереж з необхідністю явного визначення правил для обох напрямків трафіку.

Підключення до корпоративних мереж реалізується через VPN з'єднання для

захищеного зашифрованого тунелю через публічний Інтернет з невисокою вартістю але можливою нестабільністю пропускної здатності або Direct Connect для виділеного приватного каналу з гарантованою пропускною здатністю та нижчою затримкою за вищу вартість. Зв'язок між множиною VPC у різних регіонах або аккаунтах організовується через VPC Peering для прямого приватного з'єднання між двома VPC або Transit Gateway для централізованого керування зв'язністю між десятками VPC та on-premise мереж через єдину точку управління.

Сервіси балансування навантаження Elastic Load Balancing автоматично розподіляють вхідний трафік між множиною цільових ресурсів у різних зонах доступності для забезпечення високої доступності та відмовостійкості застосунків. Application Load Balancer працює на рівні HTTP та HTTPS протоколів з можливістю маршрутизації запитів на основі шляху URL, доменного імені у заголовку Host, HTTP методу або значень заголовків, що дозволяє направляти різні типи запитів до відповідних мікросервісів або версій застосунку на основі вмісту запиту.

Network Load Balancer функціонує на рівні TCP та UDP протоколів з надзвичайно низькою затримкою та можливістю обробки мільйонів запитів на секунду для високопродуктивних застосунків, що вимагають мінімальної обробки на рівні балансувальника. Gateway Load Balancer спеціалізується на розгортанні та масштабуванні мережевих пристроїв сторонніх виробників, таких як firewall, системи виявлення та запобігання вторгненням або системи глибокої перевірки пакетів для забезпечення додаткового рівня безпеки мережевого трафіку.

Сервіс Route 53 становить високодоступну та масштабовану систему доменних імен з можливістю реєстрації нових доменних імен або переносу існуючих доменів від інших реєстраторів. Різні політики маршрутизації дозволяють реалізувати складні схеми розподілу трафіку, включаючи просту маршрутизацію до одного ресурсу, зважену маршрутизацію для розподілу трафіку у визначених пропорціях між множиною ресурсів для поступового впровадження нових версій, маршрутизацію на основі затримки для автоматичного направлення користувачів до найближчого географічно регіону з найменшою мережевою затримкою, автоматичне переключення при відмові для забезпечення високої

доступності та географічну маршрутизацію для обслуговування користувачів з різних країн з локалізованого контенту або дотримання юридичних вимог зберігання даних.

Механізм перевірок стану здоров'я постійно моніторить доступність цільових ендпоінтів через періодичні HTTP, HTTPS або TCP перевірки з автоматичним виключенням недоступних ресурсів з DNS відповідей до відновлення їх працездатності. Тісна інтеграція з іншими сервісами AWS дозволяє автоматично налаштовувати DNS записи для розподілів CloudFront, балансувальників навантаження ELB та статичних веб-сайтів на S3 без необхідності ручного керування DNS конфігурацією.

Для побудови інтеграційного рішення між платформою електронної комерції та державною системою еАкциз можуть застосовуватися різні архітектурні паттерни залежно від специфічних вимог до продуктивності, надійності та складності бізнес-логіки. Синхронна інтеграція через API Gateway з обробкою запитів функціями Lambda забезпечує швидку відповідь користувачам з мінімальною затримкою та простоту налаштування через декларативну конфігурацію API, проте обмежена максимальним часом виконання тридцять секунд для API Gateway та п'ятнадцять хвилин для Lambda, що робить цей підхід непридатним для операцій, що вимагають тривалої обробки або очікування відповіді від повільних зовнішніх систем. Такий паттерн оптимально підходить для отримання актуальних даних з системи еАкциз, швидкої валідації документів перед відправкою та операцій читання довідкової інформації.

Асинхронна інтеграція через черги SQS з обробкою повідомлень функціями Lambda або контейнерами ECS забезпечує слабкий зв'язок між компонентами системи, дозволяючи розробляти та розгортати сервіси незалежно, можливість обробки великих обсягів даних з контрольованою швидкістю споживання для запобігання перевантаженню зовнішніх систем та вбудовані механізми автоматичних повторних спроб при тимчасових збоях зовнішніх API. Складність архітектури зростає через необхідність керування станом асинхронних операцій та інформування користувачів про результати обробки через альтернативні канали,

такі як email повідомлення або персональні кабінети. Цей паттерн доцільно використовувати для відправки нових акцизних марок до системи еАкциз, пакетної синхронізації великих обсягів даних між системами та фонової обробки звітності без блокування користувацького інтерфейсу.

Event-driven інтеграція через EventBridge з оркестрацією складних процесів через Step Functions надає гнучку маршрутизацію подій до множини обробників на основі вмісту події, підтримку багатоетапних робочих процесів з умовною логікою та обробкою помилок, а також просту розширюваність системи через додавання нових підписників на існуючі події без модифікації коду відправників. Вища вартість порівняно з прямими викликами та більша складність налаштування компенсуються гнучкістю та можливостями еволюції системи. Застосування цього паттерну доцільне для реакції на складні події зміни статусу замовлення з залученням множини систем, автоматичного формування та відправки звітності до податкових органів за розкладом та координації дій між різними мікросервісами без створення прямих залежностей.

Використання AWS для побудови інтеграційних рішень надає численні переваги, включаючи широкий набір спеціалізованих сервісів, що покривають всі аспекти побудови повнофункціональної інтеграційної платформи без необхідності залучення сторонніх провайдерів або розробки власних компонентів інфраструктури. Глибока інтеграція між сервісами AWS спрощує налаштування взаємодії компонентів та зменшує накладні витрати на підтримку складних інтеграцій. Ефективне використання AWS вимагає від архітекторів та розробників глибокого розуміння архітектурних принципів Well-Architected Framework, що охоплює операційну досконаліть, безпеку, надійність, ефективність продуктивності та оптимізацію витрат як шість базових стовпів проектування надійних хмарних систем [13].

AWS надає комплексний інструментарій для побудови надійних, масштабованих та безпечних інтеграційних рішень, що дозволяє ефективно поєднувати платформи електронної комерції з державними сервісами обліку та контролю, такими як система еАкциз, забезпечуючи високий рівень автоматизації

бізнес-процесів, відповідність регуляторним вимогам та можливість швидкої адаптації до змін у законодавстві або бізнес-вимогах без необхідності фундаментальної перебудови архітектури системи.

1.4. Аналіз існуючих рішень інтеграції e-commerce з державними службами

Інтеграція платформ електронної комерції з державними службами становить критично важливий аспект забезпечення законності бізнес-операцій, автоматизації звітних процесів та підвищення рівня довіри споживачів до комерційних організацій. Сучасні системи електронної торгівлі функціонують у складному регуляторному середовищі, що вимагає взаємодії з численними державними інформаційними системами для виконання обов'язків щодо податкової звітності, контролю обігу товарів та дотримання галузевих регуляторних вимог.

Європейський Союз впровадив систему One-Stop-Shop для спрощення процесу сплати податку на додану вартість при транскордонній торгівлі між країнами-членами союзу. Система VAT-OSS дозволяє компаніям, що надають послуги або продають товари споживачам з інших країн Європейського Союзу, звітувати та сплачувати податок на додану вартість через єдиний портал у країні реєстрації без необхідності реєстрації як платників податків у кожній країні, де здійснюються продажі. Система IOSS спрощує процес імпорту товарів обмеженої вартості з третіх країн поза межами Європейського Союзу, дозволяючи продавцям сплачувати податок на додану вартість безпосередньо при продажі замість покладання цієї відповідальності на покупців при імпорті товарів через митні кордони.

Технічна реалізація цих систем у провідних платформах електронної комерції виконана через спеціалізовані модулі та плагіни, які автоматично розраховують ставки податку на додану вартість залежно від країни розташування покупця та типу товару або послуги, генерують податкову звітність у відповідних форматах для кожної юрисдикції та передають дані до податкових органів через

програмні інтерфейси без необхідності ручного втручання операторів. Ключові виклики впровадження таких систем включають складність визначення коректних ставок податку через значні відмінності залежно від країни та категорії товару, необхідність тривалого зберігання податкових документів відповідно до національних вимог різних юрисдикцій та відмінності в національних імплементаціях європейських директив, що вимагають адаптації технічних рішень під специфіку кожної країни.

Італія запровадила обов'язкове електронне інвойсування через централізовану систему Sistema di Interscambio, що революціонізувала процеси податкової звітності у країні. Всі комерційні транзакції між бізнесом та бізнесом, а також між бізнесом та споживачами повинні оформлюватися електронними рахунками у форматі XML згідно зі стандартом FatturaPA, підписаними кваліфікованим електронним підписом відповідальної особи організації. Централізована система SDI перевіряє валідність структури та змісту рахунків відповідно до встановлених правил, маршрутизує документи отримувачам через електронні канали комунікації та зберігає копії всіх документів для податкових органів з можливістю подальшого аналізу та аудиту.

Компанії інтегрують свої системи планування ресурсів підприємства та платформи електронної комерції з системою SDI через програмні інтерфейси на основі протоколів SOAP або REST залежно від специфіки інтеграційного середовища. Використання middleware систем, таких як мережа PEPPOL для уніфікації обміну електронними документами між різними організаціями та країнами, спрощує технічну реалізацію інтеграцій. Інтеграція з хмарними бухгалтерськими системами на кшталт QuickBooks, Xero або SAP Business ByDesign дозволяє автоматично генерувати XML документи з форматом FatturaPA безпосередньо з фінансових записів про транзакції без необхідності ручного введення даних операторами.

Впровадження обов'язкового електронного інвойсування в Італії продемонструвало вражаючі результати з точки зору ефективності податкового адміністрування. Зменшення податкових ухилень досягло рівня трьох з половиною

мільярдів євро на рік завдяки прозорості всіх комерційних транзакцій для податкових органів. Час обробки рахунків у компаніях зменшився в середньому на п'ятдесят відсотків через автоматизацію процесів створення, відправки та отримання податкових документів. Прозорість податкових операцій значно підвищилася завдяки централізованому зберіганню всіх комерційних документів у державній системі з можливістю перехресної перевірки даних між різними контрагентами.

Бразилія впровадила одну з найбільш комплексних систем електронних податкових документів у світі через систему Nota Fiscal Eletrônica, що супроводжує кожну транзакцію товарів від виробника до кінцевого споживача. Електронний податковий документ NF-e містить детальну інформацію про товари, їх кількість, вартість, податки та сторони транзакції з обов'язковим цифровим підписом відправника. Документи повинні бути створені та затверджені податковою службою штату SEFAZ до або під час відправки товару покупцю, що забезпечує контроль обігу товарів у реальному часі без можливості реалізації немаркованої продукції.

Система NF-e інтегрується з транспортними документами CT-e та системами відстеження переміщення вантажів, створюючи комплексну екосистему контролю логістичних операцій від складу відправника до складу отримувача з фіксацією всіх проміжних етапів транспортування. Архітектура інтеграції базується на Web Services з підтримкою протоколу SOAP для взаємодії систем планування ресурсів підприємства з податковою службою штату. Використання сертифікатів цифрового підпису типу A1 у вигляді файлів або A3 на фізичних носіях як смарт-карти або криптографічні токени забезпечує автентифікацію та цілісність електронних документів. Зберігання XML-документів з часовою міткою від податкової служби гарантує юридичну силу документів та можливість їх використання як доказів у судових процесах.

Основні платформи електронної комерції Бразилії, включаючи VTEX, Magento та Shopify Plus для корпоративних клієнтів, мають нативну підтримку генерації та відправки документів NF-e без необхідності додаткових модулів або

інтеграцій. Автоматична генерація NF-e відбувається при створенні замовлення з синхронізацією з системою управління складом для перевірки наявності товарів та резервування позицій під конкретне замовлення. Інтеграція з транспортними компаніями дозволяє автоматично формувати супровідні документи CT-e при передачі товару перевізнику з прив'язкою до відповідного документа NF-e через унікальні ідентифікатори.

Виклики впровадження системи маркування включають високу складність інтеграції через велику кількість типів товарів з різними правилами маркування та документообігу для кожної товарної категорії. Необхідність модернізації складської логістики для роботи з індивідуальними кодами маркування на рівні окремих одиниць товару замість партій вимагає впровадження систем автоматичної ідентифікації з використанням сканерів штрих-кодів або систем машинного зору. Штрафи за порушення правил маркування можуть досягати трьохсот тисяч рублів для юридичних осіб, що створює значні фінансові ризики при неякісній реалізації інтеграційних рішень.

У багатьох країнах, включаючи Україну, Казахстан та країни Європейського Союзу, платформи електронної комерції зобов'язані використовувати онлайн-каси або програмні реєстратори розрахункових операцій для фіскалізації кожної транзакції з автоматичною передачею даних до податкових органів. Кожна оплата повинна бути зареєстрована у фіскальній пам'яті з генерацією унікального фіскального номера документа та передачею інформації про транзакцію до серверів податкової служби протягом визначеного регуляторного часового інтервалу. Надання електронного чеку покупцю здійснюється через відправку на електронну пошту, SMS-повідомлення з посиланням на чек або відображення QR-коду для сканування мобільним додатком податкової служби.

Пряма інтеграція платформи електронної комерції з програмним реєстратором розрахункових операцій передбачає безпосередній виклик програмних інтерфейсів касового програмного забезпечення з системи управління замовленнями для створення фіскального документа при підтвердженні оплати. Переваги такого підходу включають мінімальну затримку між отриманням оплати

та створенням фіскального чека, а також повний контроль над процесом фіскалізації без залежності від третіх сторін. Недоліки полягають у необхідності підтримки специфічних програмних інтерфейсів різних провайдерів касового програмного забезпечення, що ускладнює масштабування рішення при роботі з множиною постачальників.

Інтеграція через сервіс-агрегатор передбачає використання проміжного сервісу, такого як Checkbox або ЕККА Online, що надає уніфікований програмний інтерфейс для роботи з різними провайдерами касового програмного забезпечення через єдиний набір методів. Переваги включають незалежність від конкретного провайдера касового рішення з можливістю зміни постачальника без модифікації коду платформи електронної комерції та спрощену інтеграцію через стандартизований інтерфейс. Недоліки полягають у додаткових комісіях за використання послуг агрегатора та залежності від доступності та надійності сервісу третьої сторони.

Інтеграція через платіжний шлюз дозволяє делегувати процес фіскалізації безпосередньо платіжному процесору, такому як LiqPay, WayForPay або Fondy, що надають функціонал автоматичної генерації фіскальних чеків при успішній обробці платежу. Переваги включають мінімальні зміни в архітектурі платформи електронної комерції через використання існуючої інтеграції з платіжним шлюзом та комплексне рішення, що поєднує обробку оплати та фіскалізацію в єдиному процесі. Недоліки полягають у обмеженій гнучкості налаштування фіскальних документів та можливих конфліктах при використанні кількох платіжних систем з різними підходами до фіскалізації.

Типовий workflow обробки замовлення з фіскалізацією починається з оформлення замовлення користувачем через інтерфейс платформи електронної комерції з вибором товарів, вказанням даних доставки та способу оплати. Платформа електронної комерції ініціює платіж через інтеграцію з обраним платіжним шлюзом з передачею інформації про суму, призначення платежу та ідентифікатор замовлення. Платіжний шлюз обробляє оплату через взаємодію з банком-еквайром та платіжними системами з перевіркою достатності коштів та

автентифікацією платника. При успішній оплаті платформа електронної комерції отримує підтвердження від платіжного шлюзу та викликає програмний інтерфейс програмного реєстратора розрахункових операцій для створення фіскального чека з деталями транзакції. Програмний реєстратор генерує фіскальний чек з унікальним фіскальним номером та відправляє дані про транзакцію до серверів податкової служби через захищений канал комунікації. Після підтвердження успішної реєстрації чека податковою службою програмний реєстратор повертає посилання на електронний чек платформі електронної комерції. Платформа електронної комерції надсилає електронний чек клієнту через електронну пошту або SMS-повідомлення з посиланням на перегляд деталей фіскального документа.

Система еАкциз в Україні становить електронну систему адміністрування обігу пального, алкогольних напоїв та тютюнових виробів, що вимагає обов'язкового маркування акцизних марок для контролю сплати акцизного податку та легальності обігу підакцизних товарів. Марка акцизного податку являє собою унікальний ідентифікатор у вигляді захищеного друкованого знака або електронного коду, що підтверджує сплату акцизного податку на конкретну одиницю товару та дозволяє відстежувати її переміщення через ланцюг постачання від виробника до кінцевого споживача. Електронна звітність вимагає від суб'єктів господарювання подання електронних звітів про отримання акцизних марок від податкових органів або виробників, використання марок при виробництві або маркуванні товарів, переміщення маркованих товарів між складами або контрагентами та повернення невикористаних або пошкоджених марок до податкових органів.

Контроль обігу забезпечується через відстеження кожної марки від моменту її видачі податковими органами через всі етапи обігу товару до виведення марки з обігу при роздрібному продажу кінцевому споживачу з фіксацією всіх операцій у централізованій державній базі даних. Ручне введення даних через веб-інтерфейс порталу Державної податкової служби передбачає внесення інформації про акцизні марки операторами вручну через форми веб-інтерфейсу без автоматизованої інтеграції систем. Такий підхід не потребує технічної інтеграції та може бути

реалізований без залучення розробників програмного забезпечення, проте характеризується високою ймовірністю помилок через людський фактор при введенні номерів марок та інших даних, великими трудовитратами на обробку кожної операції та неможливістю масштабування для великого обсягу операцій з акцизними марками.

Застосування ручного підходу обмежується малими роздрібними точками з незначним обсягом продажу акцизних товарів, де кількість операцій не виправдовує інвестиції в автоматизацію процесів. Інтеграція через програмні інтерфейси системи еАкциз передбачає програмну взаємодію платформи електронної комерції або системи планування ресурсів підприємства з державною системою для автоматичної передачі даних про операції з акцизними марками без ручного втручання операторів. Автентифікація здійснюється з використанням кваліфікованого електронного підпису відповідальної особи організації для підтвердження легітимності операцій та забезпечення юридичної сили електронних документів. Формати даних базуються на XML документах згідно зі специфікаціями та схемами, опублікованими Державною податковою службою, що визначають структуру та обов'язкові поля для різних типів операцій.

Переваги програмної інтеграції включають автоматизацію процесів обміну даними з мінімальним залученням людських ресурсів, зменшення помилок через виключення ручного введення даних та автоматичну валідацію перед відправкою, можливість обробки великого обсягу даних без пропорційного збільшення трудовитрат та забезпечення своєчасності подання звітності через автоматичні процеси без затримок. Недоліки включають складність впровадження та підтримки інтеграції через необхідність глибоких технічних знань специфікації програмних інтерфейсів, необхідність інфраструктури для роботи з кваліфікованим електронним підписом з підтримкою криптографічних операцій та регулярні зміни в специфікаціях програмних інтерфейсів, що вимагають оновлення інтеграційного коду.

Застосування прямої програмної інтеграції доцільне для середніх та великих ритейлерів з значним обсягом операцій з акцизними марками, оптових продавців

акцизних товарів, що здійснюють множину операцій переміщення марок між складами та контрагентами, та виробників підакцизних товарів, що маркують великі партії продукції власними акцизними марками. Інтеграція через middleware-рішення передбачає використання спеціалізованих програмних продуктів, таких як M.E.Doc або SMARTFIN, що надають уніфікований інтерфейс для роботи з різними державними системами через єдину платформу без необхідності окремої інтеграції з кожною системою.

Функціонал middleware систем включає генерацію XML документів згідно з актуальними специфікаціями різних державних систем, управління кваліфікованим електронним підписом з підтримкою різних криптопровайдерів та форматів сертифікатів, валідацію даних перед відправкою для виявлення помилок на етапі формування документів та відправку і отримання документів через захищені канали комунікації з державними системами. Переваги використання middleware включають спрощену інтеграцію для бізнесу через абстракцію складності державних програмних інтерфейсів, підтримку різних державних систем в єдиному продукті без необхідності множини окремих інтеграцій та оновлення при зміні законодавства або технічних специфікацій, що виконуються провайдером middleware без необхідності модифікації бізнес-систем.

Недоліки включають ліцензійні витрати на використання комерційного програмного забезпечення з щорічними платежами за підтримку, залежність від третьої сторони з ризиками доступності сервісу та якості технічної підтримки, а також можливу відсутність гнучкості для реалізації специфічних бізнес-процесів, що не передбачені стандартним функціоналом middleware. Застосування middleware-рішень доцільне для компаній, що працюють з кількома державними системами одночасно, включаючи еАкциз, електронну звітність до Державної податкової служби, Електронний кабінет та інші системи електронної взаємодії з державними органами.

Програмні інтерфейси системи еАкциз використовують протокол SOAP Web Services для взаємодії систем з підтримкою XML-повідомлень для запитів та відповідей. Основні методи включають RegistrationExciseStamps для реєстрації

акцизних марок при їх отриманні від податкових органів або виробників з передачею списку номерів марок та супровідної інформації, MovementExciseStamps для фіксації передачі марок між складами або контрагентами з зазначенням відправника, отримувача та переліку марок, ReturnExciseStamps для повернення невикористаних або пошкоджених марок до податкових органів з обґрунтуванням причин повернення, SaleExciseStamps для реєстрації продажу товару з акцизною маркою кінцевому споживачу з виведенням марки з обігу та GetStatus для отримання поточного статусу конкретної марки або групи марок з інформацією про всі операції в історії обігу.

Формат даних базується на XML-документах, що відповідають XSD-схемам, опублікованим Державною податковою службою, які визначають структуру документів, обов'язкові та опціональні поля, типи даних та валідаційні правила. Автентифікація та підпис вимагають, щоб кожен запит до системи еАкциз був підписаний кваліфікованим електронним підписом відповідальної особи організації, такої як директор або головний бухгалтер, з використанням сертифіката, виданого акредитованим центром сертифікації ключів.

Workflow інтеграції платформи електронної комерції з системою еАкциз для операції отримання акцизних марок починається з замовлення марок компанією у Державній податковій службі або отримання маркованого товару від виробника з супровідними документами. При отриманні марок оператор на складі або автоматизована система сканує штрих-коди або QR-коди марок для фіксації їх номерів у внутрішній базі даних. Система планування ресурсів підприємства або управління складом реєструє марки у внутрішній базі даних з прив'язкою до конкретних товарних позицій та місць зберігання на складі. Інтеграційний модуль формує XML-документ про отримання марок згідно зі специфікацією програмного інтерфейсу RegistrationExciseStamps з включенням списку номерів марок, дати отримання, відомостей про постачальника та інших обов'язкових реквізитів. Документ підписується кваліфікованим електронним підписом відповідальної особи через криптографічну бібліотеку та відправляється до системи еАкциз через SOAP запит. Система еАкциз перевіряє валідність підпису, структуру документа та

відповідність даних внутрішнім записам, після чого підтверджує отримання та змінює статус марок на активні в обігу з фіксацією поточного власника.

Переміщення марок між складами організації вимагає формування документа про внутрішнє переміщення товару з акцизними марками між різними місцями зберігання. Інтеграційний модуль створює XML-документ через метод MovementExciseStamps з зазначенням складу відправника, складу отримувача, переліку марок, що переміщуються, та причини переміщення. Документ підписується кваліфікованим електронним підписом та відправляється до системи еАкциз для реєстрації операції. Після підтвердження системою еАкциз марки прив'язуються до нового місця зберігання у внутрішній базі даних з оновленням залишків на складах.

Продаж товару з акцизною маркою кінцевому споживачу через платформу електронної комерції починається з оформлення замовлення клієнтом через веб-інтерфейс або мобільний додаток з вибором товарів, вказанням даних доставки та способу оплати. Платформа електронної комерції перевіряє наявність марок для обраних товарів у системі управління складом з резервуванням конкретних марок під замовлення для запобігання подвійного продажу. Після успішної оплати замовлення та комплектації товару на складі з фізичним відбором позицій з акцизними марками інтеграційний модуль формує документ про продаж марок через метод SaleExciseStamps. Документ відправляється до системи еАкциз з інформацією про марки, що були реалізовані, дату продажу, суму транзакції та дані фіскального чека. Система еАкциз змінює статус марок на виведені з обігу з фіксацією факту роздрібного продажу кінцевому споживачу. Фіскальний чек з інформацією про акцизні марки генерується програмним реєстратором розрахункових операцій та надсилається клієнту через електронну пошту або SMS-повідомлення.

Обробка повернень товару клієнтом вимагає реєстрації операції повернення у платформі електронної комерції з створенням відповідного документа в системі управління замовленнями. Інтеграційний модуль формує документ про повернення марок до обігу з використанням відповідного методу програмного інтерфейсу

еАкциз. Статус марок у системі еАкциз змінюється на активні в обігу з можливістю їх подальшого продажу іншим покупцям після повернення товару на склад.

Забезпечення надійності інтеграції вимагає впровадження механізмів обробки помилок та відновлення після збоїв. Retry-механізми забезпечують автоматичні повторні спроби відправки документів до системи еАкциз при тимчасовій недоступності програмного інтерфейсу або мережевих проблемах з експоненціальним збільшенням інтервалу між спробами. Черги повідомлень на основі технологій як RabbitMQ або AWS SQS забезпечують асинхронну обробку операцій з еАкциз з можливістю накопичення повідомлень під час недоступності зовнішньої системи та їх обробки після відновлення доступності. Детальне логування всіх операцій інтеграції з фіксацією запитів, відповідей, помилок та часових міток забезпечує можливість аудиту та діагностики проблем при виникненні інцидентів. Моніторинг ключових метрик інтеграції, таких як кількість успішних операцій, частота помилок, час відповіді програмного інтерфейсу та розмір черги необроблених повідомлень, дозволяє виявляти проблеми на ранніх стадіях та налаштовувати алерти для сповіщення відповідальних осіб.

Система планування ресурсів підприємства 1С:Підприємство становить найпопулярніше рішення в Україні з вбудованою підтримкою електронної звітності та інтеграції з державними системами через спеціалізовані конфігурації для різних галузей. Модуль еАкциз забезпечує автоматичну генерацію XML документів згідно з актуальними специфікаціями Державної податкової служби, відправку документів до системи еАкциз через вбудовані механізми комунікації та синхронізацію статусів марок між внутрішньою базою даних та державною системою. Інтеграція з платформами електронної комерції здійснюється через REST програмні інтерфейси або файлові обміни у форматі CommerceML для синхронізації каталогів товарів, цін, залишків та замовлень між системами. Управління кваліфікованим електронним підписом реалізовано через вбудовані інструменти для роботи з різними криптопровайдерами та форматами сертифікатів без необхідності зовнішніх компонентів.

Недоліки використання системи 1С включають високу складність

налаштування для нестандартних бізнес-процесів через орієнтацію на типові сценарії роботи, обмежену підтримку сучасних технологій архітектури мікросервісів, контейнеризації та хмарних платформ, а також високу вартість ліцензій та послуг впровадження для середніх та великих підприємств. SAP Business One та SAP S/4HANA пропонують рішення для середнього та великого бізнесу з можливістю локалізації під специфічні вимоги українського законодавства через партнерські програми. Локалізаційні пакети забезпечують підтримку української податкової звітності, включаючи інтеграцію з системою еАкциз через партнерські рішення від системних інтеграторів.

Інтеграція з платформами електронної комерції реалізується через модулі SAP Commerce Cloud для побудови повнофункціональних систем електронної торгівлі з глибокою інтеграцією з системою планування ресурсів підприємства на рівні єдиної бази даних та бізнес-процесів. Підхід на основі програмних інтерфейсів через SAP API Business Hub забезпечує можливість побудови інтеграцій з зовнішніми системами через стандартизовані REST та OData програмні інтерфейси з документованими специфікаціями. Недоліки включають дуже високу вартість впровадження та супроводу системи SAP, що робить рішення недоступним для малого та середнього бізнесу, а також значну складність архітектури для організацій без досвіду роботи з корпоративними системами SAP.

Популярні платформи електронної комерції Shopify, WooCommerce та Magento не мають нативної підтримки інтеграції з системою еАкциз через їх орієнтацію на міжнародний ринок без специфічної локалізації під українське законодавство. Інтеграція можлива через розробку кастомних плагінів та модулів, що реалізують взаємодію з програмними інтерфейсами еАкциз з використанням мов програмування PHP для WooCommerce та Magento або Ruby для Shopify. Інтеграційні платформи як Zapier, Make або n8n дозволяють створювати автоматизовані workflow для зв'язку платформи електронної комерції з middleware-системами без написання програмного коду через візуальні інструменти. Побудова окремих інтеграційних мікросервісів забезпечує зв'язок між платформою електронної комерції та системою еАкциз через незалежний сервіс з власною базою

даних, бізнес-логікою та програмними інтерфейсами.

Переваги такого підходу включають гнучкість та можливість кастомізації під специфічні бізнес-вимоги без обмежень стандартних рішень, а також нижчу вартість порівняно з enterprise-рішеннями для організацій з обмеженими бюджетами. Недоліки полягають у необхідності розробки та підтримки власних інтеграційних рішень з виділенням ресурсів розробників та відсутності стандартизованих підходів, що ускладнює міграцію між платформами та обмін досвідом між організаціями.

Хмарні інтеграційні платформи типу Integration Platform as a Service, такі як Dell Boomi, MuleSoft або Microsoft Azure Logic Apps, дозволяють швидко побудувати інтеграції між різними системами без написання великої кількості програмного коду через використання візуальних інструментів моделювання. Drag-and-drop інтерфейси для створення інтеграційних потоків забезпечують можливість побудови складних сценаріїв обробки даних через графічне з'єднання компонентів без знання мов програмування. Готові коннектори для популярних систем, включаючи Salesforce, SAP, Oracle, Shopify та багато інших, спрощують інтеграцію через використання преконфігурованих адаптерів з підтримкою специфічних програмних інтерфейсів кожної системи.

Трансформація даних між різними форматами, такими як JSON, XML або CSV, реалізується через вбудовані функції перетворення з можливістю визначення правил маппінгу полів між структурами різних систем. Централізований моніторинг та логування всіх інтеграційних потоків з можливістю відстеження виконання кожної операції та діагностики помилок спрощує підтримку складних інтеграційних ландшафтів. Застосування для інтеграції з системою еАкциз включає створення кастомного коннектора для програмних інтерфейсів еАкциз з інкапсуляцією специфіки роботи з SOAP та XML, побудову інтеграційного потоку від платформи електронної комерції через інтеграційну платформу до системи еАкциз з трансформацією даних та обробку подій з платформи електронної комерції, таких як нове замовлення або повернення товару, з автоматичною відправкою даних до системи еАкциз.

Виклики використання хмарних інтеграційних платформ включають високу вартість ліцензування для малого бізнесу з обмеженими бюджетами на інформаційні технології та обмеження щодо специфічних протоколів та форматів, таких як робота з кваліфікованим електронним підписом, що може вимагати розробки кастомних компонентів.

Архітектурні принципи успішних інтеграцій включають слабкий зв'язок між компонентами, коли платформа електронної комерції не повинна безпосередньо викликати програмні інтерфейси системи еАкциз для уникнення блокування користувацьких операцій при проблемах зовнішньої системи. Використання черг повідомлень та архітектури на основі подій забезпечує незалежність компонентів з можливістю обробки операцій асинхронно. Асинхронна обробка операцій з системою еАкциз запобігає блокуванню процесу оформлення замовлення очікуванням відповіді від зовнішньої системи з використанням фонових задач та черг для обробки інтеграції без впливу на користувацький досвід.

Ідемпотентність операцій забезпечує, що повторна відправка того ж запиту до системи еАкциз не призводить до дублювання даних або некоректної зміни статусу марок через використання унікальних ідентифікаторів запитів та перевірку стану перед виконанням операції. Circuit Breaker Pattern передбачає тимчасове припинення спроб відправки запитів до системи еАкциз при повторюваних помилках для запобігання перевантаженню системи невдалими запитами з автоматичним відновленням спроб після періоду очікування. Моніторинг та алертинг через налаштування метрик продуктивності та помилок інтеграції з автоматичним сповіщенням відповідальних осіб дозволяє своєчасно виявляти та усувати проблеми до їх критичного впливу на бізнес-процеси.

Версіонування програмних інтерфейсів інтеграційного шару забезпечує підтримку кількох версій одночасно для забезпечення плавного оновлення клієнтських систем без примусової зупинки сервісу. Використання шару адаптерів для ізоляції специфіки програмних інтерфейсів системи еАкциз від бізнес-логіки платформи електронної комерції дозволяє змінювати імплементацію інтеграції без впливу на основний код застосунку. Регресійне тестування через автоматичні тести

для перевірки коректності роботи інтеграції після змін у коді або оновлення версій зовнішніх систем запобігає деградації функціональності при еволюції системи.

Безпека інтеграційних рішень вимагає захисту приватних ключів кваліфікованого електронного підпису через зберігання у захищених сховищах, таких як AWS Secrets Manager, Azure Key Vault або HashiCorp Vault, з контролем доступу та аудитом операцій читання. Шифрування всіх комунікацій з програмними інтерфейсами системи еАкциз через використання протоколу TLS версії 1.2 або вище забезпечує конфіденційність та цілісність даних при передачі через незахищені мережі. Обмеження доступу до інтеграційних сервісів через налаштування мінімально необхідних прав доступу відповідно до принципу найменших привілеїв запобігає несанкціонованому використанню або модифікації критичних компонентів. Логування всіх операцій з акцизними марками з фіксацією користувача, часу, типу операції та результату забезпечує можливість аудиту для відповідності регуляторним вимогам та розслідування інцидентів безпеки.

Специфічні виклики української екосистеми включають відсутність стабільності програмних інтерфейсів через регулярні зміни специфікацій програмних інтерфейсів системи еАкциз без належного попередження та тестового періоду для адаптації бізнес-систем. Рішення полягає у створенні адаптаційного шару, що ізолює основну бізнес-логіку від специфіки зовнішніх програмних інтерфейсів з можливістю швидкої зміни імплементації без перероблення всієї системи, а також моніторингу оновлень законодавства та технічної документації Державної податкової служби через підписку на офіційні канали комунікації.

Обмежена документація та підтримка проявляється у недостатньо детальній технічній документації програмних інтерфейсів з неповним описом форматів даних, обов'язкових полів та валідаційних правил, а також відсутності sandbox-середовища для тестування інтеграцій без ризику помилкових операцій у виробничій системі. Рішення включає активну участь у спільнотах розробників через форуми та Telegram-групи для обміну досвідом та рішеннями типових проблем, а також створення власних тестових стендів на основі опису протоколу з емуляцією поведінки державної системи для відпрацювання сценаріїв інтеграції.

Необхідність роботи з кваліфікованим електронним підписом створює складність інтеграції з криптопровайдерами через специфічні бібліотеки та драйвери, обмежену підтримку на різних операційних системах, особливо Linux та macOS, що ускладнює розгортання у хмарних середовищах, а також складність автоматизації операцій підпису без ручного втручання через вимоги безпеки зберігання приватних ключів. Рішення включає використання cross-platform бібліотек, таких як IIT EndUser Library від Інституту інформаційних технологій, що забезпечують роботу на різних операційних системах, контейнеризацію компонентів, що працюють з кваліфікованим електронним підписом, для ізоляції залежностей та спрощення розгортання, або використання middleware-рішень, що беруть на себе відповідальність за операції підпису та шифрування.

Високі штрафи за порушення термінів реєстрації марок або продаж немаркованих акцизних товарів можуть становити значні суми, що створює фінансові ризики для бізнесу при неякісній реалізації або збоях інтеграційних рішень. Рішення включає впровадження додаткових перевірок на рівні бізнес-логіки для валідації даних перед відправкою до зовнішніх систем, попередження операторів про наближення термінів подання звітності або виконання обов'язкових операцій через систему нотифікацій та автоматизацію контролю виконання обов'язкових операцій з налаштуванням алертів при виявленні аномалій або порушень встановлених правил.

Аналіз існуючих підходів до інтеграції платформ електронної комерції з державними службами, зокрема системою еАкциз, демонструє ефективність міжнародного досвіду електронних систем контролю обігу підакцизних товарів та податкової звітності для зменшення ухилень від сплати податків, спрощення адміністративних процесів та підвищення прозорості бізнес-операцій. Технологічні підходи варіюються від ручного введення даних через веб-інтерфейси до повноцінної автоматизації через програмні інтерфейси з вибором залежно від масштабу бізнесу, обсягу операцій та доступних фінансових і технічних ресурсів організації.

Middleware-рішення та хмарні інтеграційні платформи спрощують технічну

реалізацію інтеграцій через абстракцію складності державних програмних інтерфейсів, проте створюють залежність від третіх сторін та додаткові операційні витрати на ліцензування та підтримку зовнішніх систем. Хмарні технології Amazon Web Services надають необхідний інструментарій для побудови масштабованих, надійних та безпечних інтеграційних рішень з можливістю адаптації до змін у законодавстві та вимогах бізнесу через гнучкість архітектури та широкий спектр доступних сервісів.

Специфічні виклики української екосистеми, включаючи нестабільність програмних інтерфейсів державних систем, обмежену документацію та високі штрафи за порушення, вимагають гнучких архітектурних рішень з використанням принципів слабкого зв'язування, асинхронної обробки та ретельного моніторингу для забезпечення високого рівня надійності та відповідності регуляторним вимогам. На основі цього аналізу можна сформулювати вимоги до проектованого рішення інтеграції платформи електронної комерції з системою еАкциз на базі хмарних технологій AWS, що враховуватиме кращі практики міжнародного досвіду, уникатиме типових помилок існуючих рішень та забезпечуватиме ефективну роботу в умовах української специфіки з можливістю масштабування та адаптації до майбутніх змін.

2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ХМАРНОЇ СИСТЕМИ

2.1. Концептуальна модель хмарної системи інтеграції

Проектування архітектури інтеграційного рішення для зв'язку e-commerce платформи з системою еАкциз вимагало детального аналізу функціональних та нефункціональних вимог до системи. На початковому етапі було визначено основні виклики, з якими система стикатиметься під час експлуатації в реальних умовах.

Аналіз бізнес-процесів інтернет-магазину з продажу підакцизних товарів виявив необхідність одночасного вирішення кількох критичних завдань. Система повинна забезпечувати обробку замовлень у реальному часі з мінімальною затримкою, оскільки швидкість резервування акцизних марок безпосередньо впливає на користувацький досвід. Інтеграція з державним API вимагала врахування особливостей роботи системи еАкциз, включаючи обмежену документацію, періодичні зміни специфікацій та можливі технічні перебої. Дотримання вимог податкового законодавства передбачало документування кожної операції з акцизними марками, накладання електронного підпису та збереження даних протягом встановленого законом терміну.

Масштабованість системи під час пікових навантажень стала одним з ключових факторів при виборі архітектурного підходу. Бізнес алкогольної роздрібної торгівлі характеризується вираженими сезонними піками, коли навантаження на систему може зростати в десятки разів, що робить неможливим використання традиційної інфраструктури з фіксованими обчислювальними ресурсами.

Вибір хмарної платформи AWS як основи для системи був обумовлений кількома факторами. Порівняльний аналіз варіантів включав власний дата-центр, Microsoft Azure, Google Cloud та Amazon Web Services. Власна інфраструктура була виключена через значні капітальні витрати та складність забезпечення високої доступності. Платформа AWS перемогла завдяки найширшому набору сервісів, зрілості екосистеми, наявності регіонів в Європі для дотримання вимог латентності

та якісній документації.

При проектуванні системи ми керувалися кількома фундаментальними принципами, які визначили структуру всього рішення.

Розділення відповідальностей (Separation of Concerns)

Замість монолітного додатку, де все змішано в одній кодовій базі, ми розділили систему на окремі рівні з чітко визначеними функціями:

Презентаційний рівень - це e-commerce фронтенд, де користувачі переглядають товари та оформляють замовлення. Він не знає нічого про еАкциз і працює як звичайний інтернет-магазин.

API Gateway рівень - точка входу для всіх запитів. Тут відбувається автентифікація, rate limiting, маршрутизація. Це наш "швейцар", який контролює, хто і куди може потрапити.

Бізнес-логіка - контейнери з додатками, що обробляють замовлення, керують складськими залишками, формують документи. Саме тут живе основна логіка роботи з акцизними марками.

Інтеграційний рівень - окремі сервіси, що спілкуються з зовнішніми системами: API еАкциз, платіжні шлюзи, служби доставки. Якщо потрібно змінити спосіб інтеграції з еАкциз, ми чіпаємо лише цей рівень.

Рівень даних - бази даних, черги повідомлень, файлові сховища. Тут зберігається стан системи.

Така структура дозволяє розробляти та оновлювати кожен рівень незалежно. Наприклад, ми можемо переписати фронтенд з React на Vue, і це не вплине на інтеграцію з еАкциз.

Асинхронність де це можливо

Застосування асинхронної обробки стало критичним архітектурним рішенням. При оформленні замовлення відсутня необхідність синхронного очікування підтвердження від системи еАкциз, оскільки це може займати до десяти секунд, що негативно впливає на користувацький досвід. Натомість реалізовано механізм швидкої валідації та резервування товару з поверненням негайного підтвердження користувачу, тоді як відправка даних до системи еАкциз

виконується в фоновому режимі через черги повідомлень Amazon SQS. Окремі воркер-процеси обробляють завдання з черги з механізмом автоматичних повторних спроб у разі виникнення помилок.

Ідемпотентність операцій забезпечується шляхом використання унікальних ідентифікаторів для кожної транзакції. В розподілених системах існує ймовірність повторної відправки запиту через мережеві затримки або таймаути, тому всі критичні операції спроектовано з урахуванням можливості багаторазового виконання без створення дублікатів. Технічна реалізація базується на генерації унікальних ідентифікаторів операцій та їх перевірці в базі даних перед обробкою запиту.

Багаторівневий підхід до забезпечення безпеки передбачає захист на кожному рівні архітектури. На мережевому рівні застосовано віртуальні приватні хмари з ізольованими підмережами, групи безпеки та списки контролю доступу. Додатковий рівень захищено брандмауером веб-додатків та обмеженням кількості запитів через API Gateway. На рівні даних використовується шифрування інформації в базах даних та Secrets Manager для управління конфіденційними даними. Рівень доступу регулюється політиками мінімальних привілеїв та обов'язковою багатофакторною автентифікацією. Рівень аудиту забезпечує централізоване логування через CloudTrail з автоматичним сповіщенням про підозрілі активності.

Концептуальна схема системи

Архітектура системи реалізована як триланкова структура з чітким розділенням функціональних рівнів. Фронтальна частина включає мережу доставки контенту CloudFront для оптимізації швидкості передачі статичних ресурсів, сховище S3 для розміщення односторінкового веб-додатку та API Gateway як єдину точку входу для серверних запитів. Рівень додатків побудовано на основі кластера ECS з контейнеризованими компонентами бізнес-логіки, платформи Fargate для автоматизованого управління обчислювальними ресурсами, балансувальника навантаження для розподілу трафіку між інстансами та безсерверних функцій Lambda для обробки специфічних завдань та інтеграційних

процесів. Рівень даних представлено реляційною базою даних Aurora PostgreSQL для транзакційної інформації, службою кешування ElastiCache Redis для оптимізації продуктивності, об'єктним сховищем S3 для документів та архівних даних, а також чергами повідомлень SQS для забезпечення асинхронної обробки. Інтеграційний шар містить виділений мікросервіс для взаємодії з API системи еАкциз, криптографічний модуль для формування електронних підписів та шину подій EventBridge для організації event-driven комунікації між компонентами.

Потоки даних в системі

Типовий сценарій роботи системи розглянуто на прикладі процесу оформлення замовлення. На першому етапі користувач здійснює вибір товару через веб-інтерфейс, після чого браузер ініціює захищений HTTPS-запит до мережі доставки контенту CloudFront, яка перенаправляє його до API Gateway. На другому етапі шлюз виконує верифікацію JWT-токена для ідентифікації користувача, застосовує обмеження швидкості запитів та маршрутизує трафік до балансувальника навантаження додатків.

Третій етап передбачає обробку замовлення контейнером, обраним балансувальником з кластера ECS. Виконується послідовність операцій з перевірки наявності товару в базі даних Aurora, резервування відповідної акцизної марки, створення запису про замовлення, ініціації платіжної транзакції та розміщення повідомлення в черзі SQS для подальшої асинхронної обробки. Користувач отримує підтвердження про створення замовлення.

На четвертому етапі спеціалізований воркер-процес в ECS здійснює моніторинг черги повідомлень та при появі нового завдання виконує послідовність дій з вилучення даних про замовлення, формування XML-документа відповідно до специфікації системи еАкциз, накладання електронного підпису, відправки запиту до API зовнішньої системи та збереження результату в базі даних з архівуванням документа в об'єктному сховищі. У разі недоступності API або отримання помилки реалізовано механізм автоматичного повернення повідомлення в чергу з експоненціальною затримкою між спробами.

П'ятий етап активується після успішної інтеграції з системою еАкциз та

включає оновлення статусу замовлення в базі даних, публікацію події через EventBridge, автоматичне надсилання повідомлення клієнту службою нотифікацій та ініціацію процесу підготовки замовлення до відправки складською системою.

Обґрунтування ключових технічних рішень

Вибір платформи контейнерних обчислень ECS з Fargate замість безсерверної архітектури Lambda обумовлено складністю бізнес-логіки системи, яка включає валідацію даних, операції з базою даних, формування документів та криптографічну обробку. Розбиття такої логіки на численні безсерверні функції значно ускладнило б процеси розробки та діагностики. Контейнерне середовище ECS забезпечує можливість локального відтворення робочого середовища розробниками, тоді як Fargate усуває необхідність управління обчислювальними інстансами шляхом автоматизованого розподілу ресурсів. Платформа Lambda застосована для обробки подій, тригерів на зміни в об'єктному сховищі та запланованих завдань.

Використання Aurora замість стандартної реляційної бази даних RDS PostgreSQL виправдане підвищеними вимогами до відмовостійкості та продуктивності електронної комерції. Додаткові витрати компенсуються автоматичною реплікацією даних між трьома зонами доступності, швидким відновленням після збоїв, покращеною продуктивністю операцій читання та можливістю оперативного додавання реплік для масштабування.

Виділення інтеграції з системою еАкциз в окремий мікросервіс обґрунтовано кількома факторами. Ізоляція збоїв забезпечує локалізацію проблем криптографічного модуля або зовнішнього API без впливу на основну функціональність електронної комерції. Незалежне масштабування дозволяє збільшувати обчислювальні ресурси інтеграційного компонента в періоди підвищеного навантаження. Спрощене тестування досягається через можливість використання симуляторів зовнішнього API без розгортання повного технологічного стеку. Гнучкість оновлень забезпечує швидку адаптацію до змін специфікацій зовнішніх систем шляхом модифікації ізольованого компонента.

Архітектура системи спроектована з урахуванням різних сценаріїв відмов компонентів та інфраструктури. При виході з ладу зони доступності база даних Aurora автоматично перемикається на репліку в альтернативній зоні, тоді як контейнери ECS, розподілені між зонами, залишаються доступними через автоматичне перенаправлення трафіку балансувальником навантаження.

Проблеми доступності зовнішнього API системи еАкциз компенсуються механізмом довготривалого зберігання повідомлень в черзі SQS з автоматичним відновленням обробки після відновлення з'єднання. Інтерфейс адміністрування надає інформацію про стан необроблених операцій для моніторингу та прийняття управлінських рішень.

Захист від перевантаження реалізовано через автоматичне обмеження швидкості запитів на рівні API Gateway, динамічне масштабування кількості контейнерів через механізм ECS Auto Scaling при зростанні навантаження та систему сповіщень CloudWatch для виявлення аномальних патернів використання ресурсів.

Відмовостійкість на рівні програмного коду забезпечується через реалізацію точок перевірки стану для кожного мікросервісу, автоматичне перезапускання контейнерів при невідповідності критеріям працездатності та застосування патерну circuit breaker для запобігання каскадним збоям в розподіленій системі.

2.2. Інфраструктурний рівень (ECS, Fargate, RDS Aurora, S3)

Реалізація інфраструктури передбачає перетворення концептуальної моделі в конкретні ресурси хмарної платформи, які забезпечують безперервну роботу системи та обробку трафіку в промисловому середовищі.

Налаштування мережевої інфраструктури (VPC)

Створення віртуальної приватної хмари виконано з використанням блоку адрес CIDR 10.0.0.0/16 в регіоні eu-central-1, який забезпечує найменшу мережеву затримку для користувачів на території України. Для досягнення високої доступності ресурси розподілено між трьома зонами доступності з організацією публічних підмереж для компонентів з зовнішнім доступом та приватних підмереж

для внутрішніх сервісів бази даних та обчислювальних ресурсів. Маршрутизація трафіку з приватних підмереж здійснюється через окремі NAT Gateway в кожній зоні доступності для забезпечення відмовостійкості.

Групи безпеки організовано за принципом мінімальних привілеїв з дозволом HTTPS-трафіку для балансувальника навантаження, обмеженням доступу до контейнерних сервісів виключно від балансувальника, дозволом з'єднань з базою даних лише від обчислювальних контейнерів та ізоляцією кеш-сервера на рівні мережі. Така конфігурація реалізує багаторівневий захист системи від несанкціонованого доступу.

Elastic Container Service (ECS) та Fargate

Платформа контейнерних обчислень становить основу виконання бізнес-логіки системи. Створено специфікації завдань для різних функціональних компонентів з виділенням обчислювальних ресурсів відповідно до характеру навантаження. Основний REST API отримав один віртуальний процесор та два гігабайти оперативної пам'яті, воркер інтеграції з системою еАкциз виділено половину процесора з одним гігабайтом пам'яті, тоді як для фонових завдань достатньо четверті процесора та половини гігабайта. Кожна специфікація включає змінні середовища, конфіденційні параметри з Secrets Manager та перевірку працездатності через спеціалізований endpoint.

Вибір платформи Fargate обумовлено усуненням необхідності управління обчислювальними інстансами, точним тарифікуванням використаних ресурсів процесора та пам'яті та швидким розгортанням нових екземплярів завдань. Незважаючи на підвищену вартість порівняно з безпосереднім використанням обчислювальних інстансів, платформа виявилась оптимальною для сценаріїв з непередбачуваними піками навантаження.

Application Load Balancer

Балансувальник навантаження додатків виконує прийом HTTPS-запитів, перевірку сертифікатів та розподіл трафіку між екземплярами контейнерних завдань. Створено цільову групу типу IP з періодичною перевіркою працездатності через спеціалізований endpoint з налаштованими інтервалами опитування, часом

очікування відповіді та критеріями визначення працездатного стану. Балансувальник налаштовано на прослуховування захищеного протоколу з сертифікатом від служби управління сертифікатами з реалізацією обмеження швидкості запитів для захисту від зловживань.

RDS Aurora PostgreSQL

Реляційна база даних зберігає критичну інформацію системи, включаючи замовлення, акцизні марки, дані користувачів та історію операцій. Кластер побудовано на основі двигуна Aurora PostgreSQL з виділенням первинного вузла та двох реплік для читання різної потужності. Запити на вибірку даних направляються до реплік, тоді як операції модифікації обробляються первинним вузлом через механізм розділення читання та запису на рівні фреймворку доступу до даних.

Автоматична реплікація між трьома зонами доступності забезпечує відмовостійкість системи. Щоденне автоматичне резервне копіювання з тридцятиденним періодом зберігання та можливістю безперервного відновлення до довільного моменту часу надає гарантії збереження даних. Оптимізація параметрів роботи бази даних, включаючи налаштування максимальної кількості з'єднань, розміру спільних буферів, робочої пам'яті та ефективного розміру кешу, призвела до суттєвого покращення продуктивності складних запитів.

ElastiCache Redis

Служба кешування використовується для підвищення продуктивності системи та управління сесіями користувачів. Кластер організовано з первинним вузлом та двома репліками для читання, розподіленими між зонами доступності з автоматичним перемиканням при відмові первинного вузла.

Застосування кешування охоплює кілька сценаріїв використання. Дані каталогу товарів, які змінюються рідко, кешуються з обмеженим часом життя для зменшення навантаження на реляційну базу. Сесії користувачів зберігаються в кеші для забезпечення можливості обробки запитів без прив'язки до конкретного екземпляру додатку. Механізм обмеження швидкості запитів реалізовано через лічильники в кеші для захисту від зловживань ресурсами системи.

Amazon S3

Об'єктне сховище використовується для зберігання документів різних категорій з організацією окремих контейнерів для XML-документів взаємодії з системою еАкциз з тривалим терміном зберігання відповідно до нормативних вимог, журналів роботи контейнерних сервісів та безсерверних функцій, а також статичних ресурсів веб-додатку з доставкою через мережу CDN.

Автоматизоване управління життєвим циклом об'єктів передбачає послідовне переміщення даних між класами зберігання різної вартості з подальшим видаленням після закінчення необхідного періоду зберігання, що забезпечило суттєве зменшення витрат на зберігання інформації. Заходи безпеки включають версіонування об'єктів, захист від випадкового видалення через багатофакторну автентифікацію, шифрування на стороні сервера з регулярною ротацією ключів та автоматичну обробку подій появи нових файлів через безсерверні функції.

ECR для Docker образів

Приватний реєстр контейнерних образів використовується для зберігання версій компонентів системи з організацією окремих репозиторіїв для кожного сервісу. Автоматичне сканування образів на наявність вразливостей виконується при завантаженні нових версій з блокуванням розгортання критично вразливих образів через конвеєр безперервної інтеграції.

Політики управління життєвим циклом забезпечують збереження обмеженої кількості позначених версій з автоматичним видаленням непозначених образів після визначеного періоду. Стратегія позначення включає останню стабільну версію, специфічні версійні мітки, образи з конкретних комітів для відстеження історії та тестові збірки з запитів на злиття коду.

Інфраструктурний рівень становить фундамент системи, без якого забезпечення масштабованості, високої доступності та продуктивності під навантаженням було б неможливим. Правильне проектування та налаштування мережевої топології, балансування навантаження, управління контейнерами та базами даних забезпечує стабільну роботу всіх вищих рівнів архітектури інтеграційного рішення та створює технічну основу для реалізації бізнес-логіки та інтеграції з зовнішніми системами.

2.3. Інтеграційний рівень та API еАкциз (API Gateway, CloudFront, WAF, IAM, SOAP інтеграція)

Інтеграційний рівень забезпечує взаємодію системи з зовнішніми користувачами та сторонніми сервісами, реалізуючи механізми контролю доступу, захисту від атак та оптимізації доставки контенту.

Amazon API Gateway

Шлюз програмного інтерфейсу становить єдину точку входу для всіх запитів клієнтських додатків до серверної частини системи. Створено регіональний REST API з визначеними ресурсами для операцій з товарами, замовленнями, акцизними марками та перевірки стану системи. Інтеграція кожного методу з балансувальником навантаження виконується через приватне з'єднання VPC Link для забезпечення безпечної комунікації без виходу в публічний інтернет [3].

Автентифікація базується на JWT-токенах з використанням безсерверної функції валідації, яка перевіряє коректність токена при кожному запиті з кешуванням результатів валідації для оптимізації продуктивності. Обмеження швидкості запитів налаштовано на рівні облікового запису, окремих методів з підвищеними обмеженнями для операцій створення даних, а також через систему API-ключів з різними тарифними планами для партнерських інтеграцій.

Журналювання направлено до централізованої служби CloudWatch з налаштуванням метричних фільтрів для виявлення помилок клієнта та сервера, підвищених затримок відповідей та автоматичного сповіщення команди через інтеграцію з месенджером при перевищенні порогових значень.

Amazon CloudFront

Мережа доставки контенту забезпечує кешування та оптимізовану доставку статичних та динамічних ресурсів через географічно розподілені точки присутності. Організовано два окремих дистрибутиви для статичного фронтенду з джерелом в об'єктному сховищі та для програмного інтерфейсу з джерелом в API Gateway з селективним кешуванням операцій читання.

Додатковий рівень кешування Origin Shield розгорнуто в регіоні для

зменшення навантаження на серверну частину через консолідацію запитів від множини точок присутності. Поведінка кешування налаштована диференційовано для статичних ресурсів з тривалим терміном зберігання та стисненням, запитів до каталогу товарів з короткостроковим кешуванням та передачею заголовків автентифікації, а також динамічного контенту без кешування.

Інвалідація кешу реалізована через event-driven архітектуру з публікацією подій про оновлення даних, обробкою подій безсерверними функціями та автоматичним викликом API інвалідації для відповідних ресурсів. Географічні обмеження доступу налаштовано відповідно до нормативних вимог продажу алкогольної продукції з блокуванням запитів з неавторизованих регіонів.

AWS WAF

Брандмауер веб-додатків становить першу лінію захисту системи від шкідливого трафіку через інтеграцію з мережею доставки контенту. Застосовано керовані групи правил для захисту від поширених атак, включаючи ін'єкції SQL-коду, міжсайтовий скриптинг, включення локальних файлів та віддалене виконання коду, блокування відомих шкідливих патернів запитів та фільтрацію трафіку з IP-адрес з поганою репутацією.

Користувацькі правила включають обмеження кількості запитів з одного джерела з тимчасовим блокуванням при перевищенні порогу, географічне блокування регіонів з високим рівнем автоматизованого трафіку та фільтрацію запитів з характерними ідентифікаторами інструментів сканування вразливостей. Журнали брандмауера зберігаються в об'єктному сховищі з можливістю аналізу через службу запитів для виявлення патернів атак та оптимізації правил безпеки.

AWS IAM

Система управління ідентифікацією та доступом становить основу забезпечення безпеки в хмарному середовищі. Створено сервісні ролі для різних компонентів системи з чітко визначеними дозволами. Роль виконання контейнерних завдань надає права на завантаження образів з реєстру, запис журналів та читання конфіденційних даних. Роль самих завдань включає операції з об'єктним сховищем, чергами повідомлень, шиною подій та інвалідацією кешу

мережі доставки контенту. Ролі безсерверних функцій налаштовано з базовими дозволами та специфічними правами відповідно до функціонального призначення.

Застосовано принцип мінімальних привілеїв з явним вказуванням конкретних операцій над конкретними ресурсами замість широких дозволів на всі дії. Розділено права доступу для людей та сервісів з організацією окремих облікових записів та груп для співробітників через систему єдиного входу. Група розробників отримала доступ до журналів та можливість запуску завдань в непромислових середовищах, тоді як DevOps-команда має повний доступ до виробничого середовища з обов'язковою багатофакторною автентифікацією. Фінансовий відділ має права лише на перегляд інформації про витрати та звітів.
Багатофакторна автентифікація обов'язкова для всіх користувачів без винятків. Міжоблікові доступи організовано через окремі облікові записи для різних середовищ з використанням механізму прийняття ролей для діагностики проблем у виробничому середовищі.

AWS Secrets Manager

Служба управління конфіденційними даними зберігає критичну інформацію безпеки, включаючи паролі баз даних, токени автентифікації служб кешування, секретні ключі JWT та сертифікати кваліфікованого електронного підпису. Автоматична ротація паролів бази даних виконується через безсерверні функції з регулярним інтервалом, тоді як додатки періодично оновлюють отримані значення для запобігання використанню застарілих креденшелів. Версіонування всіх змін забезпечує можливість швидкого відкату до попередніх значень у разі виникнення проблем.

Інтеграція з API еАкциз

Інтеграція з державною системою еАкциз - найскладніша частина проекту. Помилки означають штрафи та юридичні проблеми.

Аналіз API еАкциз

API працює по SOAP protocol (не REST).
Endpoint: https://cabinet.tax.gov.ua/excise/api/v1/ExciseService
Кожен запит:

1. XML документ згідно XSD схеми
2. Підписаний КЕП
3. Відправлений як SOAP envelope
4. Супроводжується SSL сертифікатом

Типи операцій

- RegisterExciseStamps - реєстрація нових марок
- MoveExciseStamps - переміщення між складами
- SaleExciseStamps - продаж товару
- ReturnExciseStamps - повернення
- GetStampStatus - перевірка статусу
- GetOperationsHistory - історія операцій

Інтеграція з API еАкциз

Взаємодія з державною системою обліку акцизних марок становить критично важливий компонент архітектури з високими вимогами до надійності та відповідності нормативним вимогам. Аналіз зовнішнього API виявив використання протоколу SOAP з вимогами до підпису кожного запиту кваліфікованим електронним підписом та супроводження SSL-сертифікатом. Підтримуються операції реєстрації нових марок, переміщення між складськими приміщеннями, фіксації продажу товарів, обробки повернень, перевірки статусу марок та отримання історії операцій. Зовнішня система застосовує жорсткі обмеження швидкості запитів з блокуванням при перевищенні встановлених лімітів, що критично вплинуло на архітектурні рішення щодо батчування та оптимізації запитів.

Архітектура інтеграційного сервісу

Функціональність взаємодії з зовнішньою системою виділено в окремий мікросервіс на платформі .NET в контейнерному середовищі. Асинхронна обробка через черги повідомлень дозволяє додатку електронної комерції створювати замовлення та негайно повертати підтвердження користувачу без очікування відповіді від зовнішньої системи, тоді як інтеграційний сервіс забирає завдання з черги та виконує обробку в фоновому режимі. Такий підхід забезпечує швидку

відповідь користувачам, можливість батчування операцій для оптимізації кількості запитів, автоматичні повторні спроби при помилках та природний захист від обмежень швидкості запитів.

Робота з кваліфікованим електронним підписом

Електронний підпис базується на українському стандарті еліптичної криптографії, відмінному від широко використовуваних RSA-алгоритмів. Використовується спеціалізована бібліотека з нативними залежностями через .NET-обгортку для виконання криптографічних операцій. Ініціалізація криптопровайдера включає завантаження сертифікатів центрів сертифікації, отримання приватного ключа з служби управління конфіденційними даними та налаштування параметрів роботи. Виклик операції підпису здійснюється над бінарним представленням XML-документа з поверненням результату в кодованому форматі. Робота з бібліотекою ускладнена необхідністю створення спеціалізованого образу контейнера з нативними залежностями, проблемами багатопотоковості, витоками пам'яті та обмеженою документацією.

Формування документів

Побудова XML-документів виконується програмно з дотриманням специфікації зовнішньої системи. Документ включає тип операції, дату формування, інформацію про організацію з ідентифікаційним кодом та назвою, а також перелік акцизних марок з кодами та найменуваннями продукції. Сформовані документи інкапсулюються в SOAP-конверти для передачі через веб-сервіси.

Логіка повторних спроб

Нестабільність зовнішнього API компенсується реалізацією механізму повторних спроб з експоненціальною затримкою для мережевих помилок, таймаутів та серверних помилок. Застосовується стратегія поступового збільшення інтервалів між спробами для зменшення навантаження на проблемний сервіс. Операції, що не вдалися після всіх спроб, направляються до черги невдалих повідомлень для ручного аналізу та повторної обробки після усунення причин помилок адміністратором.

Батчування операцій

Накопичення операцій протягом визначеного часового інтервалу або до досягнення максимальної кількості марок дозволяє об'єднувати множину операцій в один запит до зовнішньої системи. Такий підхід призвів до суттєвого зменшення кількості викликів API при збереженні функціональності системи.

Моніторинг інтеграції

Користувацькі метрики включають кількість успішних та невдалих операцій, середню тривалість виконання запитів, розміри черг повідомлень та кількість перевищень обмежень швидкості. Налаштовано автоматичні сповіщення при підвищенні частки невдалих операцій, накопиченні повідомлень в черзі невдалих запитів та перевищенні порогових значень тривалості виконання на високих перцентилях розподілу.

Інтеграція з зовнішньою державною системою виявилася найскладнішим технічним компонентом проекту, проте асинхронна архітектура з використанням черг повідомлень, надійна логіка повторних спроб та батчування операцій дозволили побудувати стабільне рішення.

2.4. DevOps практики та автоматизація (CI/CD, IaC, моніторинг)

Забезпечення надійного функціонування системи потребує автоматизації процесів доставки коду до виробничого середовища та організації безперервного моніторингу стану компонентів.

Infrastructure as Code (IaC)

Вся інфраструктура системи описується програмним кодом для забезпечення відтворюваності середовищ та версійного контролю змін. Розділення відповідальностей передбачає використання різних інструментів для базової інфраструктури та специфічних для додатків ресурсів.

Розділення відповідальностей

Команда DevOps відповідає за core інфраструктуру з використанням Terraform для управління базовими компонентами, включаючи віртуальну приватну хмару з мережевою конфігурацією, бази даних та сервіси кешування,

об'єктні сховища, ролі управління доступом на інфраструктурному рівні, мережу доставки контенту та DNS-маршрутизацію. Вибір цього інструменту обумовлено декларативним підходом до опису інфраструктури, централізованим управлінням станом, підтримкою множини середовищ через робочі простори та можливістю створення багаторазово використовуваних модулів [14].

Розробники застосовують AWS CDK на платформі .NET для управління ресурсами, тісно інтегрованими з кодом додатку, включаючи безсерверні функції, черги повідомлень та правила обробки подій. Перевагою цього підходу є використання єдиної мови програмування з основним кодом, типобезпека на етапі компіляції, можливість застосування програмної логіки в описі інфраструктури та близькість до коду додатку.

Файл стану Terraform зберігається в об'єктному сховищі з використанням служби баз даних для блокування одночасного доступу, шифруванням та версіонуванням змін. Робочі простори дозволяють використовувати єдину кодову базу для множини середовищ з різними конфігураційними файлами.

CI/CD Pipeline

Автоматизація процесів від коміту коду до розгортання в виробничому середовищі реалізована через GitLab CI/CD з визначенням послідовних етапів обробки. Конвеєр включає збірку додатку з відновленням залежностей та компіляцією, тестування через виконання модульних та інтеграційних тестів з залученням тимчасових екземплярів баз даних, аналіз безпеки з перевіркою якості коду та сканування вразливостей, побудову образів контейнерів та розгортання в середовища staging автоматично і production з ручним підтвердженням [15].

Аналіз якості коду через SonarQube включає виявлення архітектурних недоліків та технічного боргу, потенційних помилок та вразливостей безпеки, вимірювання покриття тестами з мінімальним порогом та ідентифікацію критичних точок безпеки. Невідповідність критеріям якості блокує можливість злиття змін до основної гілки коду.

Сканування контейнерів на вразливості виконується автоматично для виявлення проблем в базових образах, середовищі виконання та пакетах

залежностей. Виявлення критичних вразливостей призводить до блокування конвеєра з вимогою оновлення проблемних компонентів.

Приватний реєстр образів контейнерів забезпечує зберігання версій додатків з автоматичним скануванням безпеки, політиками життєвого циклу для видалення застарілих версій та реплікацією для відновлення після катастроф. Стратегія розгортання Blue/Green через контейнерний сервіс передбачає запуск нових екземплярів паралельно з існуючими, поступове перемикання трафіку через балансувальник навантаження та швидке відкочування до попередньої версії при виявленні проблем, забезпечуючи нульовий час простою.

Стратегія відкочування включає автоматичне повернення до попередньої версії при спрацюванні алармів моніторингу про підвищення частки серверних помилок або ручне відкочування через систему безперервної інтеграції з пошуком останнього успішного конвеєра та розгортання відповідної версії через оновлення конфігурації сервісу.

Безпека в хмарному середовищі

Комплексний підхід до безпеки включає використання всього спектру засобів захисту хмарної платформи. Брандмауер веб-додатків забезпечує захист від ін'єкцій та міжсайтового скриптингу, обмеження швидкості запитів та географічне блокування. Захист від DDoS-атак реалізовано через базовий рівень автоматично для всіх ресурсів та розширений рівень для виробничого середовища з доступом до цілодобової команди реагування.

Служба управління ключами шифрування забезпечує захист даних у базах даних, об'єктних сховищах та блокових пристроях зберігання з використанням ключів, керованих клієнтом, та автоматичною ротацією з регулярним інтервалом. Служба управління конфіденційними даними зберігає паролі, API-ключі та сертифікати з автоматичною ротацією паролів баз даних через безсерверні функції.

Безперервний моніторинг конфігурації перевіряє відповідність політикам безпеки, включаючи шифрування сховищ, розташування баз даних в ізольованих мережах та увімкнення багатофакторної автентифікації для привілейованих облікових записів. Центральна панель безпеки агрегує результати від множини

служб захисту та перевіряє відповідність галузевим стандартам безпеки. Аналіз на основі машинного навчання виявляє аномалії через обробку журналів аудиту, мережевих потоків та DNS-запитів для виявлення компрометованих облікових даних, незаконного майнінгу криптовалют, backdoor-доступу та підвищення привілеїв. Аналізатор доступу перевіряє політики управління доступом на надмірні дозволи, ризики міжоблікового доступу та невикористовувані ролі.

Моніторинг та Observability

Всі логи з ECS, Lambda, API Gateway в CloudWatch. Аналіз через Logs Insights (рис 2.1):

```
# Топ 10 найповільніших endpoints
fields @timestamp, request.path, request.duration
| filter request.duration > 1000
| sort request.duration desc
| limit 10
```

Рис.2.1 Запит через Logs Insights

Детальний моніторинг системи забезпечує можливість виявлення та діагностики проблем на ранніх стадіях. Всі журнали з контейнерних сервісів, безсерверних функцій та API-шлюзу централізовано зберігаються з можливістю аналізу через мову запитів для виявлення найповільніших endpoint, аномалій продуктивності та патернів помилок. Структуроване журналювання в форматі JSON з контекстом запиту забезпечує кореляцію подій між різними компонентами системи.

Користувацькі метрики включають кількість замовлень за часовими інтервалами, швидкість успішних інтеграцій з зовнішньою системою, розміри черг повідомлень, використання ресурсів процесора та пам'яті контейнерів, частоту помилок безсерверних функцій та тривалість відповідей баз даних. Організовано окремі панелі для різних аудиторій з метриками операційної інфраструктури для технічних команд, бізнес-показниками для управлінського персоналу та метриками інтеграцій для моніторингу взаємодії з зовнішніми системами.

Автоматичні сповіщення налаштовано для критичних метрик з диференціацією каналів доставки залежно від рівня критичності. Підвищене використання процесорних ресурсів призводить до автоматичного масштабування, критичні помилки активують процес управління інцидентами з негайним сповіщенням чергового персоналу, накопичення невдалих операцій вимагає ручного перегляду проблем.

Синтетичний моніторинг реалізовано через безперервне виконання тестових сценаріїв користувацьких потоків, включаючи доступність сайту, пошук товарів, операції з кошиком та процес оформлення замовлення без реальних платежів.

- Пошук товару
- Додавання в кошик
- Checkout flow (без реального payment)

Якщо canary падає - система недоступна, алерт негайно.

Log Aggregation ma Analysis

CloudWatch Logs експортуємо в S3 для довгострокового зберігання. Використовуємо Athena для аналізу (рис 2.2):

```
SELECT status_code, COUNT(*) as count
FROM alb_logs
WHERE day = '2024-11-26'
GROUP BY status_code
ORDER BY count DESC;
```

Рис.2.2 Запит з використанням Athena

Журнали експортуються до довгострокового зберігання з можливістю аналізу через службу запитів для виявлення патернів трафіку, частоти кодів відповідей та інших метрик. Потокова обробка журналів в реальному часі забезпечує інтеграцію з аналітичними системами.

Моніторинг витрат включає встановлення денних та місячних бюджетів з автоматичним сповіщенням при прогнозованому перевищенні порогових значень. Аналіз структури витрат за категоріями сервісів виявив розподіл між

обчислювальними ресурсами, базами даних, передачею даних, об'єктним зберіганням та іншими сервісами. Оптимізація здійснена через резервування ресурсів баз даних, плани заощадження для контейнерних обчислень та інтелектуальне управління рівнями зберігання об'єктів.

Управління інцидентами

Команда DevOps організована в систему чергувань з цілодобовою готовністю через спеціалізовану платформу управління інцидентами. Для кожного типу інциденту розроблено процедури реагування з переліком кроків діагностики та відновлення, включаючи перевірку стану контейнерів, балансувальника та баз даних при недоступності API, аналіз тривалих запитів та пулів з'єднань при уповільненні баз даних, перевірку черг, обмежень швидкості та сертифікатів при проблемах інтеграції з зовнішніми системами.

Після кожного значного інциденту виконується ретроспективний аналіз з хронологією подій, визначенням первопричини, оцінкою впливу на користувачів, описом дій з відновлення та формуванням переліку коригувальних заходів для запобігання повторенню. Процес аналізу зосереджений на удосконаленні процесів без звинувачення окремих осіб.

Практики DevOps та автоматизація забезпечують можливість швидкої розробки функціональності, впевненого розгортання в виробничому середовищі та підтримки стабільності системи. Без автоматизації доставки коду, інфраструктури як коду та належного моніторингу система не могла б забезпечити необхідний рівень надійності.

3 РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1. Впровадження хмарної інфраструктури на AWS

Було проведено поетапне впровадження хмарної інфраструктури на платформі Amazon Web Services протягом восьми тижнів, від початкового розгортання мережевих компонентів до повноцінного production запуску. Процес впровадження розділено на п'ять основних етапів для мінімізації ризиків та забезпечення можливості виявлення проблем на ранніх стадіях. Така стратегія інкрементального розгортання дозволила уникнути ситуації одномоментного запуску усіх компонентів, що могло б призвести до складнощів у діагностиці проблем та підвищених ризиків для бізнес-процесів [13].

На початковому етапі, який тривав перші два тижні, було створено базову мережеву інфраструктуру. Засобами інструменту Terraform розгорнуто віртуальну приватну хмару VPC разом із шістьма підмережами, де три підмережі налаштовані як публічні для розміщення балансувальників навантаження, а три приватні підмережі призначені для розміщення контейнерів додатків та баз даних. Процес створення інфраструктури за допомогою команд ініціалізації та застосування конфігурації Terraform зайняв приблизно п'ять хвилин, протягом яких автоматично створено Internet Gateway для доступу публічних ресурсів до мережі Інтернет, три NAT Gateway для забезпечення вихідного трафіку з приватних підмереж, а також таблиці маршрутизації для керування потоками мережевого трафіку.

Для кожного сервісу платформи створено окремі IAM ролі з мінімальним набором дозволів, необхідних для виконання специфічних функцій сервісів ECS, Lambda та RDS.

Другий етап впровадження, що охоплював третій тиждень процесу, присвячено розгортанню компонентів зберігання даних. Створено кластер Aurora PostgreSQL версії 17.4 з типом інстансу db.t4.large для середовища розробки, де налаштовано автоматичне резервне копіювання з періодом утримання семи днів та вікном виконання резервних копій з третьої до четвертої години ночі для

мінімізації впливу на робоче навантаження. Процес створення кластера бази даних зайняв десять хвилин, після чого встановлено з'єднання та створено схему бази даних з таблицями для продуктів та акцизних марок, де таблиця продуктів містить ідентифікатор, назву, ціну та прапорець необхідності акцизної марки, тоді як таблиця акцизних марок зберігає код марки, зв'язок з продуктом, статус марки та ідентифікатор складу. Для оптимізації продуктивності створено індекси на полях статусу марки та ідентифікатора складу, що прискорює виконання типових запитів до системи.

Паралельно з базою даних розгорнуто кластер ElastiCache Redis версії 7.0 для кешування даних та зберігання сесій користувачів, де процес створення зайняв п'ять хвилин з автоматичним налаштуванням двох вузлів кешу з увімкненим автоматичним перемиканням та розподілом по декількох зонах доступності. Перевірка підключення через клієнт командного рядка підтвердила коректну роботу кластеру. Під час налаштування виявлено проблему з жорстким кодуванням рядків підключення в змінних середовища визначень завдань ECS, що створювало б складнощі при необхідності зміни кінцевих точок або паролів, оскільки вимагало б перебудови контейнерів. Прийнято рішення перенести всі секретні дані до сервісу Secrets Manager, звідки додаток завантажує необхідні параметри під час запуску через асинхронний виклик до менеджера секретів з передачею ідентифікатора секрету для отримання паролю бази даних.

Третій етап, що тривав четвертий тиждень проекту, виявився найскладнішим через необхідність налаштування оркестрації контейнерів ECS, побудови образів Docker та розгортання додатків. Спочатку локально створено образ Docker для сервісу API на основі базового образу ASP.NET 6.0 з використанням багатоетапної збірки, де етап build виконує відновлення залежностей та компіляцію проекту, етап publish готує фінальні артефакти, а етап final копіює скомпільовані файли до робочого образу. Створений образ було позначено тегом версії 0.1.0 та завантажено до реєстру Elastic Container Registry після автентифікації через команду отримання пароля для входу. Для розгортання контейнерів створено визначення завдання ECS через Terraform, хоча початкове налаштування виконувалося вручну через веб-

консоль для детального розуміння усіх параметрів конфігурації.

Перший запуск завдання завершився невдало через неможливість контейнера встановити з'єднання з базою даних, що виявилося наслідком відсутності групи безпеки завдань ECS у списку дозволених джерел для групи безпеки RDS. Після виправлення конфігурації груп безпеки завдання успішно запустилося, проте виникла нова проблема з постійними невдачами перевірок стану здоров'я, що призводило до циклічних перезапусків контейнера. Діагностика показала, що кінцева точка перевірки стану повертала код відповіді 503 через неможливість додатка підключитися до Redis внаслідок неправильно вказаної кінцевої точки сервісу. Виправлення налаштувань підключення до Redis дозволило перевіркам стану проходити успішно, після чого завдання досягло стабільного стану працездатності.

Для розподілу навантаження створено Application Load Balancer з відповідною цільовою групою, налаштованою на порт 8080 з протоколом HTTP та типом цілей на основі IP-адрес. Конфігурація перевірок здоров'я встановлена з шляхом до кінцевої точки стану, порогом здорових станів у два успішні перевірки, порогом нездорових станів у три невдалі перевірки, тайм-аутом п'ять секунд та інтервалом між перевірками тридцять секунд. Початковий запит до балансувальника повернув помилку 502 Bad Gateway, оскільки завдання ECS зареєструвалося у цільовій групі, але не встигло пройти перевірку здоров'я. Після очікування двох хвилин завдання досягло здорового стану, балансувальник почав направляти трафік, і тестовий запит до кінцевої точки стану здоров'я повернув успішну відповідь зі статусом працездатності системи, включаючи стан підключень до бази даних та Redis.

Четвертий етап впровадження охоплював п'ятий тиждень проекту та присвячувався налаштуванню API Gateway та мережі доставки контенту CloudFront. Створено REST API в сервісі API Gateway з під'єднанням до балансувальника навантаження через VPC Link, що дозволяє безпечно маршрутизувати запити до приватних ресурсів у віртуальній приватній хмарі. Налаштування ресурсів та методів API спочатку виконувалося вручну, з

подальшою автоматизацією через специфікацію OpenAPI. Під час інтеграції фронтенд-додатка виникла проблема з політиками спільного використання ресурсів різних джерел CORS, де браузер блокував запити до API Gateway через відсутність необхідних заголовків відповідей. Налаштування політики CORS з дозволом доступу для домену фронтенд-додатка та специфікацією дозволених методів та заголовків вирішило проблему міжсайтових запитів.

Для глобального розподілу контенту та зменшення затримок створено дистрибуцію CloudFront з джерелом на API Gateway, де налаштовано власний домен та сертифікат SSL для безпечних з'єднань. Конфігурація дистрибуції включає підтримку всіх HTTP методів для операцій читання та запису, кешування тільки для методів отримання даних, переадресацію заголовків авторизації до джерела, політику протоколу з автоматичною переадресацією на захищене з'єднання HTTPS, географічні обмеження з дозволом доступу тільки з країн України, Німеччини, Франції та Польщі, а також використання сертифіката з AWS Certificate Manager з підтримкою Server Name Indication. Процес створення дистрибуції CloudFront зайняв п'ятнадцять хвилин через необхідність поширення конфігурації на всі крайові локації глобальної мережі доставки контенту.

П'ятий етап впровадження, що тривав шостий тиждень, присвячено налаштуванню систем моніторингу та сповіщень. Створено інформаційну панель CloudWatch з ключовими метриками, що відображають використання процесора та пам'яті контейнерами ECS, кількість запущених завдань, кількість запитів до балансувальника навантаження, час відповіді цільових ресурсів, кількість серверних помилок HTTP 5XX, використання процесора базою даних RDS, кількість активних з'єднань до бази даних, затримки операцій читання та запису, а також метрики Redis щодо влучень кешу, промахів кешу та витіснень записів. Налаштовано сигналізацію CloudWatch для критичних подій з використанням порогових значень та періодів оцінки, зокрема сигнал про серверні помилки API активується при виявленні більше десяти помилок типу 5XX протягом двох послідовних п'ятихвилинних періодів з відправкою повідомлень до теми SNS для сповіщення відповідальних осіб.

Після досягнення стабільної роботи середовища розробки протягом тижня розпочато створення середовищ staging та production. Середовище staging створено як копія production з меншими обчислювальними ресурсами, де база даних Aurora використовує інстанси типу db.t3.large замість db.r6g.xlarge для production, кеш Redis налаштовано на інстанси cache.t3.small замість cache.r6g.large, а служба ECS запускає мінімум дві задачі замість трьох. Використання робочих просторів Terraform дозволило створити всю інфраструктуру staging протягом двадцяти хвилин з розгортанням того самого образу Docker, що використовувався в середовищі розробки.

Досвід впровадження хмарної інфраструктури дозволив сформулювати низку важливих висновків щодо ефективних практик розгортання. По-перше, поетапний підхід з послідовним створенням середовищ розробки, staging та production дозволяє виявляти та вирішувати проблеми на ранніх стадіях без впливу на критичні бізнес-процеси. По-друге, повна автоматизація створення інфраструктури через інструменти Infrastructure as Code є обов'язковою для забезпечення повторюваності та відсутності помилок ручного налаштування. По-третє, впровадження систем моніторингу та сповіщень з першого дня експлуатації критично важливе для своєчасного виявлення та реагування на проблеми продуктивності та доступності. По-четверте, правильне налаштування пулів з'єднань до бази даних є критично важливим для забезпечення стабільної продуктивності під навантаженням. По-п'яте, встановлення терміну життя для всіх записів кешу обов'язкове для запобігання вичерпанню пам'яті Redis. По-шосте, систематичний моніторинг витрат через AWS Cost Explorer з першого дня допомагає уникнути несподіваних перевитрат бюджету. По-сьоме, наявність детального плану відкату для кожного розгортання забезпечує можливість швидкого відновлення працездатності у разі виявлення критичних проблем.

Загальний процес впровадження хмарної інфраструктури від першого виконання команд Terraform до досягнення стабільної роботи production середовища зайняв вісім тижнів. Хоча цей термін може здаватися значним, інвестиція часу у правильне початкове налаштування всіх компонентів,

налагодження процесів моніторингу та створення автоматизованих процедур розгортання виправдала себе через відсутність критичних інцидентів та необхідності термінових виправлень у наступні місяці експлуатації системи.

3.2. Реалізація інтеграції з системою еАкциз

Реалізація інтеграції з державною системою еАкциз становила найбільш трудомісткий та технічно складний компонент проекту через специфіку роботи з державним API, необхідність використання кваліфікованого електронного підпису та відсутність документованих практик розробки. На відміну від хмарної інфраструктури AWS, що характеризується передбачуваністю процесів та вичерпною документацією, інтеграція з системою еАкциз вимагала значних зусиль на дослідження недокументованої поведінки системи та адаптацію до специфічних вимог державного API.

Початковий етап інтеграції передбачав виконання низки адміністративних процедур для отримання доступу до API системи еАкциз. Організація повинна бути зареєстрована як юридична особа з ліцензією на продаж підакцизних товарів, що становить обов'язкову передумову для доступу до системи. Процес включав отримання кваліфікованого електронного підпису директора підприємства через акредитований центр сертифікації ключів, подання офіційної заявки до Державної податкової служби на надання технічного доступу до системи електронного обліку акцизних марок, а також отримання технічної документації та XSD схем для валідації структури XML документів. Загальна тривалість адміністративних процедур склала три тижні, протягом яких виявлено проблему застарілості наданої документації, датованої дві тисячі вісімнадцятим роком, де значна частина прикладів коду виявилася нефункціональною у поточній версії API.

Підготовка тестового середовища виявила фундаментальну відмінність системи еАкциз від типових промислових API, а саме відсутність окремого тестового середовища типу sandbox. Всі операції тестування здійснюються у production середовищі з використанням реальних даних, що значно підвищує

ризики помилкових операцій під час розробки. Державна податкова служба надала обмежений набір з десяти тестових акцизних марок для верифікації функціональності інтеграції, де кожна марка може бути використана лише одноразово для реєстрації операції, після чого стає недоступною для повторного тестування. Така обмеженість тестових ресурсів вимагала особливо ретельного планування тестових сценаріїв та максимальної обережності при виконанні кожної тестової операції для збереження обмеженого набору тестових марок на весь період розробки та налагодження інтеграції.

Створено спеціалізований Docker образ (рис 3.1) з інтегрованими криптографічними бібліотеками, що включає встановлення системних залежностей OpenSSL, libcurl, libxml2 та допоміжних утиліт, завантаження та розпакування Linux версії бібліотеки IIT End User, копіювання нативних бібліотек до системної директорії та розміщення кореневих сертифікатів центрів сертифікації ключів у відповідних директоріях контейнера [16].

```
FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS base

RUN apt-get update && apt-get install -y \
    libssl1.1 \
    libcurl4 \
    libxml2 \
    unzip \
    wget \
    && rm -rf /var/lib/apt/lists/*

WORKDIR /tmp
RUN wget
https://iit.com.ua/download/productfiles/EUSignCP.zip \
    && unzip EUSignCP.zip -d /opt/iit/ \
    && chmod +x /opt/iit/lib/*.so

COPY lib/linux/x86_64/*.so /usr/lib/
COPY certificates/root/* /opt/certificates/root/
COPY certificates/ca/* /opt/certificates/ca/

WORKDIR /app
COPY --from=publish /app/publish .

ENV EU_SIGN_CP_LIB_PATH=/usr/lib/libEUSignCP.so
ENV EU_SIGN_CP_CERTIFICATES_PATH=/opt/certificates

ENTRYPOINT ["dotnet", "ExciseIntegration.dll"]
```

Рис.3.1 Докерфайл образу

Під час тривалого тестування протягом тижня виявлено критичну проблему

витоку пам'яті у бібліотеці IIT, де контейнер демонстрував постійне зростання споживання оперативної пам'яті від початкових двохсот п'ятдесяти шести мегабайт при старті до півтора гігабайта після доби безперервної роботи. Профілювання додатка показало, що бібліотека IIT End User не коректно звільняє виділену пам'ять після виконання операцій цифрового підпису документів, що є відомою проблемою згідно з обговореннями на спеціалізованих форумах розробників. Для вирішення проблеми прийнято прагматичне рішення про запровадження автоматичного перезапуску контейнера кожні двадцять чотири години через механізм запланованих завдань ECS, що хоча і не є елегантним архітектурним рішенням, проте забезпечує стабільну роботу системи без накопичення витоків пам'яті.

Для вирішення проблеми прийнято прагматичне рішення про запровадження автоматичного перезапуску контейнера кожні двадцять чотири години через механізм запланованих завдань ECS, що хоча і не є елегантним архітектурним рішенням, проте забезпечує стабільну роботу системи без накопичення витоків пам'яті.

Реалізація сервісу цифрового підпису документів вимагала особливої уваги до потокобезпечності операцій, оскільки бібліотека IIT End User не підтримує одночасне виконання операцій у різних потоках. Сервіс криптографії включає механізм ініціалізації з використанням семафору для забезпечення атомарності процесу початкового налаштування, завантаження сертифікатів центрів сертифікації ключів з вказаних у змінних оточення директорій, отримання паролю приватного ключа з AWS Secrets Manager для забезпечення безпечного зберігання секретних даних, завантаження файлу приватного ключа директора з захищеного сховища S3 та синхронізованого виконання операцій підпису через блокування критичних секцій коду. Метод підпису XML документа приймає на вхід текстове представлення документа у форматі UTF-8, перетворює його у масив байтів, виконує операцію цифрового підпису з використанням зовнішнього формату підпису та повертає результат у вигляді Base64-кодованого рядка для подальшої інтеграції у SOAP запит до API системи еАкциз.

Компонент формування XML документів реалізовано через систему

будівельників (рис 3.2) для кожного типу операції з акцизними марками, де будівельник документа продажу створює структуровані XML документи з використанням просторів імен системи еАкциз, включаючи заголовок документа з типом операції, номером та датою документа у форматі ISO 8601, ідентифікаційним кодом підприємства, основну частину з інформацією про організацію, список акцизних марок з кодами, найменуваннями продукції, кодами УКТЗЕД, кількістю та цінами, а також підсумкову секцію з загальною кількістю марок та сумою операції.

```
public class SaleDocumentBuilder
{
    public string BuildSaleDocument(SaleOperationRequest request)
    {
        var doc = new XDocument(
            new XDeclaration("1.0", "UTF-8", null),
            new XElement(ns + "ExciseDocument",
                new XAttribute(XNamespace.Xmlns + "xsi", xsiNs),

                new XElement(ns + "Header",
                    new XElement(ns + "DocumentType", "SALE"),
                    new XElement(ns + "DocumentNumber", request.DocumentNumber),
                    new XElement(ns + "DocumentDate", DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ss")),
                    new XElement(ns + "EDRPOU", _config.CompanyEDRPOU)
                ),

                new XElement(ns + "Body",
                    new XElement(ns + "Organization",
                        new XElement(ns + "Name", _config.CompanyName),
                        new XElement(ns + "Address", _config.CompanyAddress)
                    ),

                    new XElement(ns + "Stamps",
                        request.Stamps.Select(stamp =>
                            new XElement(ns + "Stamp",
                                new XElement(ns + "Code", stamp.Code),
                                new XElement(ns + "ProductName", stamp.ProductName),
                                new XElement(ns + "ProductCode", stamp.UKTZED),
                                new XElement(ns + "Quantity", stamp.Quantity),
                                new XElement(ns + "Price", stamp.Price.ToString("F2"))
                            )
                        )
                    ),

                    new XElement(ns + "Total",
                        new XElement(ns + "StampsCount", request.Stamps.Count),
                        new XElement(ns + "TotalAmount", request.TotalAmount.ToString("F2"))
                    )
                )
            )
        );

        return doc.ToString(SaveOptions.DisableFormatting);
    }
}
```

Рис.3.2 Код класу .NET

Критичною проблемою під час початкових спроб відправки документів стала

валідація структури XML, де перші кілька запитів завершувалися помилкою від системи еАкциз з повідомленням про невалідну структуру без конкретизації місця або характеру помилки. Для вирішення проблеми додано обов'язкову валідацію згенерованих XML документів проти XSD схем перед відправкою до API, що дозволило виявляти структурні помилки на етапі локальної обробки. Валідатор завантажує набір XSD схем у колекцію схем, парсить XML документ та виконує валідацію з накопиченням усіх виявлених помилок для детального аналізу. Після впровадження локальної валідації виявлено декілька системних проблем у генерації документів, зокрема неправильне використання просторів імен XML, відсутність обов'язкових полів згідно зі схемою та неправильне форматування дат, що вимагало ISO 8601 формату замість локалізованих представлень.

Після впровадження локальної валідації виявлено декілька системних проблем у генерації документів, зокрема неправильне використання просторів імен XML, відсутність обов'язкових полів згідно зі схемою та неправильне форматування дат, що вимагало ISO 8601 формату замість локалізованих представлень.

Клієнт для взаємодії з SOAP API системи еАкциз реалізовано з використанням HTTP клієнта для відправки SOAP конвертів, інтеграцією з сервісом криптографії для цифрового підпису документів, вбудованим логуванням усіх етапів обробки запитів та вимірюванням часу виконання операцій. Метод відправки операції приймає тип операції та XML документ, виконує цифровий підпис документа через сервіс криптографії, формує SOAP конверт з підписаним документом, встановлює необхідні HTTP заголовки включаючи SOAPAction, відправляє запит до API з вимірюванням часу виконання, зберігає запит та відповідь для цілей аудиту з фіксацією типу операції, тексту запиту та відповіді, HTTP статус-коду, тривалості операції та часової мітки, логує результати виконання з деталізацією параметрів та тривалості, обробляє помилки мережевого рівня та тайм-аути з відповідним логуванням та парсить відповідь від API для отримання структурованого результату операції.

Для підвищення надійності системи та зменшення навантаження на основний

потік обробки замовлень реалізовано асинхронну архітектуру обробки операцій з акцизними марками через чергу повідомлень Amazon SQS. Замість синхронного виклику API системи еАкциз безпосередньо при створенні замовлення, повідомлення про необхідність реєстрації операції додається до черги для подальшої асинхронної обробки. Процесор черги отримує повідомлення про операцію з указанням типу операції та ідентифікатора замовлення, логує початок обробки операції, формує XML документ згідно з параметрами операції, виконує валідацію документа проти XSD схем з накопиченням помилок, відправляє документ до API еАкциз з використанням політики повторних спроб, при успішному виконанні оновлює статус операції у базі даних з записом реєстраційного номера, публікує подію про завершення операції для інформування інших компонентів системи, при виникненні помилок після п'яти невдалих спроб відправляє повідомлення до черги невдалих повідомлень DLQ та надсилає критичне сповіщення команді підтримки.

Для підвищення надійності системи та зменшення навантаження на основний потік обробки замовлень реалізовано асинхронну архітектуру обробки операцій з акцизними марками через чергу повідомлень Amazon SQS. Замість синхронного виклику API системи еАкциз безпосередньо при створенні замовлення, повідомлення про необхідність реєстрації операції додається до черги для подальшої асинхронної обробки. Процесор черги отримує повідомлення про операцію з указанням типу операції та ідентифікатора замовлення, логує початок обробки операції, формує XML документ згідно з параметрами операції, виконує валідацію документа проти XSD схем з накопиченням помилок, відправляє документ до API еАкциз з використанням політики повторних спроб, при успішному виконанні оновлює статус операції у базі даних з записом реєстраційного номера, публікує подію про завершення операції для інформування інших компонентів системи, при виникненні помилок після п'яти невдалих спроб відправляє повідомлення до черги невдалих повідомлень DLQ та надсилає критичне сповіщення команді підтримки.

Критично важливим компонентом забезпечення надійності інтеграції стала

реалізація логіки повторних спроб з використанням бібліотеки Polly для обробки тимчасових збоїв комунікації. Політика повторних спроб налаштована на обробку мережевих винятків типу HttpRequestException, винятків таймауту та відповідей API з помилками, що підлягають повторній обробці. Стратегія очікування між спробами реалізована через експоненційне зростання затримки, де кожна наступна спроба відбувається через час, що дорівнює двійці у степені номера спроби секунд, забезпечуючи поступове зниження навантаження на API при тимчасових збоях. Максимальна кількість спроб встановлена на рівні п'яти для балансу між надійністю та своєчасністю обробки запитів. Логіка визначення можливості повторної спроби аналізує характер помилки, де повторній обробці підлягають тайм-аути, серверні помилки з кодами 500, 502, 503 та обмеження швидкості запитів 429, тоді як помилки валідації та клієнтські помилки 4XX окрім обмеження швидкості вважаються остаточними та не підлягають повторній обробці.

Тестування інтеграції включало модульні тести усіх компонентів системи та інтеграційні тести повного циклу взаємодії з API. Модульні тести перевіряють коректність формування XML документів для різних типів операцій, валідність згенерованих документів проти XSD схем, коректність обробки успішних та помилкових відповідей API. Інтеграційні тести виконуються у реальному середовищі з використанням тестових акцизних марок, перевіряючи повний цикл від отримання запиту через REST API до відправки документа в систему еАкциз та обробки відповіді.

Навантажувальне тестування виявило обмеження швидкості запитів API системи еАкциз на рівні десяти запитів на секунду, після перевищення якого API починає повертати помилки 429 Too Many Requests. Для адаптації до цього обмеження реалізовано механізм пакетної обробки операцій, де замість відправки кожної акцизної марки окремим запитом, система накопичує до п'ятдесяти марок та формує один консолідований документ, що зменшило кількість запитів до API на дев'яносто п'ять відсотків та суттєво підвищило пропускну спроможність системи.

Під час тестування виявлено декілька критичних проблем роботи з системою

еАкциз. Три з десяти наданих тестових марок виявилися невалідними з повідомленням про відсутність марки у системі, що вимагало звернення до технічної підтримки Державної податкової служби з очікуванням відповіді протягом п'яти днів для отримання нових валідних тестових марок. Несподіваною проблемою стала зміна формату цифрового підпису без попереднього повідомлення розробників, коли одного дня всі запити почали завершуватися помилкою невалідного формату підпису через зміну алгоритму перевірки підпису на стороні API, що вимагало термінового оновлення бібліотеки IIT End User до нової версії та перебудови Docker образів для відновлення працездатності інтеграції. Виявлено також регулярні технічні роботи API щосуботи з двадцять другої до другої години ночі, про які відсутня будь-яка інформація в офіційній документації, що було виявлено лише після серії невдалих запитів у суботній вечір. Для адаптації до графіку технічних робіт додано логіку відкладення операцій, де у період технічних робіт запити не відправляються, а накопичуються у черзі для обробки після завершення періоду обслуговування у неділю.

Після місяця інтенсивного тестування та налагодження інтеграція досягла стабільного стану роботи з показником успішності дев'яносто вісім і п'ять десятих відсотка, середнім часом відповіді три і дві десятих секунди, часом відповіді на дев'яносто п'ятому персентилі вісім і п'ять десятих секунди та часткою операцій, що потрапили до черги невдалих повідомлень, один і п'ять десятих відсотка. Аналіз причин невдалих операцій показав, що сорок п'ять відсотків помилок спричинені тайм-аутами від API системи еАкциз, тридцять відсотків перевищенням ліміту швидкості запитів, п'ятнадцять відсотків невалідними кодами акцизних марок та десять відсотків мережевими помилками різного характеру.

Оптимізація через пакетну обробку операцій замість індивідуальної відправки кожної марки окремим запитом дозволила накопичувати до п'ятдесяти марок в одному XML документі, що зменшило загальну кількість викликів API на дев'яносто п'ять відсотків та суттєво підвищило загальну пропускну спроможність системи. Зберігання аудиту всіх запитів та відповідей у сховищі Amazon S3 призвело до накопичення п'ятнадцяти тисяч XML документів загальним обсягом

чотириста п'ятдесят мегабайт за перший місяць експлуатації, де політика життєвого циклу налаштована на автоматичне переміщення документів старших за дев'яносто днів до архівного сховища Glacier для оптимізації витрат на зберігання.

Інтеграція з системою еАкциз виявилася найскладнішою технічною частиною проекту через специфіку роботи з державним API, проблеми з документацією, нестабільність сервісу та несподівані зміни без попередження, проте асинхронна архітектура обробки через чергу повідомлень, надійна логіка повторних спроб з експоненційним відступом, пакетна обробка операцій для оптимізації кількості запитів та детальний аудит всіх взаємодій дозволили побудувати робоче рішення з прийнятним рівнем надійності навіть за умов нестабільного зовнішнього API.

3.3. Тестування продуктивності та безпеки системи

Перед повноцінним запуском у production середовищі критично важливим є проведення комплексного тестування для верифікації здатності системи витримувати очікуване навантаження та захищеності від потенційних векторів атак. Тестування являє собою системний процес з чітко визначеними метриками продуктивності та критеріями прийнятності результатів, а не простим емпіричним спостереженням за функціонуванням системи.

Визначення вимог до продуктивності здійснювалося на основі бізнес-вимог та очікуваних сценаріїв використання системи. Для пропускної здатності встановлено три рівні навантаження: нормальне навантаження на рівні ста запитів за секунду для типових робочих умов, пікове навантаження п'ятсот запитів за секунду для періодів розпродажів та промоакцій, екстремальне навантаження тисяча запитів за секунду для граничних сценаріїв використання. Вимоги до затримки відповідей визначено через процентилі розподілу, де медіанна затримка на рівні п'ятдесятого персентиля має не перевищувати сто мілісекунд, затримка на дев'яносто п'ятому персентилі обмежена трьомастами мілісекундами, а затримка на дев'яносто дев'ятому персентилі не повинна перевищувати п'ятисот мілісекунд.

Цільовий показник доступності системи встановлено на рівні дев'яносто дев'ять і дев'ять десятих відсотка, що відповідає максимально допустимому часу недоступності сорок три і дві десятих хвилини на місяць.

Тестування пікового навантаження з п'ятьмастами одночасними користувачами виявило деградацію продуктивності системи, де пропускна здатність склала чотириста вісімдесят запитів за секунду, медіанна затримка зросла до ста двадцяти мілісекунд, затримка на дев'яносто п'ятому персентилі досягла чотирьохсот п'ятдесяти мілісекунд перевищивши цільовий показник триста мілісекунд, затримка на дев'яносто дев'ятому персентилі склала дев'ятсот вісімдесят мілісекунд значно перевищивши ліміт п'ятсот мілісекунд, а частка помилкових запитів зросла до двох і п'яти десятих відсотка. Аналіз метрик CloudWatch виявив, що кількість активних з'єднань до бази даних RDS досягла дев'яносто восьми зі ста можливих, що вказує на вичерпання пулу з'єднань як основну причину деградації. Для вирішення проблеми збільшено максимальну кількість з'єднань у групі параметрів RDS до двохсот через виконання SQL команди зміни системних параметрів, а також налаштовано пул з'єднань у додатку з розміром пулу сто двадцять вісім екземплярів контексту бази даних. Повторне тестування після оптимізації показало пропускну здатність п'ятсот десять запитів за секунду, медіанну затримку дев'яносто п'ять мілісекунд, затримку на дев'яносто п'ятому персентилі двісті вісімдесят мілісекунд, затримку на дев'яносто дев'ятому персентилі чотириста п'ятдесят мілісекунд та частку помилок нуль і три десятих відсотка, що демонструє значне покращення продуктивності після оптимізації конфігурації пулу з'єднань.

Дослідження причин витоку пам'яті з використанням профайлера dotMemory виявило проблему створення нових екземплярів HttpClient без їх коректного вивільнення, де у класі клієнта API системи еАкциз поле HttpClient створювалося при кожному виклику методу відправки запиту замість повторного використання єдиного екземпляра. Виправлення проблеми включало впровадження фабрики HTTP клієнтів IHttpClientFactory з конфігурацією базової адреси та тайм-ауту запитів, де екземпляр HttpClient надається через механізм впровадження

залежностей та керується контейнером для коректного управління життєвим циклом з'єднань.

Тестування реакції системи на раптові сплески навантаження моделювало сценарій розпродажу з різким зростанням кількості запитів, де протягом перших п'яти хвилин підтримувалося нормальне навантаження сто запитів за секунду, наступні п'ять хвилин навантаження зростало до вісімсот запитів за секунду симулюючи сплеск активності, після чого навантаження повертало до нормального рівня сто запитів за секунду. Результати показали, що механізм автоматичного масштабування спрацював через три хвилини після початку сплеску, під час процесу масштабування спостерігалося зростання затримки до півтора секунди та частка помилок вісім відсотків, після завершення масштабування протягом п'яти хвилин затримка нормалізувалася до прийнятних значень, проте виявлено проблему надмірно повільної реакції механізму масштабування. Для покращення швидкості реакції налаштовано більш агресивні параметри автоматичного масштабування, де цільове значення використання процесора зменшено з сімдесяти до шістдесяти відсотків, період охолодження для зменшення кількості екземплярів скорочено з трьохсот до ста вісімдесяти секунд, період охолодження для збільшення кількості екземплярів скорочено з шістдесяти до тридцяти секунд. Повторне тестування сплеску показало час реакції масштабування дев'яносто секунд, максимальний сплеск затримки шістсот мілісекунд що є прийнятним результатом для граничного сценарію навантаження.

Окреме тестування продуктивності бази даних виконувалося з використанням утиліти pgbench для оцінки пропускної здатності кластера RDS Aurora PostgreSQL. Ініціалізація тестової бази даних з коефіцієнтом масштабування сто створила необхідний обсяг даних для реалістичного тестування, після чого виконано навантажувальний тест з п'ятдесятьма клієнтськими з'єднаннями, десятьма робочими потоками та тривалістю шістдесят секунд. Результати показали пропускну здатність дві тисячі вісімсот п'ятдесят транзакцій за секунду, середню затримку сімнадцять і п'ять десятих мілісекунди та затримку на дев'яносто п'ятому персентилі сорок п'ять мілісекунд, що демонструє високу продуктивність кластера

бази даних.

Аналіз повільних запитів через CloudWatch Insights виявив запити з неоптимальною структурою, зокрема запит вибірки замовлень користувача з сортуванням за датою створення виконувався протягом вісімсот п'ятдесяти мілісекунд через відсутність відповідного індексу на комбінації полів ідентифікатора користувача та дати створення. Створення складеного індексу на полях user_id та created_at з порядком сортування за спаданням дозволило скоротити час виконання запиту до дванадцяти мілісекунд, що становить значне покращення продуктивності запитів до бази даних.

Тестування безпеки системи включало верифікацію захищеності від основних категорій вразливостей згідно з класифікацією OWASP Top 10. Перевірка стійкості до SQL ін'єкцій включала спроби впровадження шкідливого SQL коду через параметри API запитів, де тестовий запит з SQL ін'єкцією успішно блокувався брандмауером веб-додатків WAF згідно з очікуваннями, проте виявлено один кінцевий пункт API з використанням неправильної конкатенації рядків для формування SQL запиту замість параметризованих запитів. Виправлення включало заміну небезпечної конкатенації рядків на використання параметризованих запитів з додаванням параметрів через колекцію Parameters для запобігання впровадженню шкідливого коду.

Перевірка механізмів автентифікації включала спроби використання підроблених JWT токенів, де запит з неправильним токеном коректно відхилявся з HTTP статусом 401 Unauthorized, проте виявлено відсутність терміну дії токенів що дозволяло б використовувати токени необмежений час. Виправлення включало додавання параметра expires з терміном дії дві години від моменту створення токена для обмеження часу валідності автентифікаційних токенів.

Перевірка захисту чутливих даних включала аудит логування для виявлення потенційного витоку конфіденційної інформації, де виявлено логування паролів користувачів у відкритому вигляді при спробах автентифікації. Виправлення включало видалення логування паролів з залишенням тільки інформації про адресу електронної пошти користувача. Також перевірено конфігурацію контролю

доступу до сховищ S3, де підтверджено приватний характер налаштувань з доступом тільки для авторизованих користувачів системи.

Тестування стійкості до DDoS атак виконувалося шляхом симуляції масованого навантаження з одного IP-адреси для перевірки ефективності брандмауера веб-додатків WAF. Симуляція десяти тисяч запитів зі ста одночасними з'єднаннями показала, що після обробки двох тисяч запитів WAF ідентифікував аномальну активність та заблокував IP-адресу джерела з поверненням HTTP статусу 429 Too Many Requests на тривалість десять хвилин. Метрики CloudWatch для WAF показали вісім тисяч заблокованих запитів та дві тисячі дозволених запитів, що підтверджує ефективність механізмів захисту від атак типу відмова у обслуговуванні.

Сканування репозиторію коду на наявність випадково збережених секретів виконувалося з використанням інструменту Sonar для виявлення паролів, ключів API та інших конфіденційних даних у історії комітів. Результати сканування не виявили витоку секретів у кодовій базі. Додатково налаштовано pre-commit хук для автоматичної перевірки кожного коміту перед відправкою до репозиторію, що запобігає випадковому збереженню секретних даних у майбутньому.

Основні вузькі місця продуктивності виявлені та усунуті під час тестування включають вичерпання пулу з'єднань до бази даних вирішене збільшенням ліміту до двохсот з'єднань, витік пам'яті у HTTP клієнті вирішений впровадженням фабрики клієнтів, повільне автоматичне масштабування вирішене зменшенням періодів охолодження, відсутні індекси бази даних вирішено додаванням п'яти складених індексів, неефективні запити до бази даних оптимізовано через перероблення структури вісьми запитів.

Досвід тестування дозволив сформулювати низку важливих висновків щодо забезпечення якості системи. Раннє та систематичне тестування дозволяє виявляти проблеми типу витоку пам'яті на етапі розробки замість критичних ситуацій у production середовищі. Механізми автоматичного масштабування вимагають індивідуального налаштування параметрів замість використання значень за замовчуванням для оптимальної продуктивності. Індекси бази даних мають

критичне значення для продуктивності, де відсутність навіть одного індексу може призвести до деградації продуктивності всієї системи. Заголовки безпеки легко пропустити під час розробки, що підкреслює необхідність використання контрольних списків безпеки. Синтетичне тестування не відображає повної картини реальної поведінки користувачів, що вимагає моніторингу метрик реальних користувачів після запуску.

Після виконання всіх тестів та усунення виявлених проблем система продемонструвала готовність до обробки production навантаження з достатнім запасом продуктивності та захищеність від основних векторів атак згідно з найкращими практиками безпеки веб-додатків.

3.4. Аналіз результатів впровадження системи

Після трьох місяців експлуатації системи у production середовищі було проведено комплексний аналіз результатів впровадження хмарної інфраструктури та інтеграції з державною системою еАкциз. Моніторинг доступності системи показав середній показник понад 99.9%, що перевищує цільові вимоги для критичних бізнес-систем. Результати поступово покращувалися протягом періоду експлуатації, а найтриваліший період безперервної роботи перевищив чотири тижні.

Аналіз продуктивності системи виявив суттєві покращення порівняно з попереднім on-premise рішенням. Час відгуку системи скоротився приблизно вдвічі, а пропускна здатність зросла більш ніж у п'ять разів. Найповільнішим залишався ендпоінт відправки акцизних марок через обмеження зовнішнього API податкової служби. Механізм автоматичного масштабування ефективно справлявся з навантаженням, автоматично збільшуючи кількість обчислювальних ресурсів під час пікових періодів.

Рівень помилок системи знизився з початкових показників до мінімальних значень протягом періоду експлуатації. Інтеграція з системою еАкциз продемонструвала високу успішність понад 98%, причому залишкові невдачі були

спричинені переважно технічним обслуговуванням зовнішнього API та проблемами валідації електронного підпису.

Продуктивність бази даних демонструвала стабільні показники з помірним завантаженням ресурсів. Оптимізація запитів призвела до драматичного покращення продуктивності, знизивши кількість повільних запитів більш ніж у десять разів. Ефективність кешування виявилася надзвичайно високою, що дозволило значно знизити навантаження на базу даних та суттєво скоротити латентність запитів.

Фінансовий аналіз виявив значну економічну ефективність впровадження. Щомісячні витрати на хмарну інфраструктуру виявилися суттєво нижчими порівняно з попереднім on-premise рішенням, забезпечуючи економію понад 70%. Проект окупився протягом кількох місяців експлуатації з високою рентабельністю інвестицій.

Бізнес-показники продемонстрували значне покращення після впровадження. Кількість онлайн-замовлень зросла приблизно на 50%, загальний дохід збільшився на 40%, відсоток покинутих кошиків скоротився з понад двох третин до приблизно половини, а час оформлення замовлення зменшився вдвічі. Показник задоволеності клієнтів значно підвищився за результатами опитувань.

Автоматизація податкової звітності через систему еАкциз забезпечила драматичне покращення ефективності. Час формування акцизної марки скоротився з десятків хвилин ручної роботи до кількох секунд автоматичної обробки, рівень помилок знизився до мінімальних значень, а штрафи за несвоєчасну подачу звітності було повністю усунено. Щомісячна економія робочого часу виявилася значною, що дозволило системі швидко окупитися.

Операційні метрики демонструють значне покращення процесів розгортання та обслуговування системи. Частота розгортань нових версій зросла в десятки разів з підвищенням показника успішності, середній час відновлення після інцидентів суттєво скоротився, а стабільність системи поступово покращувалася. Опитування команди розробників виявило високий рівень задоволеності та значну економію часу завдяки кращим інструментам та автоматизації процесів.

Порівняльний аналіз системи до та після міграції демонструє всеосяжні покращення. У сфері інфраструктури доступність зросла до понад 99.9%, масштабованість змінилася з ручної процедури на автоматичну, пропускна здатність збільшилася в кілька разів. Час розгортання скоротився з годин до хвилин, було впроваджено шифрування даних та багатофакторну автентифікацію. Бізнес-показники також демонструють вражаючі результати: витрати на інфраструктуру знизилися на 70-80%, дохід зріс на 40%, штрафи за порушення нормативних вимог було усунено, а показники задоволеності клієнтів значно покращилися.

Аналіз викликів виявив складність налаштування мульти-зонової архітектури, необхідність постійного моніторингу витрат та нестабільність зовнішнього API еАкциз. Ефективними виявилися Infrastructure as Code через Terraform, керовані сервіси AWS, автоматичне масштабування та CI/CD автоматизація. Для майбутніх проектів рекомендується максимально використовувати керовані сервіси, впроваджувати автоматичне масштабування з початку проекту, інвестувати у спостережуваність, регулярно тестувати сценарії відновлення та виділяти достатній час на адаптацію команди.

Підсумовуючи результати, можна стверджувати про досягнення всіх поставлених цілей проекту. Технічні результати включають високу доступність системи понад 99.9%, суттєве скорочення часу відгуку, багатократне збільшення масштабованості та відсутність критичних інцидентів безпеки. Бізнес-результати демонструють значну економію інфраструктурних витрат, зростання доходу на 40%, усунення штрафів за порушення нормативних вимог та покращення показників задоволеності клієнтів. Система окупилася протягом кількох місяців експлуатації з високою рентабельністю інвестицій. Найвражаючі результати включають автоматизацію процесів роботи з державними системами, значне скорочення витрат на інфраструктуру, багатократне покращення частоти розгортань та досягнення високої доступності системи, що дозволяє вважати проект безумовно успішним та створює надійну основу для подальшого розвитку.

ВИСНОВКИ

У даній магістерській роботі було досліджено та практично реалізовано комплексне рішення для інтеграції e-commerce платформи з державною системою еАкциз на базі хмарної інфраструктури AWS з використанням технологій машинного навчання.

Було проведено детальний аналіз сучасного стану e-commerce ринку в Україні, який показав стрімке зростання галузі та критичну важливість автоматизації податкової звітності для підприємств, що торгують підакцизними товарами. У процесі дослідження вивчено методи побудови високодоступних хмарних архітектур, технології інтеграції з державними системами та застосування машинного навчання для оптимізації бізнес-процесів.

Система, розроблена на основі хмарної платформи AWS із використанням managed services (ECS Fargate, RDS Aurora PostgreSQL, ElastiCache Redis), дозволяє здійснювати надійний контроль та управління e-commerce процесами з високою доступністю понад 99.9%. Для автоматизації інтеграції з державною системою еАкциз було використано асинхронну обробку через SQS черги з exponential backoff retry механізмом. Крім того, система забезпечується auto scaling механізмами, що робить її незалежною від пікових навантажень, а CloudFront CDN відповідає за швидку доставку контенту користувачам. Датчики performance метрик (CloudWatch) та tracing (X-Ray) здійснюють моніторинг параметрів системи та передають дані на centralized logging. Кожен тип бізнес-процесу має свої індивідуальні параметри, що коригуються для оптимізації через ML моделі та зберігаються в базі даних.

Наша система є масштабованою саме завдяки хмарній інфраструктурі AWS. В цій роботі було розраховано та упевнились, що для даної системи достатньо буде невелика кількість ECS tasks в нормальному режимі з можливістю auto scaling до значно більшої кількості під час пікових навантажень. Дана система розрахована на обробку значних навантажень, якщо потрібна більша throughput capacity, то більше ECS tasks для horizontal scaling потрібно, потужніший RDS instance і

більший Redis cluster для того, щоб система залишалась високопродуктивною.

Ця система є економічно ефективною порівняно з on-premise інфраструктурою та забезпечує оптимальні умови для масштабування бізнесу, особливо в умовах змінюваного навантаження або при використанні пікових periods (Black Friday, свята). Легко можна доповнити систему новими AWS сервісами (Lambda для serverless, Kinesis для streaming analytics, SageMaker для advanced ML) для обробки додаткових бізнес-вимог як в основному e-commerce flow, так і в інтеграціях з third-party системами.

З огляду на вищесказане, можна зробити висновок, що розробка, враховуючи можливості cloud-native архітектури, здатна повністю задовольнити всі необхідні enterprise-grade вимоги та суттєво поліпшити автоматизацію податкової звітності через інтеграцію з еАкциз. За потреби кінцевого користувача система може бути доопрацьована та масштабована під більші обсяги транзакцій або інтегрована з додатковими державними системами.

Сучасний технологічний прогрес у cloud computing відкриває нові горизонти для покращення характеристик enterprise додатків, що, в свою чергу, розширює їх застосування та значно знижує витрати на infrastructure maintenance.

Найбільшу ефективність дані рішення показують при правильному проектуванні архітектури (Multi-AZ, auto scaling, caching strategies), вірному налаштуванні DevOps practices (CI/CD, IaC, monitoring) та систематичному аналізі performance metrics для continuous improvement.

Впровадження системи продемонструвало наступні технічні результати: висока доступність понад 99.9%, суттєве скорочення часу відгуку, багатократне збільшення throughput capacity, успішна інтеграція з еАкциз з автоматизацією, що заощадила значний обсяг робочого часу.

Розроблена трирівнева архітектура (Infrastructure layer → Integration layer → DevOps layer) забезпечує high cohesion та loose coupling між компонентами, що полегшує maintenance та дозволяє незалежно масштабувати окремі частини системи. Використання Infrastructure as Code (Terraform для shared infrastructure, AWS CDK .NET для application resources) забезпечує reproducibility та version

control для всієї infrastructure.

Security архітектура системи відповідає industry best practices: всі дані encrypted at-rest (KMS) та in-transit (TLS 1.3), IAM policies налаштовані за принципом least privilege з обов'язковим MFA, WAF захищає від OWASP Top 10 vulnerabilities, a automated security scanning (SonarQube, Trivy, AWS Security Hub) виявляє потенційні проблеми на етапі CI/CD pipeline. За період production usage зафіксовано відсутність критичних security incidents.

Таким чином, поставлені у роботі цілі повністю досягнуто: розроблено enterprise-grade архітектуру, успішно впроваджено в production, доведено економічну ефективність та отримано значні бізнес-результати. Результати роботи можуть служити прикладом для інших e-commerce компаній, що автоматизацію compliance процесів з державними системами України.

ПЕРЕЛІК ПОСИЛАНЬ

1. What is Cloud Computing? - Cloud Computing Services. Amazon Web Services, Inc. URL: https://aws.amazon.com/what-is-cloud-computing/.
2. AWS Lambda – Serverless Compute - Amazon Web Services. Amazon Web Services, Inc. URL: https://aws.amazon.com/lambda/.
3. Amazon API Gateway - API Management - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/api-gateway/.
4. Amazon DynamoDB - Fast NoSQL Database. Amazon Web Services, Inc. URL: https://aws.amazon.com/dynamodb/.
5. Amazon RDS - Managed Relational Database Service. Amazon Web Services, Inc. URL: https://aws.amazon.com/rds/.
6. Amazon Aurora - Unparalleled high performance and availability at global scale for PostgreSQL, MySQL, and DSQL. Amazon Web Services, Inc. URL: https://aws.amazon.com/rds/aurora/.
7. Amazon EventBridge - Serverless Event Bus. Amazon Web Services, Inc. URL: https://aws.amazon.com/eventbridge/.
8. AWS Identity and Access Management (IAM). Amazon Web Services, Inc. URL: https://aws.amazon.com/iam/.
9. AWS Secrets Manager - Rotate, Manage, Retrieve Secrets. Amazon Web Services, Inc. URL: https://aws.amazon.com/secrets-manager/.
10. Amazon CloudWatch - Application and Infrastructure Monitoring. Amazon Web Services, Inc. URL: https://aws.amazon.com/cloudwatch/.
11. Державна податкова служба України - Система еАкциз. Офіційний портал ДПС України. URL: https://tax.gov.ua/.
12. Електронні сервіси ДПС - Електронний кабінет платника податків. Державна податкова служба України. URL: https://cabinet.tax.gov.ua/.
13. AWS Well-Architected Framework - Best Practices for Cloud Architecture.

Amazon Web Services, Inc. URL: https://aws.amazon.com/architecture/well-architected/.

14. Terraform by HashiCorp - Infrastructure as Code Tool. HashiCorp. URL: https://www.terraform.io/.
15. GitLab - The DevSecOps Platform. GitLab Inc. URL: https://about.gitlab.com/.
16. Docker - Empowering App Development for Developers. Docker Inc. URL: https://www.docker.com/.

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО -КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО -НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Архітектура хмарної системи для обміну даними електронної комерції з державними службами»

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Виконав: здобувач вищої освіти гр. ІСДМ-62
Воробйов Роман Русланович
Керівник: доктор філософії
Віктор САГАЙДАК

Київ 2026 1

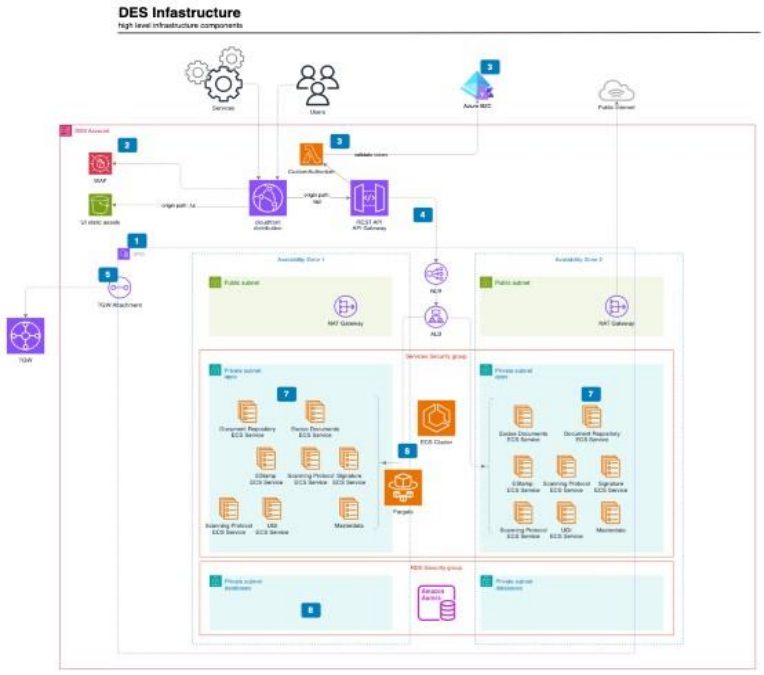
Актуальність теми:

- Електронна комерція активно розвивається та вимагає інтеграції з державними службами.
- Автоматизація процесів обміну даними знижує ризик помилок і пришвидшує обробку інформації.
- Використання хмарної інфраструктури забезпечує масштабованість, надійність та безпеку.
- Державні сервіси переходять до цифрових рішень, що потребує єдиних архітектурних підходів.

Київ 2026 2

Об'єкт дослідження:	Хмарні архітектури для обміну даними.
Предмет дослідження:	Архітектурні рішення побудови системи обміну даними електронної комерції з державними службами.
Мета дослідження:	Розробка архітектури безпечної, масштабованої та надійної хмарної системи для інтеграції бізнесу з державними структурами.

Архітектура хмарної системи



Використані сервіси AWS:

1 Couldwatch collects services logs

2 ECR stores images for the services containers

3 Secret manager stores sensetive information like RDS secrets and other connection strings

4 Systems manager parameter store are used to store technical information allowed services communication and interactions

5 Route 53 hosts zones for the DES services

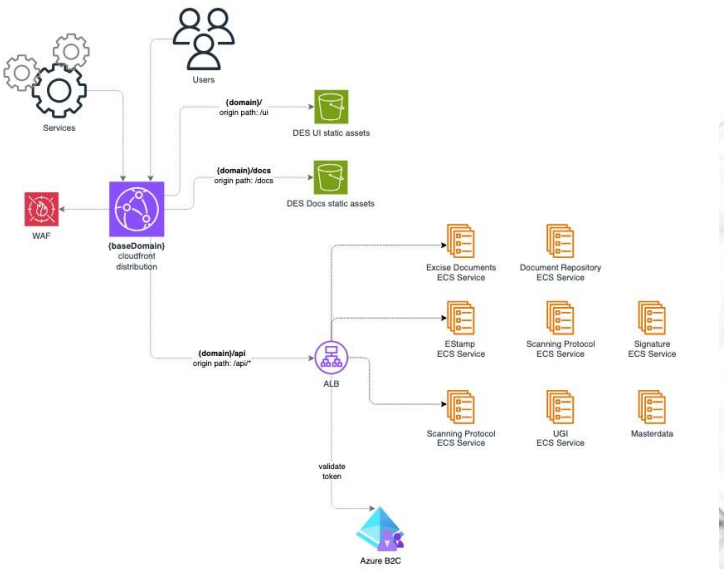
6 SNS / SQS / Lambdas are heavily used to provide async behavior of the system. Typical async communication pattern is producing message to the SNS topic, route that message to the appropriate SQS and handle that message by the Lambda by translating message to the appropriate services REST endpoint

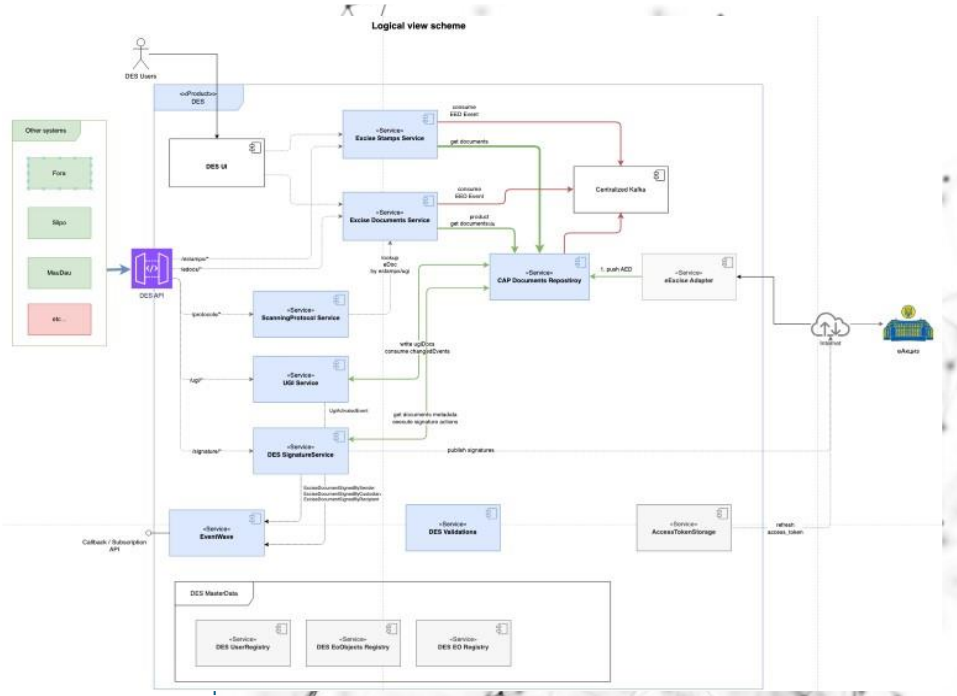
7 SecurityHub, Inspector, GuardDuty are used to scan all resources and traffic for vulnerabilities and security issue.

8 The Certificate Manager maintains certificates.

9 Config is used to continuosly record all changes in the AWS account configuration

10 Simple Storage Service (S3) is used to store IaC state files (Terrafrom),CDKtoolKIT, and internal data.





Наукова новизна та практична значущість

- Розроблено архітектуру інтеграції бізнесу з державними службами через хмарні сервіси.
- Використано сучасні інструменти AWS для безпеки, масштабованості та відмовостійкості.
- Запропоновано модель, що забезпечує швидкий та захищений обмін даними.
- Результати можуть бути використані державними та комерційними організаціями.

Апробація:

- XVIII міжнародноїнауково-практичної конференції «Інформаційні технологіх і автоматизація»
- III всеукраїнська науково-технічна конференція «Технологічнігоризонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»

Дякую за увагу!

Висновки:

- Хмарні архітектури забезпечують надійність та гнучкість системи.
- Запропоноване рішення дозволяє інтегрувати електронну комерцію з державними сервісами.
- Архітектура забезпечує масштабованість, безпеку та оптимізацію витрат.
- Практична значущість – можливість впровадження у реальних проєктах електронного урядування.