

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації

Кафедра Радіотехнологій інформаційно-комунікаційних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 126 Інформаційні системи та технології
(код і повна назва)

Освітня програма Архітектурне проєктування інформаційних систем
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Кущову Олександрю Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Архітектурне проєктування інтегрованих рішень для медицини надзвичайних ситуацій з використанням хмарних безсерверних рішень

затверджена наказом по університету від **3 жовтня 2023 р. № 1295 Ст**

2. Термін подання студентом роботи до екзаменаційної комісії **08 січня 2024 р.**

3. Вихідні дані до роботи:

розробити архітектуру інтегрованого рішення для підприємств медицини надзвичайних ситуацій, яка базується на хмарних та безсерверних рішеннях, що дозволяє використовувати переваги та потужності хмарних сервісів для досягнення високої гнучкості процесу розгортання додатку, а також доступності сервісів

4. Перелік питань, що потрібно опрацювати в роботі:

аналіз предметної області, постановка задачі, аналіз існуючих архітектур додатків, вибір загальної архітектури інтегрованого рішення, аналіз існуючих хмарних та безсерверних рішень, розробка архітектури компонентів додатку, розробка алгоритму інтеграції з зовнішніми системами, розробка підходів до процесу розгортання додатку

. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Комп'ютерна презентація

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Дудка О.О.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Ознайомлення із завданням. Уточнення технічного завдання	06.11.23	вик.
2	Аналіз літератури за темою роботи	07.11-03.12.23	вик.
3	Підготовка теоретичного розділу	04.12-25.12.23	вик.
4	Розробка проєктного розділу	26.12.23-07.01.24	вик.
5	Оформлення презентаційного матеріалу, подання роботи на кафедрі. Підготовка до захисту у ЕК.	08.12.23-08.01.24	вик.

Дата видачі завдання **4 жовтня 2023 р.**

Студент _____
(підпис)

Кушов О.С.

(прізвище та ініціали)

Керівник роботи _____
(підпис)

доц. Дудка О.О.

(посада, прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 121 с., 55 рис., 14 табл., 82 джерела, 3 додатки.

Ключові слова: МЕДИЦИНА, НАДЗВИЧАЙНІ СИТУАЦІЇ, БЕЗСЕРВЕРНІ РІШЕННЯ, ХМАРНІ ТЕХНОЛОГІЇ, АРХІТЕКТУРА, АВТОМАТИЧНЕ РОЗГОРТАННЯ, ПРОДУКТИВНІСТЬ, БЕЗПЕКА

Ця пояснювальна записка до Кваліфікаційної роботи магістра присвячена розробці та аналізу архітектурного проекту для програмних комплексів, призначених для компаній у сфері медицини надзвичайних ситуацій. Основними завданнями цього дослідження є визначення ефективних архітектурних рішень та обґрунтування їх використання для покращення якості та швидкості надання медичної допомоги в критичних обставинах.

В першому розділі пояснювальної записки розглянуті основні тенденції, виклики та особливості розвитку архітектури програмного забезпечення в медицині, а також деякі специфічні для галузі програмні поняття. Також проаналізовані хмарні та безсерверні технології, які можуть стати важливими компонентами архітектурних рішень у цій галузі.

Другий розділ присвячено самому архітектурному проекту. У цьому розділі визначені технічні вимоги до програмного комплексу та детально розглянуті архітектурні рішення, їх обґрунтування та візуалізація.

Третій розділ містить практичну реалізацію розгортання частини обраної архітектури, а також можливі результати та висновки, отримані під час впровадження.

Висновки та перспективи розглянуті в четвертому розділі допоможуть підсумувати результати дослідження та вказати напрямки подальшого вдосконалення архітектурних рішень для медицини надзвичайних ситуацій.

THE ABSTRACT

Explanatory note: 121 p., 55 fig., 14 tabl., 82 sources, 3 app.

Keywords: MEDICAL, EMERGENCY, SERVERLESS SOLUTIONS, CLOUD TECHNOLOGIES, ARCHITECTURE, AUTOMATED DEPLOYMENT, PRODUCTIVITY, SECURITY

This explanatory note for the Master's thesis is dedicated to the development and analysis of an architectural project for software complexes designed for companies in the field of emergency medicine. The main objectives of this research are to determine effective architectural solutions and justify their use to enhance the quality and speed of medical assistance in critical situations.

The first chapter of the explanatory note discusses the key trends, challenges, and peculiarities in the development of software architecture in the field of medicine, as well as some domain-specific software concepts. It also analyzes cloud and serverless technologies, which can be significant components of architectural solutions in this domain.

The second chapter is devoted to the architectural project itself. This section defines the technical requirements for the software complex and provides a detailed examination of architectural decisions, their justifications, and visualization.

The third chapter contains the practical implementation of a part of the chosen architecture deployment, as well as potential results and findings obtained during the implementation.

The conclusions and prospects discussed in the fourth chapter will help summarize the research results and indicate directions for further improvement of architectural solutions for emergency medicine.

ЗМІСТ

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	7
ВСТУП.....	9
1 ВИКОРИСТАННЯ ХМАРНИХ ТА БЕЗСЕРВЕРНИХ РІШЕНЬ В СУЧАСНОМУ МЕДИЧНОМУ ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ.....	10
1.1 EMR та EHR системи.....	10
1.2. Хмарні медичні SaaS рішен.....	14
1.3 Архітектура SaaS рішення для медичних додатків.....	15
1.4. Безпека медичних даних та персональних даних пацієнтів.....	18
1.5 Уніфіковані структури медичних даних.....	22
1.6 Джерела медичних даних.....	23
1.7 Аналіз медичних даних.....	26
1.8 Доступність медичних систем.....	28
1.9. Сервіси хмарного обчислення та інтеграції всередині системи.....	29
1.10. Розгортання систем.....	31
1.11 Висновки щодо використання хмарних та безсерверних рішень в медицині.....	32
1.12 Інформаційні системи медицини надзвичайних ситуацій.....	32
2 РОЗРОБКА АРХІТЕКТУРИ ДЛЯ ПІДПРИЄМСТВ МЕДИЦИНИ НАДЗВИЧАЙНИХ СИТУАЦІЙ.....	34
2.1 Опис компанії для аналізу.....	34
2.2 Контекст технічного завдання.....	35
2.3 Опис технічного завдання.....	36
2.4 Концептуальна архітектура.....	42
2.5. Ключові особливості взаємодії компонентів.....	49
2.6 Архітектура розгортання та інфраструктури.....	53
2.7 Забезпечення атрибутів якості.....	59
2.8 Аналіз ризиків та архітектурних обмежень.....	64
2.9 Розробка та підтримка проекту.....	66
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ РОЗГОРТАННЯ АРХІТЕКТУРИ.....	71
3.1. Обґрунтування частини проекту для практичної реалізації.....	71
3.2 Загальне розуміння CloudFront CDN.....	71
3.3 Загальне розуміння CloudFormation.....	74
3.4 Формування скриптів та параметрів розгортання.....	77
3.5 Підсумки та висновки щодо практичної реалізації.....	88

4 ВИСНОВКИ ТА ПЕРСПЕКТИВИ РОЗВИТКУ ПРОГРАМНИХ РІШЕНЬ ДЛЯ МЕДИЦИНИ ТА МЕДИЦИНИ НАДЗВИЧАЙНИХ СИТУАЦІЙ.....	92
4.1 Розвиток хмарних технологій в світі.....	92
4.2 Загальносвітові тенденції розвитку програмних рішень в медицині.....	94
4.3 Розвиток програмних рішень для медицини в Україні.....	96
4.4 Висновки щодо подальшого розвитку використання хмарних рішень в медицині та медицині надзвичайних ситуацій.....	102
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	106
ДОДАТОК А ПЕРЕЛІК АРХІТЕКТУРНИХ РІШЕНЬ.....	115
ДОДАТОК Б СЛАЙДИ ПРЕЗЕНТАЦІЇ.....	119
ДОДАТОК В ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	125

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

- AWS (Amazon Web Services) - Веб-сервіси Amazon
- CDS (Clinical Decisions Support) - Підтримка клінічних рішень
- CDN (Content Delivery Network) - Мережа доставки контенту
- CORS (Cross Origin Resource Sharing) - Перехресний обмін ресурсами
- CMMD (Connected Mobile Medical Devices) - Підключені мобільні медичні пристрої
- DDoS (Distributed Denial of Service) - Розподілена атака на відмову в обслуговуванні
- EMR (Electronic Medical Records) - Електронні медичні записи
- EHR (Electronic Health Records) - Електронні історії хвороб
- ETL (Extract, Transform, Load) - Видобуток, перетворення, завантаження
- FHIR (Fast Healthcare Interoperability Resources) - Ресурси оперативної сумісності швидкого медичного обслуговування
- GCP (Google Cloud Platform) - Хмарна платформа Google
- GDPR (General Data Protection Regulation) - Загальний регламент захисту даних
- GIDH (Global Initiative on Digital Health) - Глобальна ініціатива цифрового здоров'я
- HIPAA (Health Insurance Portability and Accountability Act) - Закон про переносимість та підзвітність медичного страхування
- HTTP(S) (HyperText Transfer Protocol (Secure)) - Протокол передачі гіпертексту (захищений)
- HL7 (Health Level 7) - Рівень здоров'я 7
- IaC (Infrastructure as Code) - Інфраструктура як код
- MDD (Medical Devices Directive) - Директива щодо медичних приладів
- MDR (Medical Device Regulation) - Положення про медичні вироби
- MDT (Multidisciplinary Team) - Мультидисциплінарна команда
- PHR (Personal Health Record) - Особиста медична книжка
- PHC (Primary Health Care) - Первинна медична допомога

REST (Representational State Transfer) - Передача представницького стану

SOA (Service Oriented Architecture) - Сервісно-орієнтована архітектура

SiMD (Software in a Medical Device) - Програмне забезпечення в медичному пристрої

SaMD (Software as a Medical Device) - Програмне забезпечення як медичний пристрій

SMART (Substitutable Medical Applications, Reusable Technologies) - Замінні медичні програми, багаторазові технології

SaaS (Software as a Service) - Програмне забезпечення як послуга

URI (Uniform Resource Identifier) - Уніфікований ідентифікатор ресурсів

ВСТУП

Медицина надзвичайних ситуацій охоплює широкий спектр послуг, включаючи, але не обмежуючись цим, надання невідкладної медичної допомоги, транспортування постраждалих до лікувальних закладів, координацію роботи медичного персоналу та обладнання в екстрених обставинах. Ця галузь вимагає надзвичайної точності, швидкості та координації для ефективного реагування на кризові ситуації.

Однак із зростанням глобальних загроз та насильства, трагічних природних лих та інших небезпечних ситуацій, вимоги до систем медицини надзвичайних ситуацій надзвичайно зростають. Для ефективного реагування на такі події необхідні високотехнологічні рішення, які дозволяють оперативно і точно вести координацію та надавати медичну допомогу. В останні десятиліття технології значно розвинулися і стали потужним інструментом для покращення систем медицини надзвичайних ситуацій. Хмарні та безсерверні рішення надають можливості для ефективного зберігання та обробки медичних даних, швидкого розгортання нових систем та додатків, а також забезпечують високу доступність та масштабованість. Використання таких технологій може покращити реагування на надзвичайні ситуації та допомогти врятувати більше життів.

Метою даної Кваліфікаційної роботи магістра є розробка та аналіз архітектурних рішень для програмних комплексів у галузі медицини надзвичайних ситуацій з використанням хмарних та безсерверних технологій. Дослідження спрямоване на покращення якості надання медичної допомоги у критичних ситуаціях та забезпечення більш швидкого та ефективного реагування на надзвичайні події.

Актуальність цього дослідження обумовлена зростанням потреби у сучасних технологіях для медицини надзвичайних ситуацій, а також потенціалом для покращення систем та збереження життів. Успішні рішення, розроблені в рамках цього дослідження, можуть мати великий практичний вплив на сферу медицини надзвичайних ситуацій та сприяти безпеці та добробуту громадян.

1 ВИКОРИСТАННЯ ХМАРНИХ ТА БЕЗСЕРВЕРНИХ РІШЕНЬ В СУЧАСНОМУ МЕДИЧНОМУ ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ

В сучасному світі, інформаційні технології відіграють важливу роль у сфері медицини, забезпечуючи швидкий доступ до медичних даних, покращення процесів лікування та діагностики, а також забезпечуючи координацію медичної допомоги у надзвичайних ситуаціях. Розвиток цієї галузі зазнав суттєвих змін завдяки впровадженню інноваційних архітектурних рішень та використанню хмарних та безсерверних технологій. У цьому розділі буде розглянуто сучасний стан медичного програмного забезпечення, виклики, з якими ця галузь стикається, а також основні архітектурні підходи, які допомагають подолати ці виклики.

1.1 EMR та EHR системи

Першим кроком у цьому дослідженні буде розгляд систем Електронних Медичних Записів (Electronic Medical Records - EMR) та Електронних Історій Хвороб (Electronic Health Records - EHR). Буде досліджено відмінності між ними, а також сучасні підходи до розробки цих систем в хмарних середовищах та можливості створення програмного забезпечення як сервісу (SaaS), що забезпечує доступність та скальованість.

EMR - це система для зберігання медичних записів пацієнтів в електронному форматі. Вони включають в себе історію пацієнта, результати обстежень, діагнози, призначення лікування та іншу інформацію, що важлива для медичного обслуговування конкретного пацієнта. EMR системи розробляються для внутрішнього використання в медичних установах, таких як лікарні та клініки, і зазвичай фокусуються на конкретному пацієнті.

EMR допомагають лікарям та медичному персоналу зручно зберігати та доступно управляти медичною інформацією. Це може покращити якість надання

медичної допомоги, спростити процес документування та діагностики, а також забезпечити історію лікування пацієнта для подальшого аналізу.

Електронні Історії Хвороб (EHR) розширюють поняття EMR, охоплюючи більший обсяг інформації про пацієнта та його медичну історію. EHR системи зазвичай охоплюють дані з різних медичних закладів та мають більш широкий фокус. Вони дозволяють обмін інформацією між різними установами та медичними спеціалістами, сприяючи координації медичної допомоги.

Основні відмінності між EMR та EHR полягають у здатності останнього обмінюватися даними між різними системами, стандартизації інформації та розширеному функціоналі. EHR системи допомагають створювати повнішу та загальну картину медичного стану пацієнта, що є важливим в контексті надання надзвичайної медичної допомоги.

Інтеграція електронних медичних записів (EMR) з системами електронних медичних історій (EHR) стала критично важливою в медичній галузі сьогодні. Ця інтеграція допомагає забезпечити повноту та однорідність медичної інформації про пацієнтів, сприяє безперервному обміну даних між різними медичними установами та полегшує надання координації медичної допомоги. Результатом є покращення рішень у сфері медицини та підвищення якості надання медичних послуг. Загальні принципи взаємодії EMR та EHR систем, а також систем надання персоналізованих медичних послуг PHR можна зазначити схематично як показано на рисунку 1.1 та на рисунку 1.2.

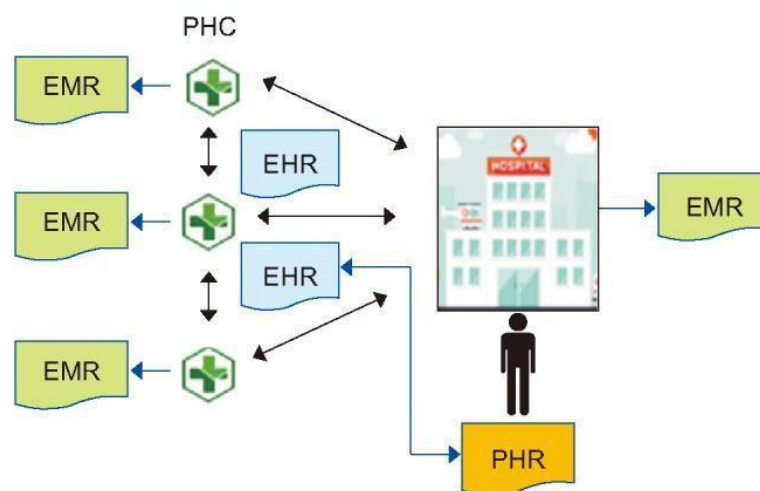


Рисунок 1.1 - Загальна схема взаємодії EMR/EHR/PHR систем [2]

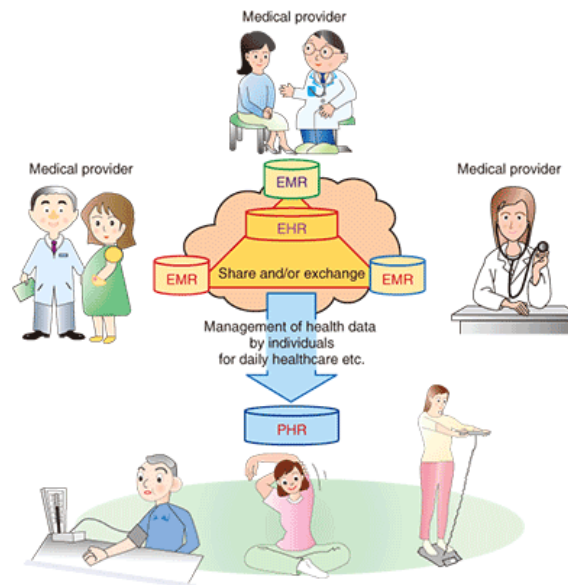


Рисунок 1.2 - Загальна схема взаємодії EMR/EHR/PHR систем [3]

Згідно з дослідженнями HealthIt.gov [4] щодо ринку США “Станом на 2021 рік майже 4 із 5 лікарів, що працюють у офісах (78%), і майже всі нефедеральні лікарні невідкладної допомоги (96%) адоптували сертифіковану EHR. Це знаменує значний 10-річний прогрес з 2011 року, коли 28% лікарень і 34% лікарів прийняли EHR.”. Візуально такий тренд представлений на рисунку 1.3.

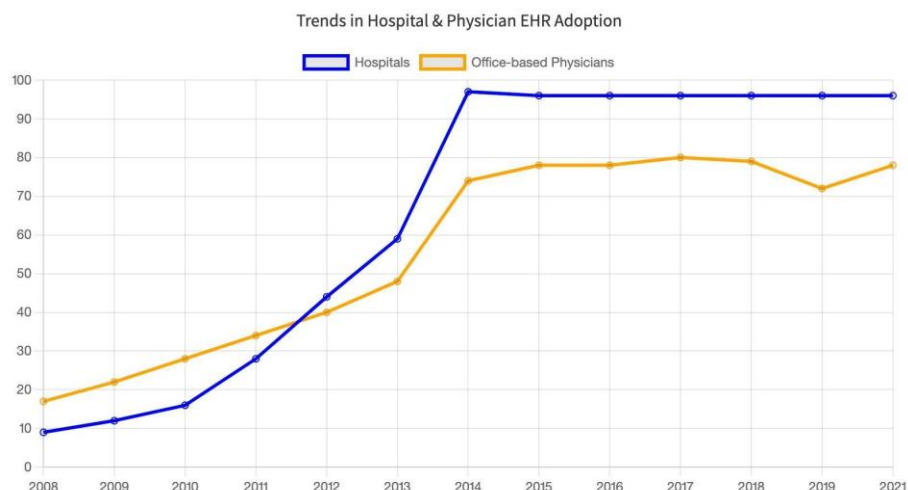


Рисунок 1.3 - Тренди інтеграції з EHR системами

Також HealthIt.gov зазначає появу та розвиток сертифікованих EHR систем. Де “Сертифікована система EHR – це система, яка відповідає вимогам, прийнятим Міністерством охорони здоров’я та соціальних служб США.”. Тенденції використання сертифікованих EHR систем зазначені на рисунку 1.4.

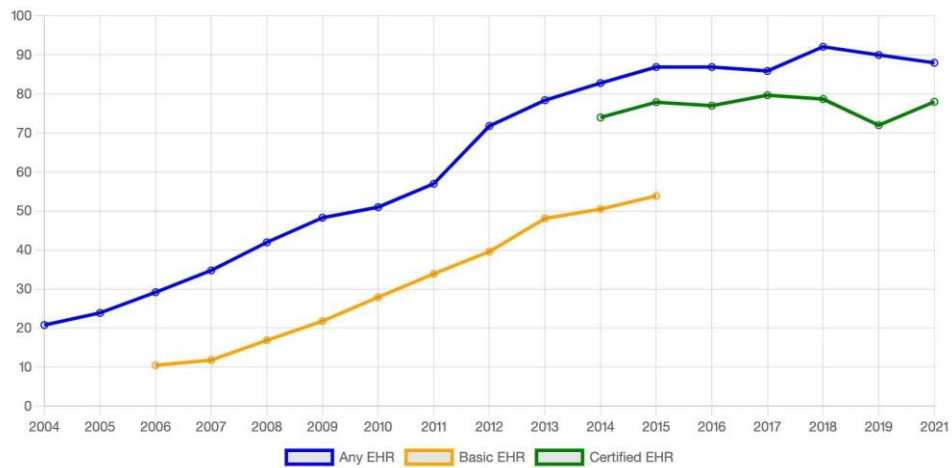


Рисунок 1.4 - Частки базових та сертифікованих EHR систем в загальній кількості

Враховуючи аналітику Data Bridge [5], щодо інвестицій в сферу медицини, можна також стверджувати, що розвиток інформаційних технологій в сфері медицини продовжуватиме свій ріст, що відображено на рисунку 1.5.

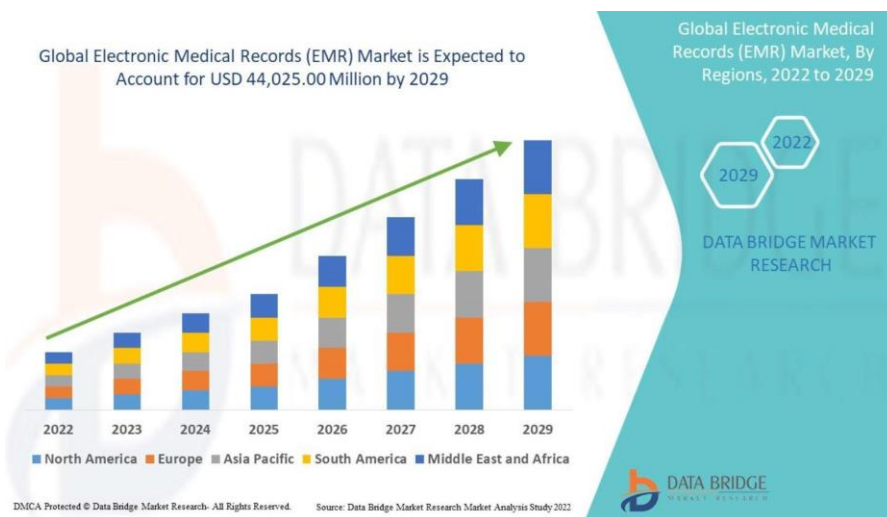


Рисунок 1.5 - Прогноз розвитку ринку EMR систем

За думкою Deloitte [6] “Майбутні рішення EHR повинні бути розроблені таким чином, щоб задовольнити потреби всіх користувачів, особливо потреб споживачів і клініцистів. Ці рішення також мають використовувати новітні технології, такі як хмарні екосистеми платформ, щоб покращити здоров’я та благополуччя споживачів.”.

1.2. Хмарні медичні SaaS рішення

За допомогою хмарних рішень, лікарні можуть легко масштабувати свої EMR системи вгору або вниз в залежності від потреби, забезпечуючи гнучкість та високу доступність. Крім того, хмарні платформи надають доступ до різноманітних ресурсів, включаючи обчислювальну потужність, сховища даних та мережеві послуги, що полегшує розгортання та забезпечує надійність системи.

Використання рішень на основі послуг (SaaS) в порівнянні з розробкою та впровадженням власної системи електронних медичних записів (EMR) має численні переваги, які роблять цей підхід привабливим для медичних установ та організацій. Серед значних переваг такого підходу слід зазначити наступні:

- використання SaaS рішень дозволяє легко та швидко впровадити систему EMR;
- розробка та підтримка власної EMR системи можуть бути вкрай витратними;
- SaaS рішення надають можливість легко масштабувати систему вгору або вниз в залежності від попиту;
- SaaS постачальники відповідають за оновлення та підтримку безпеки системи;
- використання SaaS рішень дозволяє медичному персоналу зосередитися на наданні медичних послуг, а не на технічних аспектах розробки та підтримки системи.

Загалом, використання SaaS рішень для систем EMR допомагає медичним установам знизити витрати, збільшити ефективність та покращити якість надання медичних послуг. Для кожного користувача системи доступна конфігурація власного процесу а також налаштування доступів, інтеграцій, обробки даних тощо.

1.3 Архітектура SaaS рішення для медичних додатків

В якості прикладу архітектури такого рішення обрано дослідження групи спеціалістів HIS (Health Informatics Research - Дослідження інформатики охорони здоров'я)[7].

Графічно загальна архітектура SaaS системи в представленні спеціалістів HIS наведена на рисунку 1.6.

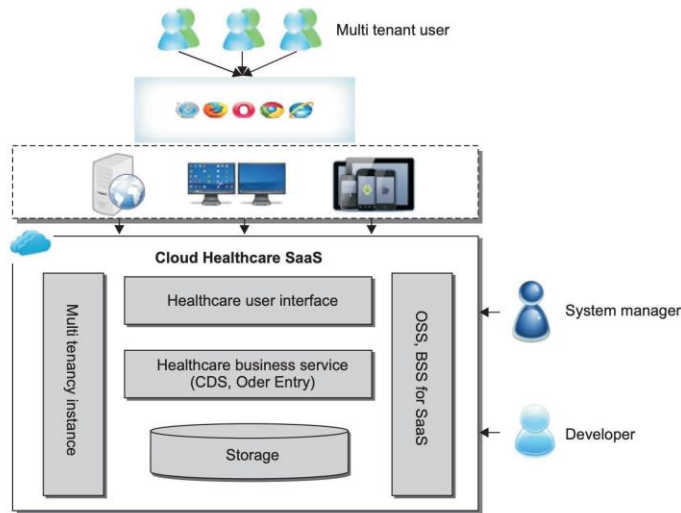


Рисунок 1.6 - Загальна бізнес архітектура медичного додатку на основі SaaS рішення

Внутрішня структура компонентів системи зазначена на рисунку 1.7.

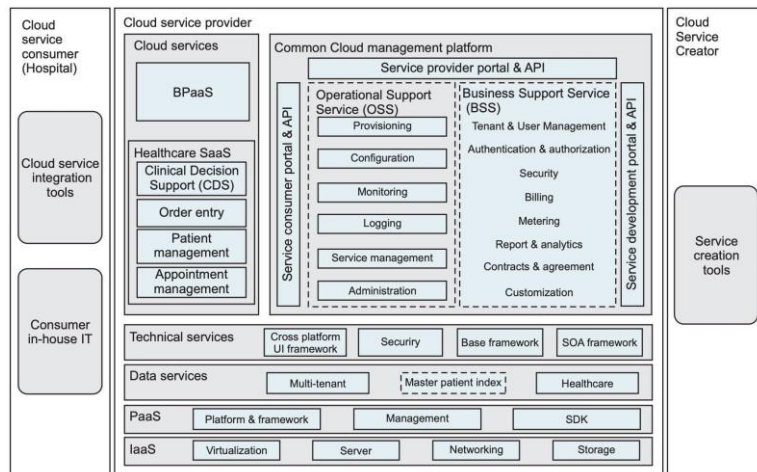


Рисунок 1.7 - Логічна структура компонентів медичного додатку на основі SaaS рішення

Така система включає наступні бізнес елементи:

- Зовнішні системи, які включають інтеграцію з іншими хмарними сервісами, а також с власними системами користувачів;
- Системи управління сервісами та їх розгортання;
- Платформенна частина - системи керування серверами та операційними частинами додатків;
- Сервіси даних, які включають роботу з структурованими медичними даними пацієнтів, налаштування та управління даними користувачів;
- Системи доступу та інтеграції, що являють собою базову частину додатку для взаємодії з користувачами через візуальні та програмні інтерфейси;
- Системи контролю для розробників, адміністраторів. Ця частина надає можливість додаткової індивідуальної конфігурації для користувачів, а також формує технічні та бізнес звіти.

З точки зору технологічного стеку, в дослідженні використані чисельні фреймворки та бібліотеки, які забезпечують роботу зазначених бізнес частин. Такі системи програмні комплекси забезпечують можливість розробки додатків зазначеного типу та є поширеними і мають гарну документацію і підтримку. Детальні технології наведені на рисунку 1.8.

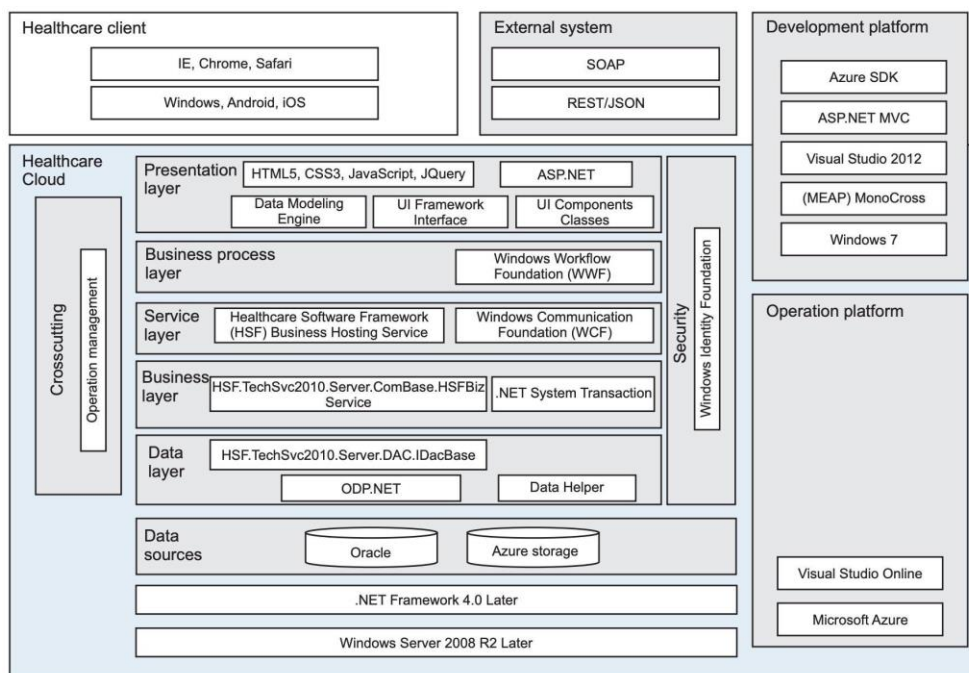


Рисунок 1.8 - Технологічний стек SaaS рішення для медичного додатку

Хмарні постачальники надають спеціалізовані сервіси для медичних установ та служб надзвичайних ситуацій. Ці сервіси можуть включати електронні медичні записи (EHR), системи аналізу даних для передбачення надзвичайних ситуацій, засоби комунікації та обміну даними між медичними працівниками та пацієнтами, інструменти для навігації та локалізації в надзвичайних ситуаціях, і багато інших. Ці готові сервіси дозволяють розробникам швидко створювати та інтегрувати додатки для медицини надзвичайних ситуацій, зменшуючи час та ресурси, необхідні для їх розробки та впровадження.

Для цілей цієї роботи зазначаються основні компоненти системи у вигляді діаграми, яка зображена на рисунку 1.9. та відображає зазначені раніше логічні компоненти системи.

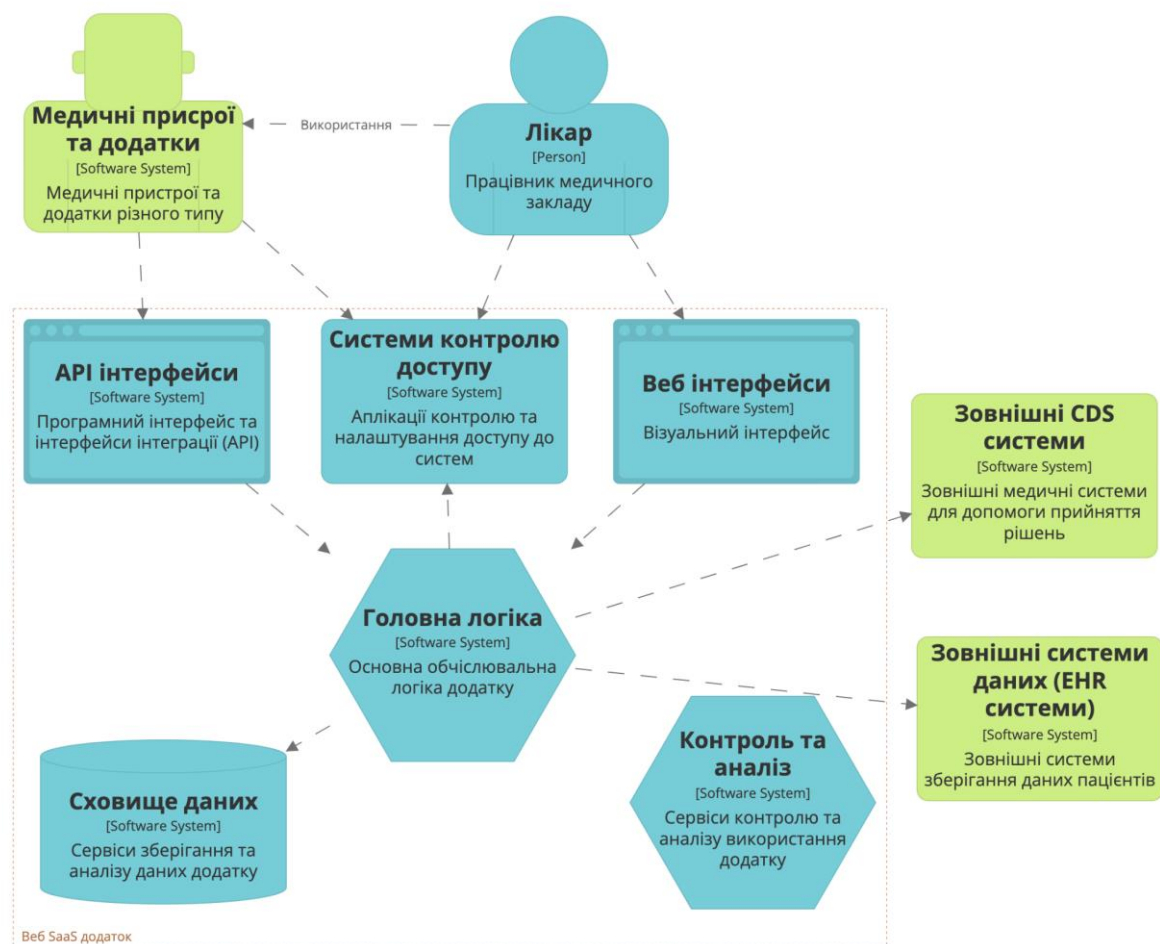


Рисунок 1.9 - Загальна логічна структура компонентів медичного додатку на основі хмарних та безсерверних рішень

Така візуалізація загальної структури додатку дозволить у подальшому

дослідити окремі компоненти системи та можливості використання існуючих хмарних та безсерверних рішень. Завдяки акценту цієї роботи на безсерверних рішеннях, базова архітектура системи не включає платформенної частини, оскільки ця робота буде покладена на постачальника хмарних послуг.

1.4. Безпека медичних даних та персональних даних пацієнтів

У сфері медицини існують строгі регуляторні акти, які регулюють збереження та обробку медичних даних. Один з найважливіших таких актів є Закон про медичну конфіденційність (HIPAA) у Сполучених Штатах, який встановлює стандарти та вимоги для захисту особистої медичної інформації. Цей закон вимагає від медичних організацій та провайдерів приймати заходи для забезпечення безпеки даних, включаючи шифрування, контроль доступу та аудити безпеки.

Крім HIPAA, існують інші міжнародні та національні нормативні акти, які регулюють захист медичних даних. В Європейському Союзі, Загальний регламент з захисту даних (GDPR) встановлює стандарти для обробки особистих даних, включаючи медичну інформацію.

Технологічні рішення, такі як шифрування даних, ідентифікація та автентифікація користувачів, а також системи моніторингу та аудиту безпеки, використовуються для виконання цих регуляторних вимог та забезпечення високого рівня захисту медичних даних. Всі ці заходи спрямовані на забезпечення конфіденційності пацієнтів та запобігання можливим порушенням безпеки даних у сфері медицини.

Стандарт "Substitutable Medical Applications, Reusable Technologies" (SMART) визначає стандартизований підхід до розробки медичних додатків та платформ, зокрема тих, які використовуються в сфері медицини надзвичайних ситуацій. SMART дозволяє легко інтегрувати та обмінювати даними між різними медичними системами та забезпечує високий рівень безпеки даних, включаючи автентифікацію та авторизацію.

Тож слід розглянути поняття безпеки окремими пунктами.

Контроль доступу. Цей аспект включає в себе управління тим, хто має доступ до медичних даних та як вони можуть взаємодіяти з цими даними. Він включає у себе аутентифікацію та авторизацію користувачів, а також встановлення рівнів доступу до різних частин системи. Відповідна частина загальної логічної системи зазначена кольором на рисунку 1.10.

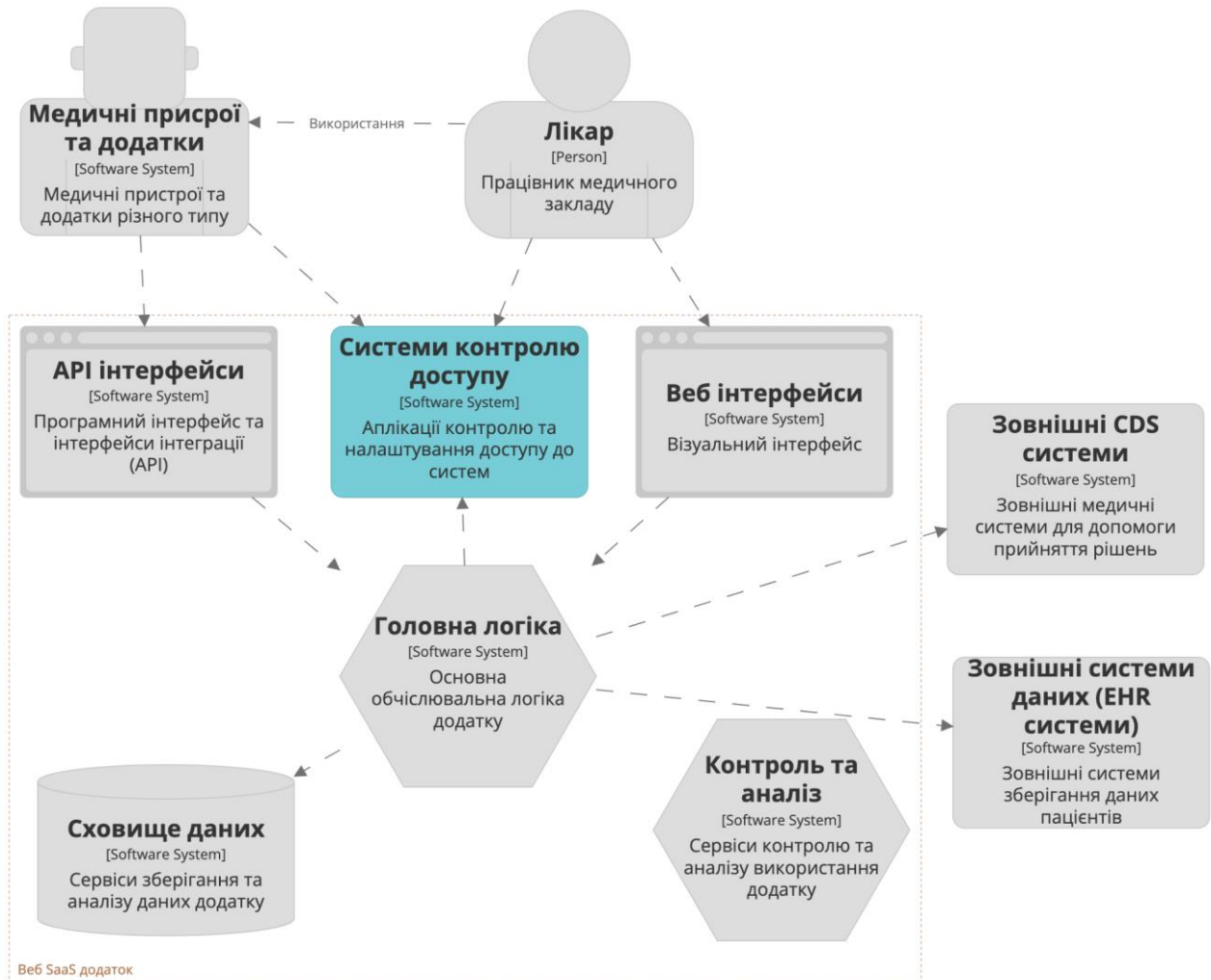


Рисунок 1.10 - Позиція систем контролю доступу у загальній архітектурі додатку

Для реалізації ефективного контролю доступу до медичної інформації в сучасних умовах, провайдери хмарних рішень надають готові сервіси, які дозволяють медичним установам ефективно керувати правами доступу користувачів. Зокрема, відомі хмарні платформи, такі як Amazon Web Services (AWS), Google Cloud Platform (GCP) та Microsoft Azure, пропонують спеціалізовані інструменти для автентифікації та авторизації користувачів.

Наприклад, AWS пропонує сервіс Cognito, який дозволяє медичним установам створювати та керувати ідентифікацією та авторизацією користувачів з використанням різних методів аутентифікації, включаючи соціальні мережі та мультифакторну аутентифікацію. Майже аналогічно, GCP пропонує свій Identity Platform, Azure надає Active Directory B2C для керування доступом користувачів та ідентифікацією.

Завдяки хмарним сервісам керування ключами та сертифікатами, такими як AWS KMS, Azure Key Vault, Google Cloud KMS, AWS Certificate Manager та інші, можливо також налаштування авторизованих доступів до зовнішніх систем, що не потребує значних витрат часу розробників.

Рисунок 1.11 зазначає можливу інтеграцію різних сервісів для забезпечення контролю доступу до додатку.

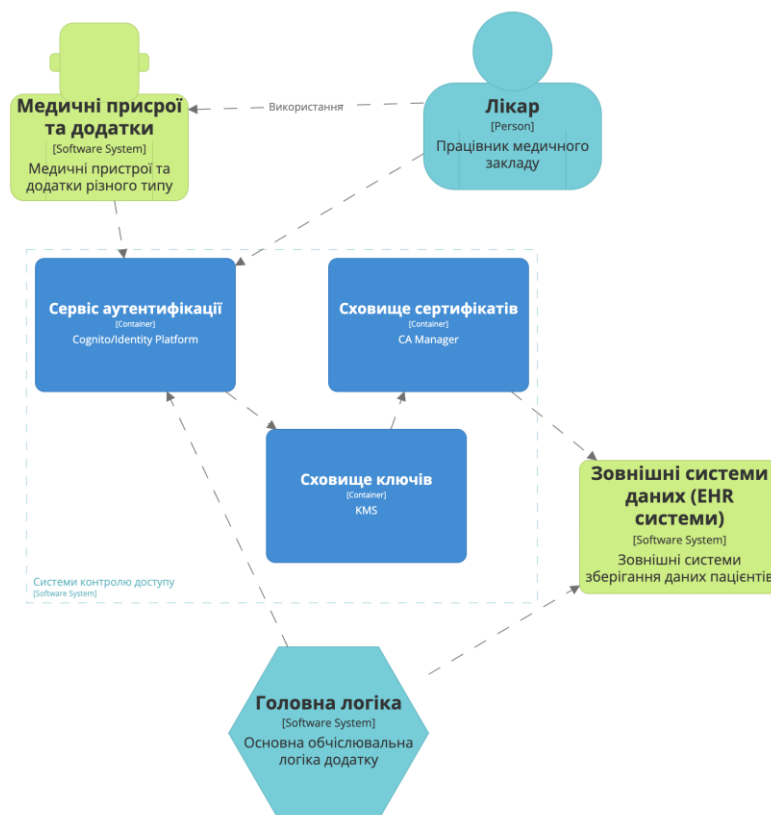


Рисунок 1.11 - Можлива інтеграція різних сервісів контролю доступу

Шифрування даних. Захист даних шляхом їх шифрування гарантує, що навіть якщо стороння особа отримає доступ до даних, вона не зможе зрозуміти інформацію без правильного розшифрування.

Забезпечення шифрування даних у медичних програмних комплексах є

важливим аспектом безпеки. Використання шифрування в передачі, в спокої, на рівні файлів та керування ключами гарантує конфіденційність медичних даних. Сучасні провайдери хмарних послуг пропонують готові сервіси для шифрування, такі як AWS KMS, Azure Key Vault, Google Cloud KMS та інші. Вони дозволяють легко керувати ключами шифрування та забезпечують високий рівень безпеки даних в медичних системах.

Аудит та моніторинг. Цей аспект включає в себе ведення журналів подій, аналіз активності користувачів та виявлення незвичайних або підозрілих дій. Це допомагає вчасно виявляти порушення безпеки.

Забезпечення аудиту та моніторингу в медичних програмних комплексах грає критичну роль у забезпеченні безпеки даних та виявленні можливих порушень. Забезпечення аудиту та моніторингу включає в себе інтеграцію з системами реєстрації подій, збереження та аналізу цих даних. Позиція компонентів моніторингу зазначена кольором на рисунку 1.12.

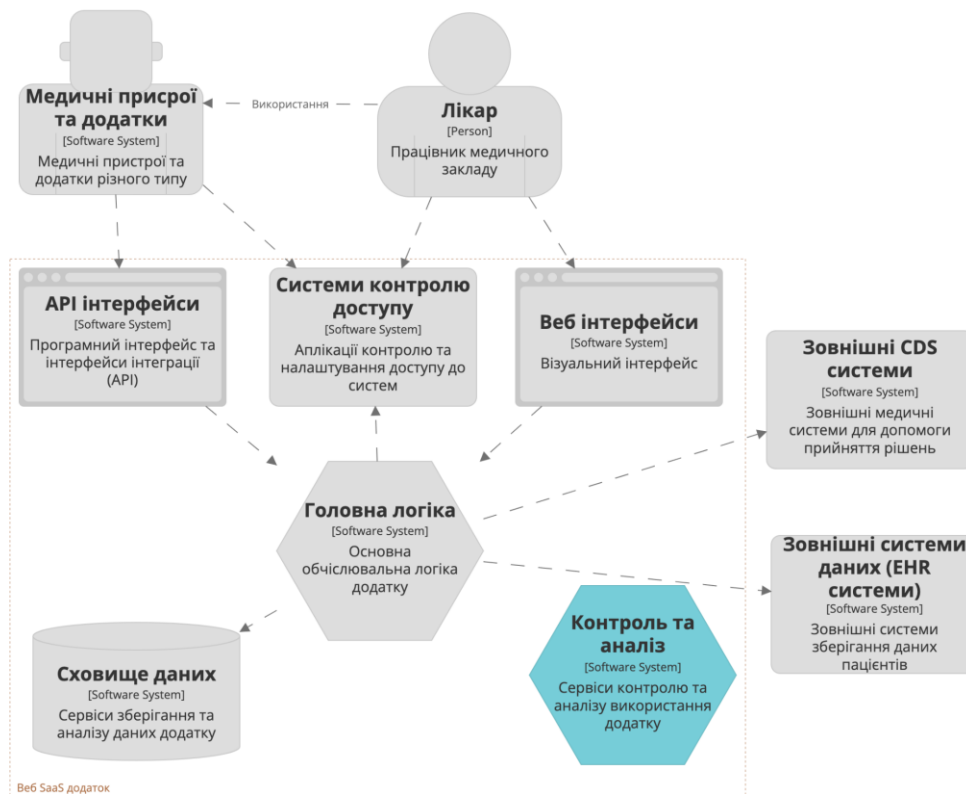


Рисунок 1.12 - Позиція сервісів контролю та аналізу у загальній архітектурі додатку

Сервіси аудиту і слідкування за діяльністю внутрішніх користувачів (адміністраторів, розробників) та ресурсів, такі як AWS CloudTrail, Google Cloud Audit Logging, Azure Activity Log та інші, відіграють важливу роль в сучасних хмарних середовищах. Такі сервіси дозволяють докладно вивчати події, пов'язані з користувачами, ресурсами та API, що використовуються в хмарному середовищі та є важливим інструментом для забезпечення безпеки та відповідності стандартам, а також для виконання аудитів діяльності. Дані з таких сервісів легко інтегруються з іншими сервісами та інструментами для подальшого аналізу та обробки подій.

Резервне копіювання та відновлення. Забезпечення можливості резервного копіювання та відновлення даних допомагає уникнути втрати інформації в разі аварій або катастроф.

Хмарні платформи, такі як Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, IBM Cloud та інші, пропонують послуги автоматичного резервного копіювання, які дозволяють регулярно створювати резервні копії даних та зберігати їх в надійних сховищах. Це зменшує ризик втрати даних в результаті технічних або природних катастроф.

Забезпечення надійного резервного копіювання та відновлення даних є ключовим аспектом забезпечення безпеки медичних програмних комплексів. Хмарні рішення надають широкий спектр інструментів та послуг для забезпечення надійності та доступності даних в умовах надзвичайних ситуацій.

1.5 Уніфіковані структури медичних даних

Медична інформація може бути дуже різноманітною і включати в себе дані про пацієнтів, їхні історії хвороб, результати обстежень, лікування та багато іншої інформації. Однак для ефективного обміну та аналізу даних у медичному секторі потрібна структурованість та стандартизація.

Стандарт FHIR (Fast Healthcare Interoperability Resources)[8] був створений організацією Health Level Seven International (HL7) з метою полегшити обмін

медичною інформацією між різними системами у сфері охорони здоров'я. Він був вперше опублікований у 2014 році і набув широкого визнання в галузі медичних інформаційних технологій завдяки своїй простоті та здатності створювати структуровані дані, що легко інтерпретуються та обмінюються між різними системами, сприяючи інтероперабельності та інтеграції медичних даних. Його основні переваги включають в себе легку розширюваність, використання стандартизованих REST API та підтримку різних медичних структур даних.

Багато сучасних EHR систем, такі як Epic Systems, Cerner, Allscripts, eClinicalWorks та інші, підтримують стандарт FHIR, для обміну медичною інформацією та інтеграції з іншими медичними системами.

Для створення та збереження структурованих медичних даних в форматі FHIR, а також для використання інтелектуальних аналітичних інструментів для отримання цінних інсайтів, хмарні провайдери пропонують готові рішення такі як AWS HealthLake, Google Cloud Healthcare API, Azure API for FHIR, IBM FHIR Server та інші.

1.6 Джерела медичних даних

Медична інформація оточує нас з усіх боків, і її джерела різноманітні. З розвитком технологій в медицині з'явилися нові джерела даних, які можуть значно збагатити інформацію, доступну для медичних систем. Розглянемо деякі з них:

- Смарт-пристрої. У сучасному світі пацієнти все частіше користуються смарт-пристроями, які можуть збирати медичні дані. Це можуть бути смартфони з датчиками для вимірювання пульсу, кров'яного тиску та інших параметрів, а також носимі пристрої, такі як фітнес-браслети і смарт-годинники;

- Медичні прилади. Медичні прилади, такі як ЕКГ-апарати, апарати для вимірювання глюкози та інші, надають деталізовані дані про стан пацієнта. Ці дані можуть бути надзвичайно важливими для діагностики та лікування;

- Лабораторні дослідження. Лабораторні дослідження надають важливу

інформацію про стан пацієнта, включаючи результати аналізів крові, сечі, аналізи тканин та інші;

- Медична інформація пацієнта. Самі пацієнти можуть надавати важливу інформацію про свій стан з використанням пацієнтських порталів і медичних додатків (включаючи EHR системи).

Позиція сервісів інтеграції зовнішніх даних в загальній логічній системі додатку зазначена на рисунку 1.13.

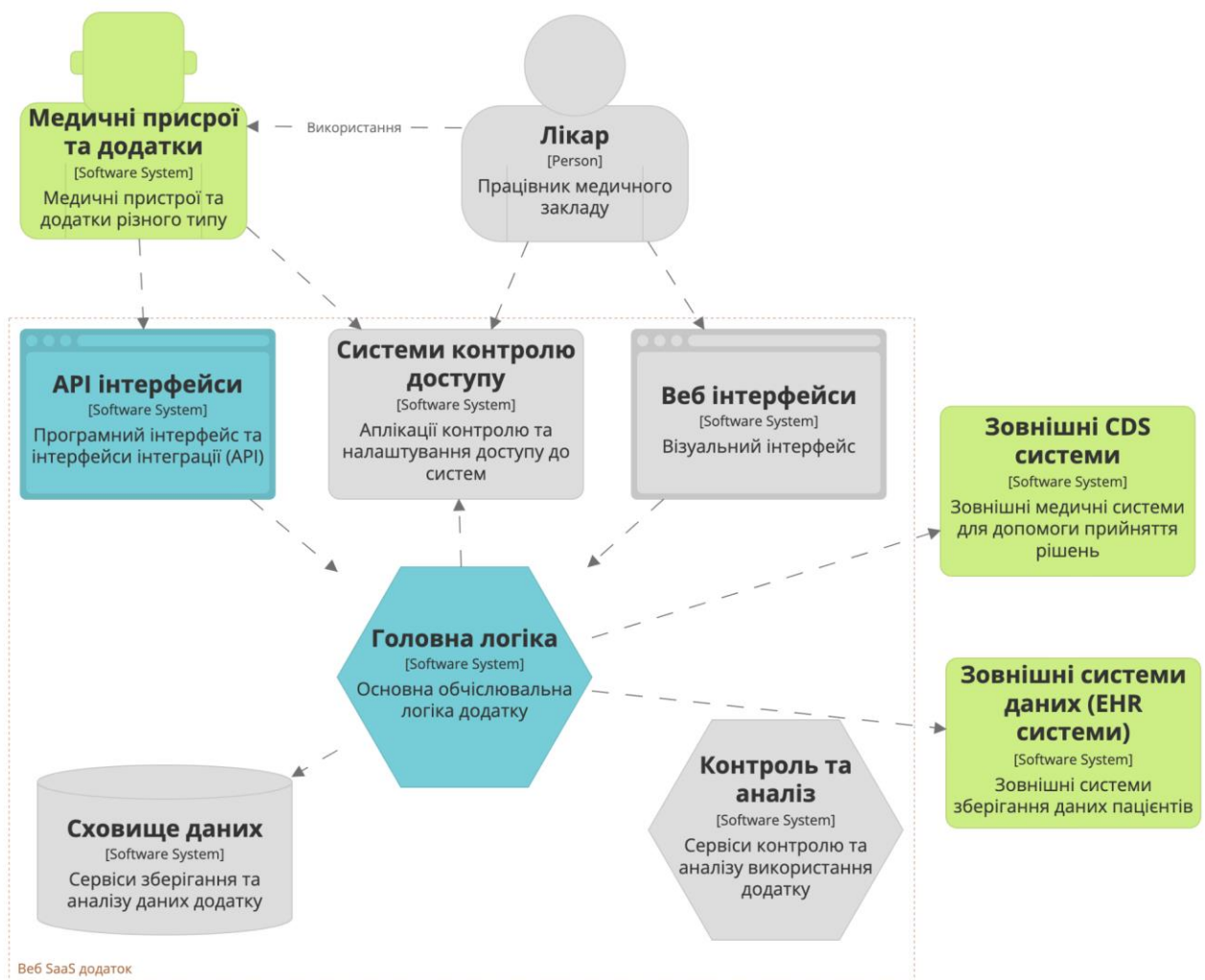


Рисунок 1.13 - Позиція сервісів інтеграції даних у загальній архітектурі додатку

Для збору та інтеграції медичних даних з різних джерел, хмарні провайдери надають велику кількість готових рішень:

- AWS IoT Core / Google Cloud IoT Core / Azure IoT Hub. Дані сервіси призначені для збору та обробки даних з IoT-пристроїв, включаючи медичні

пристрої та Смарт-пристрої;

- AWS Data Pipeline / Google Cloud Dataflow / Azure Data Factory. Ці сервіси можуть бути використані для виконання пакетної обробки даних за допомогою методу ETL - вилучення, перетворення, завантаження;

- AWS Kinesis / Azure Stream Analytics - інший тип хмарних сервісів, який може бути корисним для збору та обробки стрімінгових даних, включаючи медичні дані. Сервіси надають можливість обробки великої кількості подій в режимі реального часу та збору даних з різних джерел.

В загальній структурі додатку зазначений тип сервісів займає місце як це зазначено на рисунку 1.14.

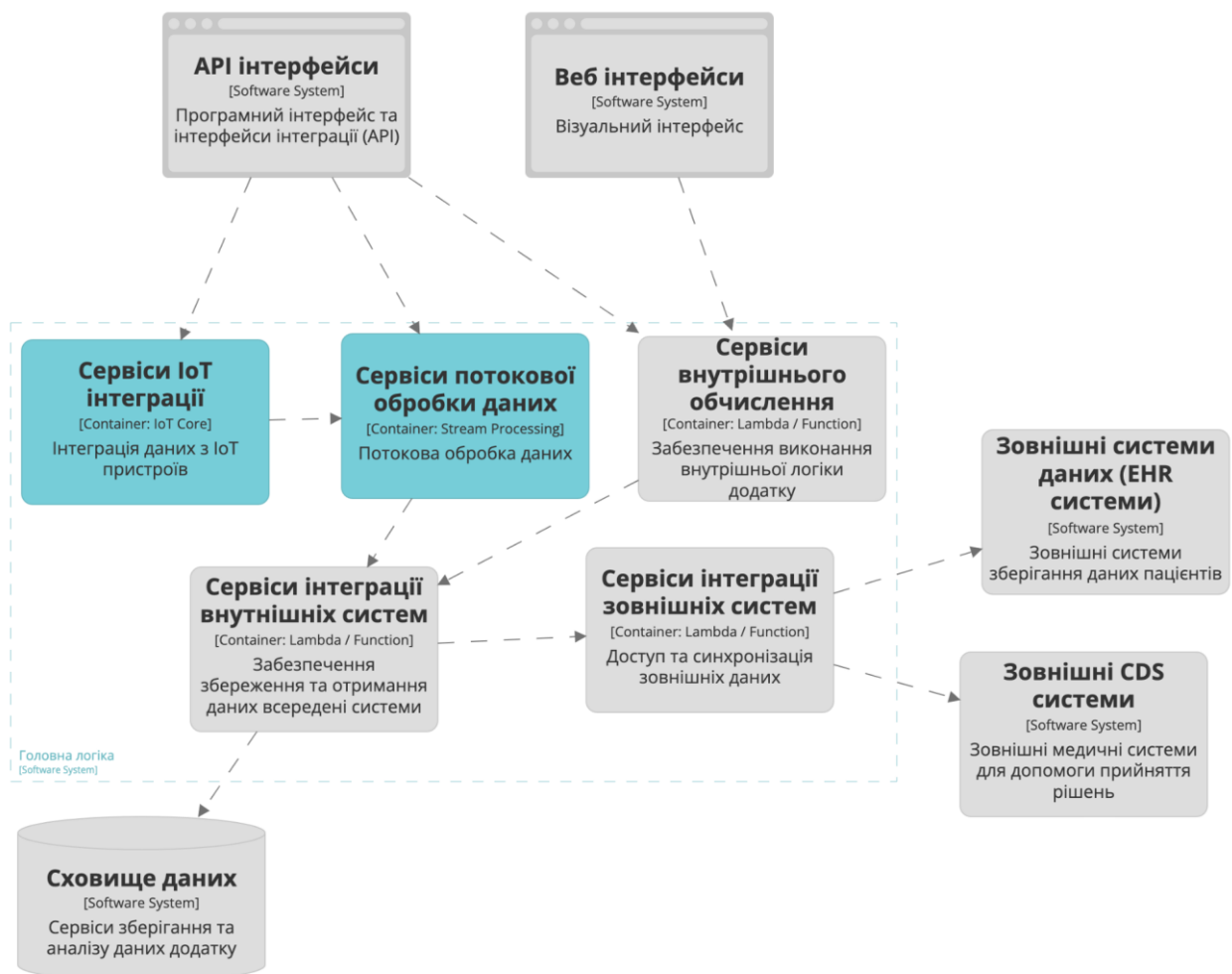


Рисунок 1.14 - Позиція сервісів потокової обробки даних в архітектурі сервісів головної логіки додатку

1.7 Аналіз медичних даних

Необхідність аналізу. Аналіз медичних даних може включати в себе дослідження та ідентифікацію кореляцій між різними параметрами здоров'я, прогнозування захворювань, моніторинг стану пацієнтів та багато іншого. Це допомагає лікарям та медичним фахівцям здійснювати більш точні діагнози та розробляти індивідуальні плани лікування.

Інструменти аналізу. Для аналізу медичних даних використовуються різні інструменти та методи, включаючи статистичний аналіз, машинне навчання, штучний інтелект та інші. Машинне навчання може допомогти виявити складні залежності між різними факторами та прогнозувати ризики. Наприклад, виявлення високого ризику серцевого нападу на підставі аналізу електрокардіограми або інших медичних параметрів.

Застосування результатів. Результати аналізу даних можуть бути використані для вдосконалення процесів у сфері медицини надзвичайних ситуацій. Наприклад, вони можуть допомогти планувати дії під час надзвичайних ситуацій, розподіляти ресурси, визначати стратегії евакуації та реагувати на ризики.

Вже згадані сервіси AWS HealthLake, Google Cloud Healthcare API, Azure API for FHIR, IBM FHIR Server дозволяють легко взаємодіяти з даними, використовувати інтелектуальний аналіз для виділення важливих інсайтів та застосовувати машинне навчання для прогнозування розвитку захворювань. Сервіси розрізняються деякою функціональністю в частині обробки зображень, додаткової безпеки, формату даних, конфігурації машинного навчання, але дуже схожі базовою функціональністю.

Хмарні сервіси аналізу даних інтегруються з хмарними сховищами, їх місце в логічній архітектурі зазначено на рисунку 1.15.

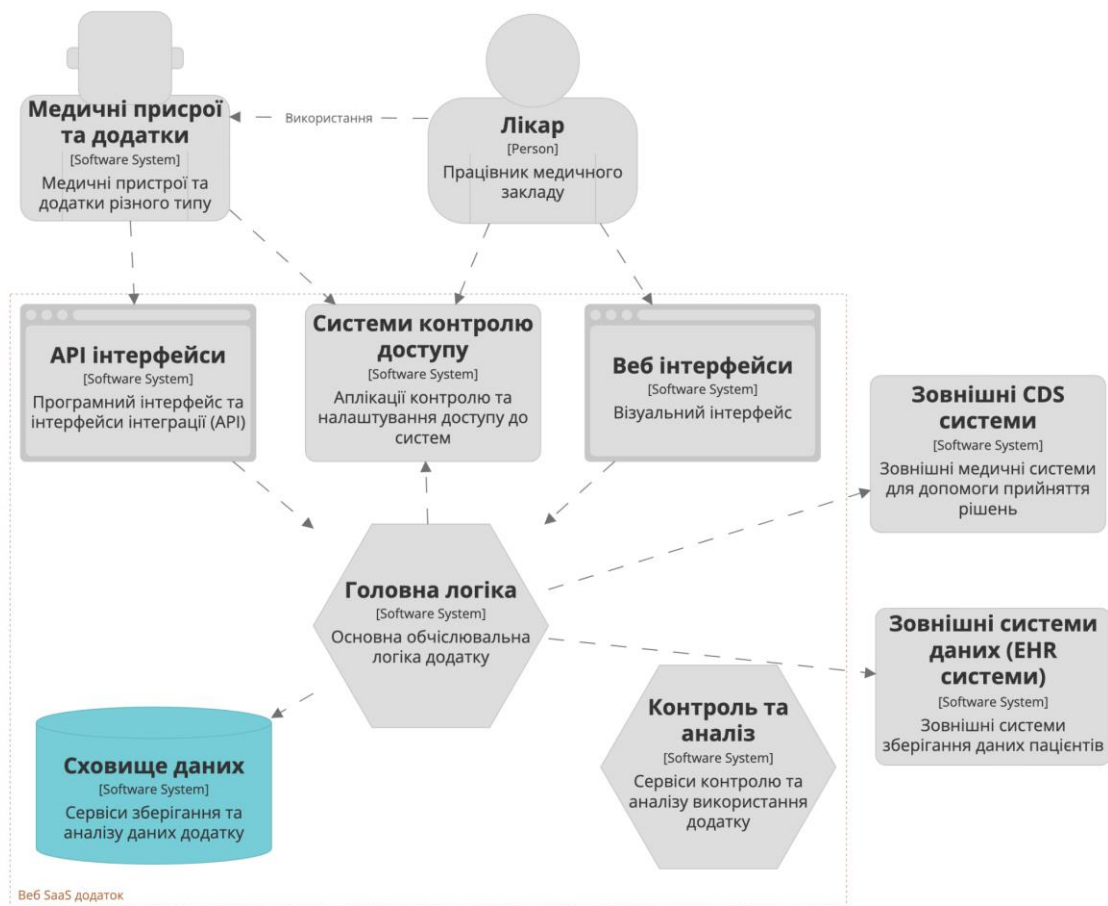


Рисунок 1.15 - Позиція сервісів зберігання та аналізу даних в загальній архітектурі додатку

Сервіси для розробки моделей штучного інтелекту, такі як AWS SageMaker, GCP AI Platform, Azure Machine Learning та інші, мають великий потенціал для застосування в галузі медицини та медицини надзвичайних ситуацій завдяки можливостям аналізу медичних зображень, прогнозування, створення персоналізованих моделей лікування, аналізу медичних досліджень та інших завдань.

За допомогою сервісів аналізу та візуалізації даних таких як AWS QuickSight, GCP Data Studio, Microsoft Power BI, для додатків доступна швидка та зручна візуалізація даних, в тому числі інтерактивна візуалізація, яка дозволяє користувачам взаємодіяти з даними та проводити аналіз у режимі реального часу.

Структуру взаємодії сервісів аналізу даних можна представити діаграмою,

як це зазначено на рисунку 1.16.

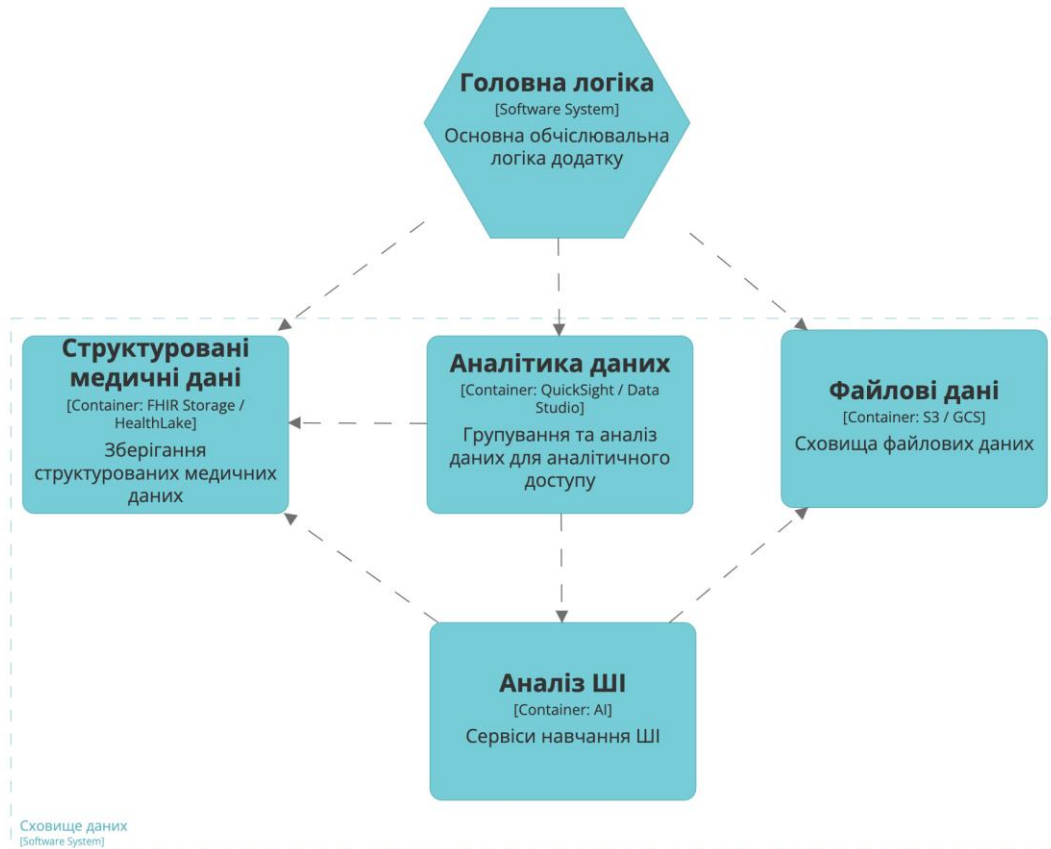


Рисунок 1.16 - Взаємодія сервісів зберігання та обробки даних

1.8 Доступність медичних систем

Ключовими аспектами доступності даних в сучасних системах є можливість віддаленого доступу та швидкості передачі даних з одночасним забезпеченням безпеки та реакції на відмову.

Хмарні провайдери пропонують різні можливості для високої доступності та забезпечення безпеки, розглянемо основні напрямлення сервісів:

- AWS CloudFront / Google Cloud CDN / Azure Content Delivery Network (CDN) - це сервіси CDN (мережі доставки контенту), які допомагають швидко доставляти вміст користувачам по всьому світу;
- AWS Lambda@Edge - це сервіс, який дозволяє виконувати код на рівні CloudFront CDN, тобто поблизу кінцевих користувачів;
- AWS Web Application Firewall (WAF) / Google Cloud Armor / Azure

Web Application Firewall - це сервіси для захисту веб-додатків від веб-загроз і атак. Вони дозволяють створювати правила фільтрації вхідного та вихідного трафіку для захисту веб-додатків від SQL-ін'єкцій, Cross-Site Scripting (XSS) і інших загроз.

Позиція сервісів, які забезпечують доступність зазначена на рисунку 1.17.

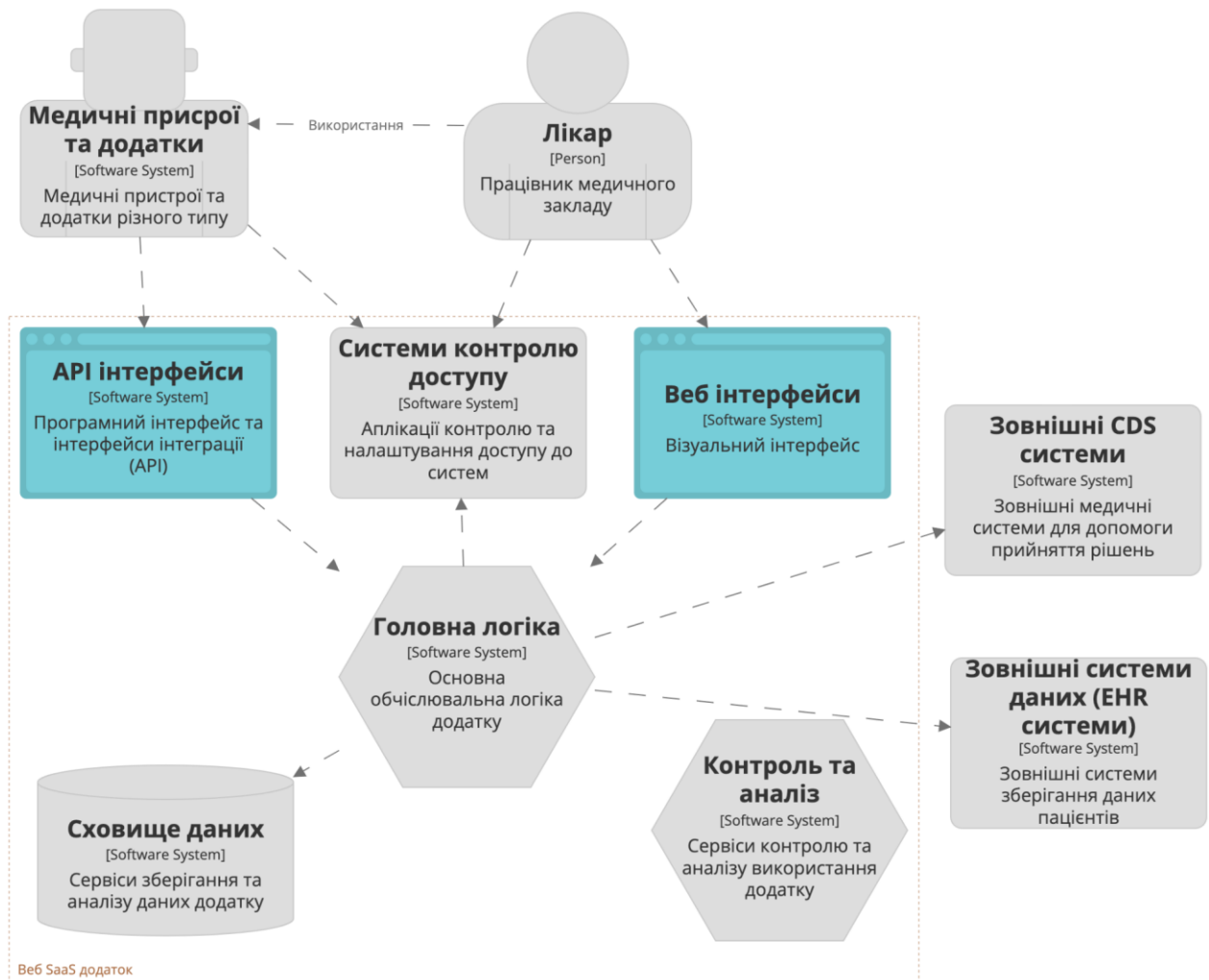


Рисунок 1.17 - Позиція сервісів доступу до даних в загальній архітектурі додатку

Кожен із цих сервісів має свої унікальні можливості та варіанти налаштувань. Вибір конкретного сервісу може залежати від потреб проекту, конкретних вимог до швидкості та безпеки.

1.9. Сервіси хмарного обчислення та інтеграції всередині системи

Сучасні медичні додатки потребують великої кількості обчислень для аналізу даних, формування звітів та надання допомоги пацієнтам. Використання безсерверних рішень для обчислень, таких як AWS Lambda, GCP Cloud Functions, Azure Functions та інші, дозволяє ефективно реагувати на зміни обсягу обчислень та реалізувати інтелектуальні сервіси на основі потреб користувачів.

Застосування безсерверних обчислень у медичних додатках допомагає покращити продуктивність, ефективність та розширюваність системи, зменшити витрати на обслуговування та зробити програмне забезпечення більш готовим до масштабування.

Сервіси безсерверних обчислень позначені кольором на рисунку 1.18.

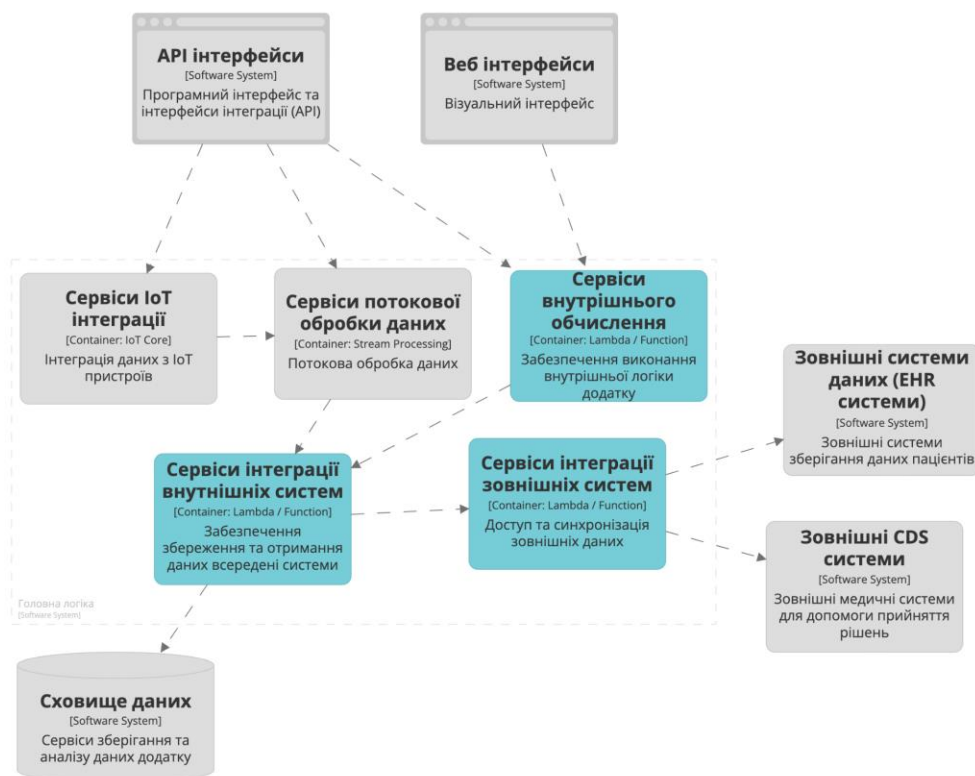


Рисунок 1.18 - Позиція сервісів обчислення та інтеграції даних в архітектурі сервісів головної логіки додатку

Сервіси безсерверного обчислення надають значні переваги у розробці програмного забезпечення серед яких: зручна інтеграція всередині хмари, обчислення в реальному часі, автоматичне масштабування, зменшення витрат на обслуговування.

Виходячи із зазначеного, безсерверні обчислення стають ключовим компонентом для реалізації медичних додатків з високою ефективністю, безпекою та можливістю швидкого реагування на потреби пацієнтів. Вони спрощують інтеграцію з різними сервісами та надають нові можливості для розробки медичного програмного забезпечення в сучасному хмарному середовищі.

1.10. Розгортання систем

Розгортання медичних інформаційних систем є критично важливою частиною процесу створення та експлуатації цих систем. Вимоги до швидкості, надійності та масштабованості вимагають нових підходів до розгортання.

Одним з ключових підходів до розгортання систем є "інфраструктура як код" (Infrastructure as Code або IaC). Цей підхід дозволяє автоматизувати процес розгортання та керування інфраструктурою за допомогою коду. Він забезпечує швидкість та консистентність при розгортанні нових екземплярів системи та її компонентів.

Інструменти управління інфраструктурним кодом (IaC) в сучасних хмарних обчисленнях грають важливу роль у спрощенні розгортання та керування інфраструктурою. AWS CloudFormation і Terraform є двома з таких інструментів, які надають можливість автоматизувати процес створення та налаштування інфраструктури.

AWS CloudFormation визначається своєю здатністю легко розгортати всю необхідну інфраструктуру на хмарних платформах AWS за допомогою шаблонів. Ця функціональність значно спрощує процес розгортання та забезпечує стандартизований підхід до управління ресурсами.

Terraform, натомість, виділяється своєю гнучкістю і можливістю підтримувати різні хмарні платформи та інші провайдери завдяки підтримці налаштуванню додаткових провайдерів. Це дозволяє розробникам та інженерам працювати з багатьма різними середовищами інфраструктури, не обмежуючи

себе однією хмарною платформою.

1.11 Висновки щодо використання хмарних та безсерверних рішень в медицині

На завершення, важливо підкреслити важливість та перспективність використання хмарних та безсерверних рішень у медичній сфері. Сучасний світ надає дедалі більше можливостей для збору, аналізу та обміну медичними даними завдяки цим технологіям. Використання EMR та EHR систем, підтримуваних хмарними та безсерверними архітектурами, дозволяє швидко та ефективно використовувати дані для надання медичної допомоги в критичних ситуаціях.

Основні переваги включають високу доступність даних, безпеку та відповідність регуляторним вимогам, стандартизацію та уніфікацію даних завдяки стандарту FHIR, можливість інтеграції з різними джерелами медичної інформації, швидкий аналіз великих обсягів даних та автоматизацію розгортання систем.

Хмарні та безсерверні рішення відкривають перед медичною галуззю нові можливості та забезпечують їй зручну та високоефективну інфраструктуру. Вони допомагають прискорити надання медичної допомоги та забезпечують надійність та безпеку даних. Сучасні медичні системи повинні використовувати ці рішення для покращення якості та швидкості надання медичної допомоги в критичних обставинах, що робить їх невід'ємною частиною майбутнього медичного програмного забезпечення.

1.12 Інформаційні системи медицини надзвичайних ситуацій

Інформаційні системи, спеціально розроблені для медицини надзвичайних ситуацій, визначають новий стандарт в ефективному управлінні та координації надзвичайних подій у медичному середовищі. Їхню важливість важко переоцінити, оскільки вони стають основним інструментом для надання

невідкладної медичної допомоги та управління ресурсами в умовах кризи.

В умовах надзвичайних ситуацій, де кожна секунда має значення, інформаційні системи стають основним механізмом для забезпечення доступності та обміну критичною медичною інформацією. Ці системи не просто автоматизують поточні медичні процеси, але і надають можливість миттєвого реагування на надзвичайні ситуації, забезпечуючи координацію та співпрацю між різними медичними структурами.

Важливо зауважити, що розглянуті інформаційні системи не є просто окремими елементами медичних технологій. Вони є невід'ємною частиною комплексного підходу до управління надзвичайними ситуаціями в медицині. У контексті загальної медичної інфраструктури вони вирізняються своєю спроможністю оперативно реагувати на зростаючі потреби та обмеження, забезпечуючи необхідні дані та інструменти для прийняття виважених медичних рішень.

Наступний розділ глибше розгляне конкретні інформаційні системи та сервіси, що знаходять своє застосування в контексті медицини надзвичайних ситуацій, підкреслюючи їхню роль та внесок у надзвичайні медичні сценарії.

2 РОЗРОБКА АРХІТЕКТУРИ ДЛЯ ПІДПРИЄМСТВ МЕДИЦИНИ НАДЗВИЧАЙНИХ СИТУАЦІЙ

2.1 Опис компанії для аналізу

У цій роботі розглядається розробка архітектури програмного забезпечення для компаній, що спеціалізуються на наданні послуг медицини надзвичайних ситуацій (МНС) на міжнародному рівні. Такі компанії покликані забезпечувати швидкий та ефективний доступ до невідкладної медичної допомоги, сприяти медичним закладам та рятувальним службам у вирішенні наслідків стихійних лих, терактів та інших надзвичайних ситуацій, а також підвищувати якість надання медичних послуг.

Для візуалізації загального бізнес процесу буде використано рисунок 2.1. запропонований Western Cape Government у своєму дослідженні архітектурної пропозиції [11].

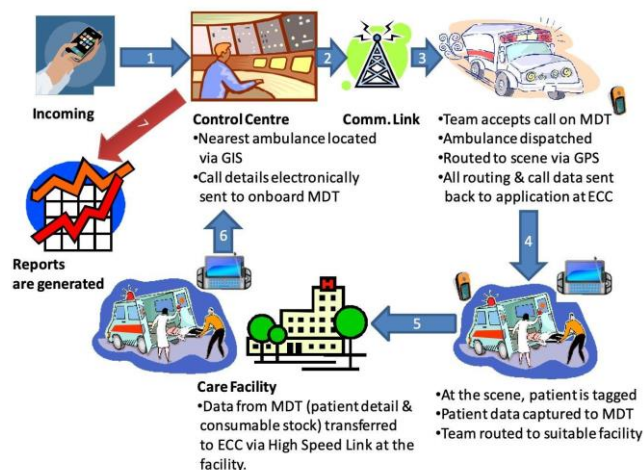


Рисунок 2.1 - Загальна візуалізація бізнес процесу

Бізнес-функції компаній зазвичай включають: Прямий виклик швидкої допомоги, Укладання договорів на обслуговування парку автомобілів швидкої допомоги, Надання послуг під час надзвичайних ситуацій, Планування та логістика ресурсів, Моніторинг стану пацієнтів, Медичну документацію, Консультації та навчання.

З точки зору програмного забезпечення вимагається наступне:

- Швидке розгортання в нових регіонах. Забезпечення можливості швидкого впровадження програмного комплексу в нових регіонах діяльності з мінімальними затратами на налаштування.

- Інтеграція з різними системами. Підтримка інтеграції з різними системами медичних даних, навігації, медичних пристроїв та інших сервісів для забезпечення обміну даними та координації дій.

- Безпекова забезпеченість. Дотримання найвищих стандартів безпеки даних, для захисту конфіденційної медичної інформації.

- Аналітика даних. Обробка великих обсягів даних та використання їх для передбачення та оптимізації надання медичних послуг.

- Масштабованість. Забезпечення можливості розширення системи з урахуванням зростання кількості підприємств та обсягу даних.

- Гнучкість і адаптивність. Здатність адаптуватися до змін у регуляторних вимогах та технологічних змін на ринку.

- Забезпечення високої доступності. Запобігання відмовам та перервам в роботі системи шляхом резервування та резервних копій даних.

2.2 Контекст технічного завдання

Загальною методологією аналізу у рамках цієї роботи буде використовуватися метод Метод аналізу компромісів архітектури (The Architecture Tradeoff Analysis Method - ATAM) [12] для аналізу та обрання архітектурних рішень, а також Модель архітектурного представлення "С4" [13].

Дана робота передбачає аналіз та використання постачальників хмарних послуг. Для цілей цієї роботи обрано Amazon Web Services згідно рішення, що наведено в додатку А.1.

З точки зору законодавчих вимог до безпеки та обробки даних пацієнтів, у цій роботі встановлено обмеження на ринки США та Європи. Таким чином зосередимося на вимогах HIPAA та GDPR. Глибокі деталі процедурних вимог, а

також детальні різниці в європейському та американському законодавстві не є предметом даного дослідження тому для загального розуміння обраний перелік перевірки, запропонований The HIPAA Journal [14]. Також додатково слід звернути увагу на те, що AWS має власні рекомендації щодо виконання вимог законодавства і у даному дослідженні будуть переважно використані сервіси, що можуть відповідати вимогам при коректному конфігуруванні [15].

2.3 Опис технічного завдання

Для коректного формування вимог проекту слід почати з визначення ключових зацікавлених сторін.

Рекомендоване [12] визначення зацікавлених сторін передбачає їх персоналізацію, тобто можливості для контактів (ім'я, контактна інформація), однак для цілей цієї роботи дані будуть виключені як зазначено в таблиці 2.1.

Таблиця 2.1 - Ключові зацікавлені сторони проекту

Відношення до проекту	Роль	Група	Можливі заперечення	Необхідні представлення
Зовнішній	Пацієнт	Клієнти	Доступність послуг, простота використання, приватність даних	Посібник користувача
Зовнішній	Партнери	Зовнішні партнери	Сумісність, інтеграція, ефективність співпраці	Посібник користувача
Зовнішній	Медичний персонал	Клієнти	Ефективність інструментів, інтеграція з існуючими системами	Посібник користувача
Замовник	ІТ-адміністратор	Технічна підтримка	Безпека даних, надійність системи, підтримка	Посібник адміністратора
Замовник	Керівник компанії	Управління	ROI, стратегічний розвиток, відповідність законодавству	Структура робіт, План розробки, Діаграми безпеки
Замовник	Менеджер проекту	Управління	Кошторис, терміни розробки, залежності команд, ризики	Структура робіт, План розробки, Контекст розробки, Залежності

Продовження таблиці 2.1

Відношення до проекту	Роль	Група	Можливі заперечення	Необхідні представлення
Розробник	Архітектор ПЗ	Розробка	Вимоги, обмеження, архітектурні рішення	Архітектура системи, усі види документації
Розробник	Керівник команди шару сервісів, презентації, підтримки	Розробка	План розробки, критичні дати/події	Архітектура системи, діаграми контексту/розгортання/послідовності/розробки
Розробник	Керівник команди контролю якості	Розробка	Вимоги, обмеження, план розробки, критичні дати/події	Посібник адміністратора/користувача
Замовник	Менеджер з безпеки	Управління	Конфіденційність, цілісність, доступність даних	Системи безпеки, політики і процедури
Зовнішній	Законодавчі органи	Регулювання	Відповідність законодавчим нормам, ліцензування	Звіти про відповідність, документація

Для забезпечення ефективності процесу аналізу, важливо, щоб усі учасники цього процесу мали чітке розуміння системи, яка підлягає такому аналізу. Рушії процесу зазначені в таблиці 2.2.

Таблиця 2.2 - Загальні бізнес задачі проекту

Рушій	Ціль	Задача
Покращення доступності та якості невідкладної медичної допомоги	Оптимізація логістики швидкої допомоги	Розробка ефективної системи для координації транспортних засобів та медичного персоналу
		Інтеграція з системами GPS та медичними даними для швидкого реагування
	Забезпечення високої доступності та надійності системи	Розробка архітектури з високою ступеню резервування та масштабованості
		Впровадження механізмів автоматичного відновлення системи та резервного копіювання даних
Забезпечення безпеки даних		Розробка та імплементація політик і процедур безпеки даних

Продовження таблиці 2.2

Рушій	Ціль	Задача
		Впровадження шифрування даних та контролю доступу до медичної інформації
	Підвищення якості медичних послуг	Розробка інтерфейсів для моніторингу стану пацієнтів
		Інтеграція з медичними базами даних для поліпшення діагностики та лікування

Перед початком розробки архітектури будь-якого проекту, критично важливим є чітке визначення та аналіз ключових вимог, які впливають на її формування. Ці вимоги підрозділяються на функціональні, які описують конкретні функції та поведінку системи, та нефункціональні, які стосуються загальних характеристик системи, таких як надійність, продуктивність, та масштабованість. Окрім цього, архітектурні обмеження, які є важливою складовою нефункціональних вимог, мають значний вплив на вибір та розробку архітектурних рішень. Ці обмеження вже були розглянуті у пункті 2.2, де було висвітлено їх роль у контексті загального дизайну системи. У таблиці 2.3. розглянуто переліку вимог, які стануть основою для подальшої роботи над архітектурою проекту, забезпечуючи її відповідність поставленим цілям та потребам користувачів.

Таблиця 2.3 - Архітектурно важливі вимоги проекту

Вимога	Тип	Опис
Централізована аутентифікація і авторизація	Функціональна	Розробка уніфікованої системи аутентифікації та авторизації, яка дозволить користувачам входити в систему та отримувати доступ до відповідних ресурсів в залежності від їх ролей і прав доступу.
Інтеграція з зовнішніми системами медичних даних (EHR)	Функціональна	Інтеграція з системами електронних медичних записів для обміну даними пацієнтів, що дозволить забезпечити координацію догляду та підтримку клінічних процесів.

Продовження таблиці 2.3

Вимога	Тип	Опис
Обробка платежів	Функціональна	Інтеграція платіжної системи для безпечної та надійної обробки фінансових транзакцій всередині веб-додатку, включаючи функції обробки платежів, виставлення рахунків та управління фінансовими звітами.
Потокова обробка даних	Функціональна	Створення механізму для надходження та аналізу потоків даних у режимі реального часу, що дозволить миттєво реагувати на зміни в даних та оптимізувати процеси прийняття рішень.
Система підтримки клінічних рішень (CDS)	Функціональна	Інтеграція інструментарію для аналізу медичних даних з метою підтримки медичного персоналу у прийнятті обґрунтованих клінічних рішень, забезпечуючи високу якість догляду за пацієнтами.
Модуль інтеграції з медичними пристроями	Функціональна	Розробка інтерфейсів для інтеграції з різноманітними медичними пристроями, які дозволять збирати важливі медичні дані, покращуючи моніторинг стану пацієнтів та ефективність медичного обслуговування.
Розширювана архітектура інтеграції	Функціональна	Забезпечення можливості легкої інтеграції нових зовнішніх систем та сервісів через модульну архітектуру, що дозволить швидко адаптувати систему до змінюваних потреб ринку без великих затрат на розробку.
Аналітика та звітність	Функціональна	Розробка комплексного інструментарію для аналітики даних, що дозволить відслідковувати показники використання сервісів, визначати тренди та складати звіти для підтримки управлінських рішень.
Керування даними та їхньою якістю	Функціональна	Створення системи керування даними, яка включає процеси збору, очищення, валідації та зберігання даних, забезпечуючи їх актуальність, консистентність і точність.
Безпека даних	Функціональна	Реалізація політик безпеки даних з використанням найкращих практик і стандартів, включаючи шифрування, контроль доступу та аудит, щоб відповідати вимогам приватності та захисту даних згідно з HIPAA, GDPR та іншими нормативними актами.
Персональна сторінка пацієнта	Функціональна	Розробка інтерфейсу для пацієнтів, що дозволить їм переглядати особисту медичну інформацію, історію звернень, результати обстежень, а також управління своїми медичними записами та зверненнями до лікарів.

Продовження таблиці 2.3

Вимога	Тип	Опис
--------	-----	------

Вимога	Тип	Опис
Сторінка партнера (замовника послуг)	Функціональна	Створення порталу для партнерів, що надасть медичним закладам, страховим компаніям та іншим замовникам послуг доступ до інформації про свої замовлення, статуси та управління контрактами.
Сумісність	Нефункціональна	Система має бути сумісною з різноманітними медичними системами та стандартами обміну даними, які існують у медичній індустрії.
Швидкий доступ до даних	Нефункціональна	Система повинна забезпечувати доступ до критичних даних протягом декількох секунд.
Масштабованість	Нефункціональна	Архітектура має бути гнучкою, щоб підтримувати масштабування без втрати продуктивності та доступності.
Висока доступність	Нефункціональна	Система повинна мати високий рівень доступності та мінімальний час простою.
Захист конфіденційної інформації	Нефункціональна	Необхідно забезпечити суворі заходи захисту персональних та медичних даних.
Продуктивність	Нефункціональна	Система повинна обробляти запити користувачів та інтеграції з зовнішніми сервісами з мінімальною затримкою, підтримуючи високу продуктивність навіть за умови великої кількості одночасних запитів.
Адаптивність до змін	Нефункціональна	Система повинна бути здатною адаптуватися до швидких змін у медичному середовищі, включаючи впровадження нових технологічних рішень, медичних протоколів та лікувальних практик.
Автоматизоване резервне копіювання	Нефункціональна	Система повинна мати вбудовані механізми автоматичного резервного копіювання даних.
Адаптивність до змін	Нефункціональна	Система повинна бути здатною адаптуватися до швидких змін у медичному середовищі, включаючи впровадження нових технологічних рішень, медичних протоколів та лікувальних практик.
Масштабованість інтеграцій	Нефункціональна	Інтеграції з зовнішніми системами повинні підтримувати легке масштабування для адаптації до змінюваного навантаження без потреби в перепроєктуванні.

На наступному етапі аналізу здійснюється важливе завдання встановлення пріоритетів та вдосконалення ключових аспектів якості системи. Це вимагає

сконцентрувати увагу на тих елементах архітектури, які визначаються як найбільш суттєві для забезпечення ефективності та надійності системи. Завдяки виявленню та аналізу специфічних вимог проекту, можна визначити та систематизувати основні атрибути якості, формуючи таким чином дерево атрибутів якості.

Основні атрибути якості системи, а також пріоритизація таких атрибутів за важливістю для бізнесу та впливом на архітектуру зазначені в таблиці 2.4. Шкала пріоритизації - В - велика, С - середня, Н - низька.

Таблиця 2.4 - Атрибути якості системи

Атрибут якості	Опис	Метрика	Архітектурний / Бізнес вплив
Ремонтопридатність (Обслуговуваність)	Можливість здійснення оновлень та змін без простою системи	Процес повторного перерозгортання системи (окремої частини) без простоїв	В / С
Продуктивність	Швидкість обробки та відгук системи на запити користувачів.	< 1 с середній час відповіді	В / В
Масштабованість	Здатність системи адаптуватися до зростання навантаження.	Система адаптована до автоматичного горизонтального масштабування	В / В
Надійність	Здатність системи функціонувати без збоїв.	MTTF > 1000 годин	В / В
Безпека	Захист даних від несанкціонованого доступу і втрат.	Використання як мінімум TLS 1.2.	Н / В
Легкість обслуговування	Простота внесення змін та оновлень у систему.	Час на деплоймент < 1 година	В / С
Інтероперабельність	Структура системи, що дозволяє легко змінювати частини взаємодії з іншими системами.	Час на розробку нового модуля взаємодії < 2 тижні	В / В

Продовження таблиці 2.4

Атрибут якості	Опис	Метрика	Архітектурний / Бізнес вплив
Доступність	Стабільний доступ до системи та швидке відновлення після збоїв.	99.9% доступність системи	В / В
Аудитопритатність	Управління та передача даних клієнтів повинні відповідати HIPAA / GDPR	Журнал аудиту фіксує 100% запитів на доступ до даних	Н / В

2.4 Концептуальна архітектура

Для початку роботи треба візуалізувати загальну архітектуру додатку, яка зазначена на рисунку 2.2. На зображенні представлена архітектурна діаграма системи, яка візуалізує компоненти та користувачів веб-додатку. Центральним елементом є "Сервіси додатку", який є основною логічною частиною системи. Він забезпечує доступу до внутрішньої логіки системи та зовнішніх компонентів і систем, включаючи зовнішню платіжну систему, систему навігації, системи даних EHR (Electronic Health Records) та допомоги прийняття рішень CDS (Clinical Decision Support). "Сервіси додатку" доступні користувачам за допомогою "Інтерфейсу доступу", який функціонує як шлюз між користувачами та основними сервісами.

До системи можуть підключатися різні типи користувачів: лікарі (які включають молодший медичний персонал), IT адміністратори, пацієнти, що використовують сервіс, та корпоративні клієнти, які можуть представляти медичні заклади, що мають угоди на обслуговування. Крім того, в архітектурі враховано інтеграцію з медичними приладами та додатками, а також внутрішні та зовнішні бази даних, що підкреслює потребу у сумісності та інтегрованості різних систем.

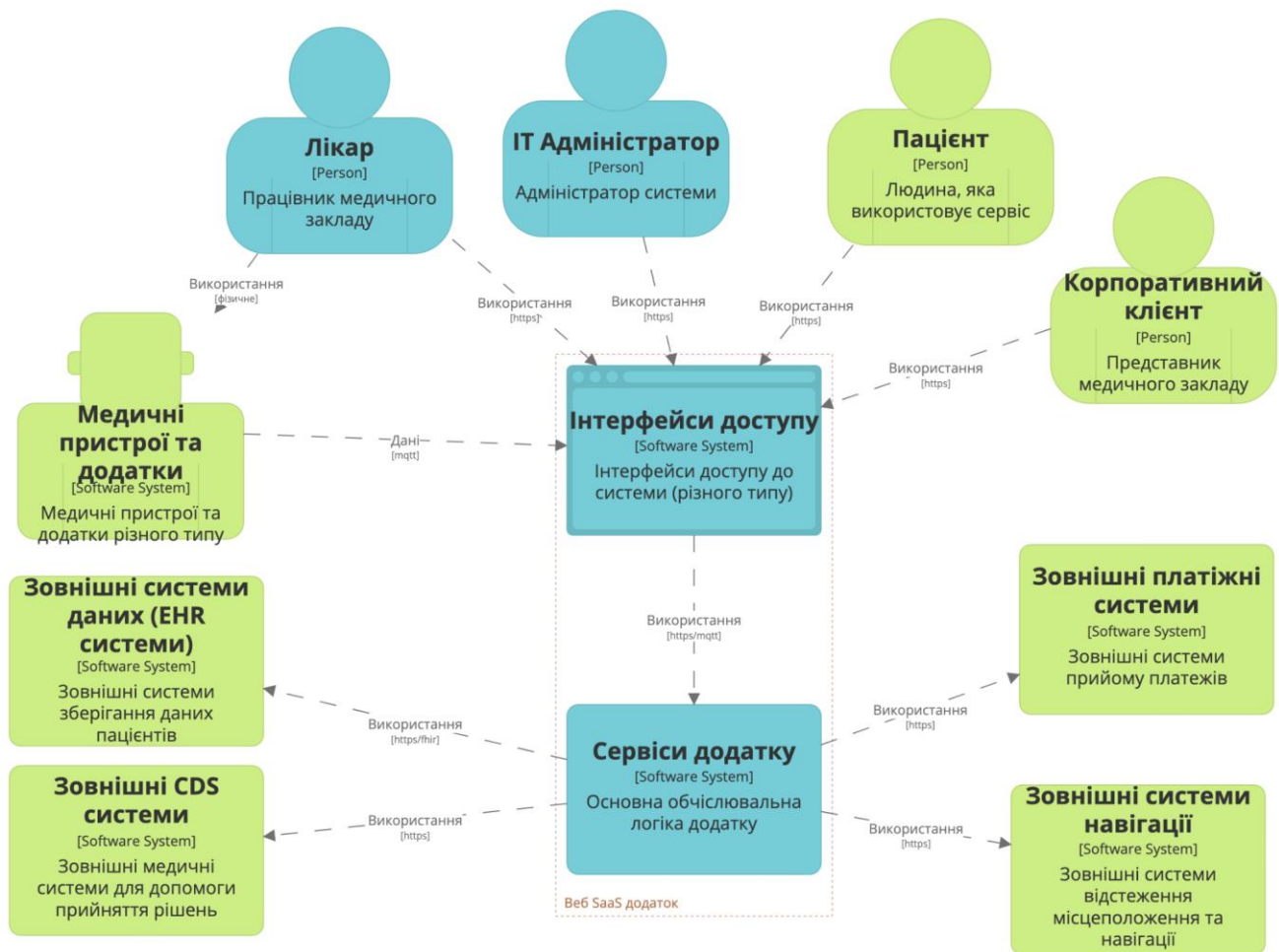


Рисунок 2.2 - Загальна архітектура додатку

Вся система побудована на моделі SaaS (Software as a Service), що забезпечує доступність сервісів через інтернет.

Ключові елементи діаграми наведені у таблиці 2.5.

Таблиця 2.5 - Ключові елементи загальної архітектури додатку

Компонент	Опис
Внутрішні користувачі - Лікар	Лікарі та молодший медичний персонал використовують систему для доступу до медичних записів, управління пацієнтами та іншими клінічними задачами.
Внутрішні користувачі - ІТ Адміністратор	ІТ Адміністратори забезпечують технічну підтримку системи, управління користувачами та безпекою.
Зовнішні користувачі - Пацієнт	Пацієнти використовують систему для отримання інформації про себе та своє здоров'я, можуть інтегрувати особисті пристрої контролю здоров'я.
Зовнішні користувачі - Корпоративний клієнт	Корпоративні клієнти, які можуть бути страховими компаніями або медзакладами, використовують систему для управління

Продовження таблиці 2.5.

Компонент	Опис
	договорами та послугами, формування звітів.
Медичні пристрої та додатки	Це обладнання та програмні рішення, які інтегруються з системою для збору та аналізу медичних даних.
Зовнішні системи - EHR	Електронні системи здоров'я, що зберігають медичні записи пацієнтів і інтегруються з додатком для доступу та обміну даними.
Зовнішні системи - CDS	Системи підтримки клінічних рішень, які надають рекомендації та інформацію для оптимізації лікування.
Зовнішні системи - Платіжні	Платформи, які обробляють фінансові транзакції, пов'язані з медичними послугами.
Зовнішні системи - Навігаційні	Системи, що використовуються для координації логістики, наприклад, для визначення місцеположення медичного обладнання або транспорту.
Інтерфейси доступу	Шлюзи, через які користувачі взаємодіють з сервісами додатку, включаючи аутентифікацію та авторизацію.
Сервіси додатку	Центральні бізнес-логіка та функціональність додатку, які надають основні послуги користувачам.

На основі функціональності, яку надає система, створено кілька точок доступу. Користувацькі портали розділені відповідно до їх функціональності. Такий підхід робить окремі додатки більш легковажними та спрощує контроль доступу як це пояснено в додатку А.2. Для досягнення спільного вигляду та поведінки можна використовувати бібліотеки компонентів. Це дозволяє створити уніфікований інтерфейс, забезпечуючи водночас гнучкість у конфігурації та зручність управління для різних груп користувачів.

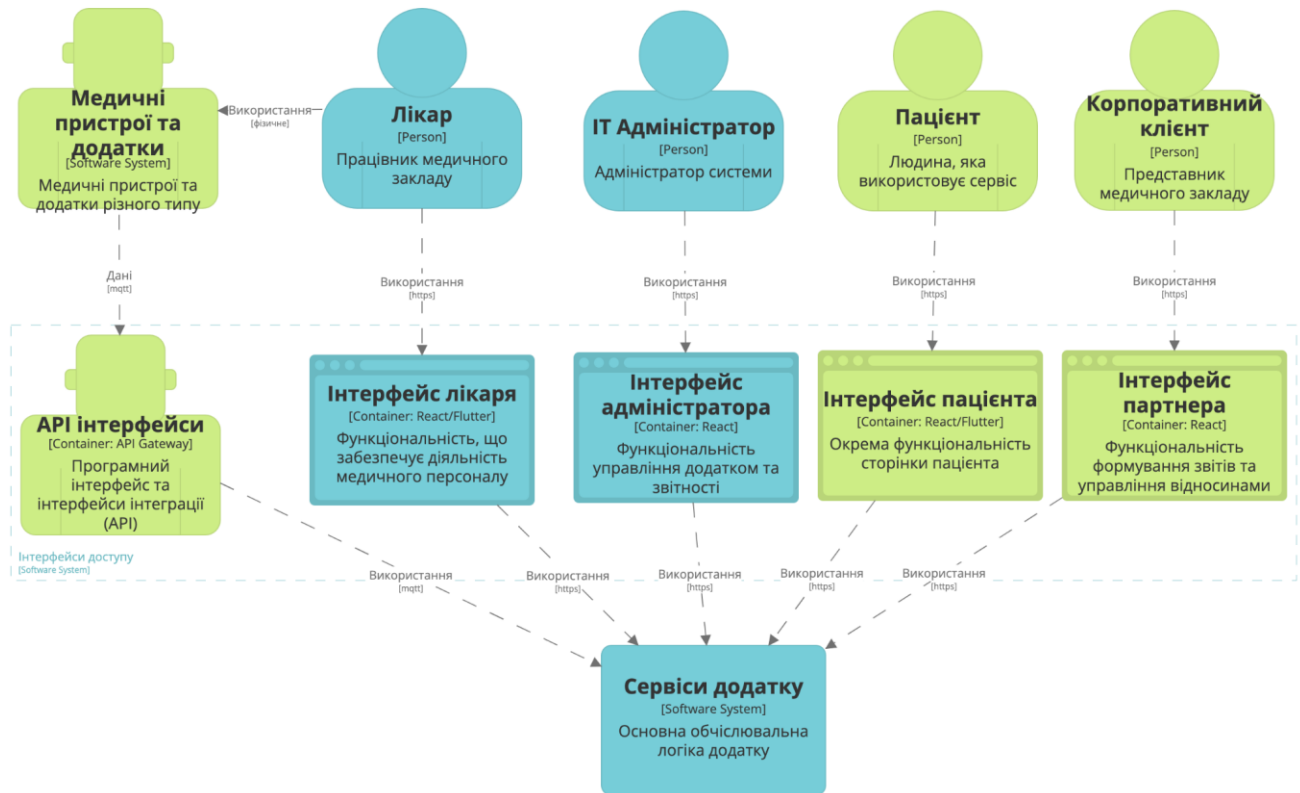


Рисунок 2.3 - Елементи інтерфейсів доступу до системи

Опис ключових елементів діаграми наведений в таблиці 2.6.

Таблиця 2.6 - Ключові елементи діаграми інтерфейсів доступу

Компонент	Опис
API Інтерфейси	Програмні інтерфейси, які дозволяють зовнішнім системам та сервісам інтегруватися з веб-додатком, забезпечуючи автоматизований обмін даними та функціональність.
Інтерфейс лікаря	Індивідуалізований інтерфейс для медичних фахівців, що надає доступ до медичних записів пацієнтів, інструментів для призначення лікування та моніторингу стану пацієнтів.
Інтерфейс адміністратора	Інтерфейс для управління користувачами, налаштуваннями системи та контролю за її безперервною роботою, призначений для ІТ спеціалістів або адміністративного персоналу.
Інтерфейс пацієнта	Особистий портал для пацієнтів, який дозволяє їм отримувати інформацію про своє здоров'я, історію лікування, взаємодіяти з медичними працівниками, формувати звіти та інтегрувати особисті медичні пристрої.
Інтерфейс партнера	Портал для корпоративних партнерів, таких як інші медичні установи та постачальники послуг, що надає функції управління контрактами, формування звітів та доступу до спільних медичних ресурсів.

На діаграмі, що наведена на рисунку 2.4 ілюструється внутрішня

архітектура "Сервісів додатку", яка є фундаментом для забезпечення взаємодії користувачів із центральними функціями веб-додатку. В основі цієї архітектури лежить множина взаємопов'язаних сервісів, кожен з яких виконує специфічну роль, сприяючи злагодженій роботі системи як єдиного механізму.

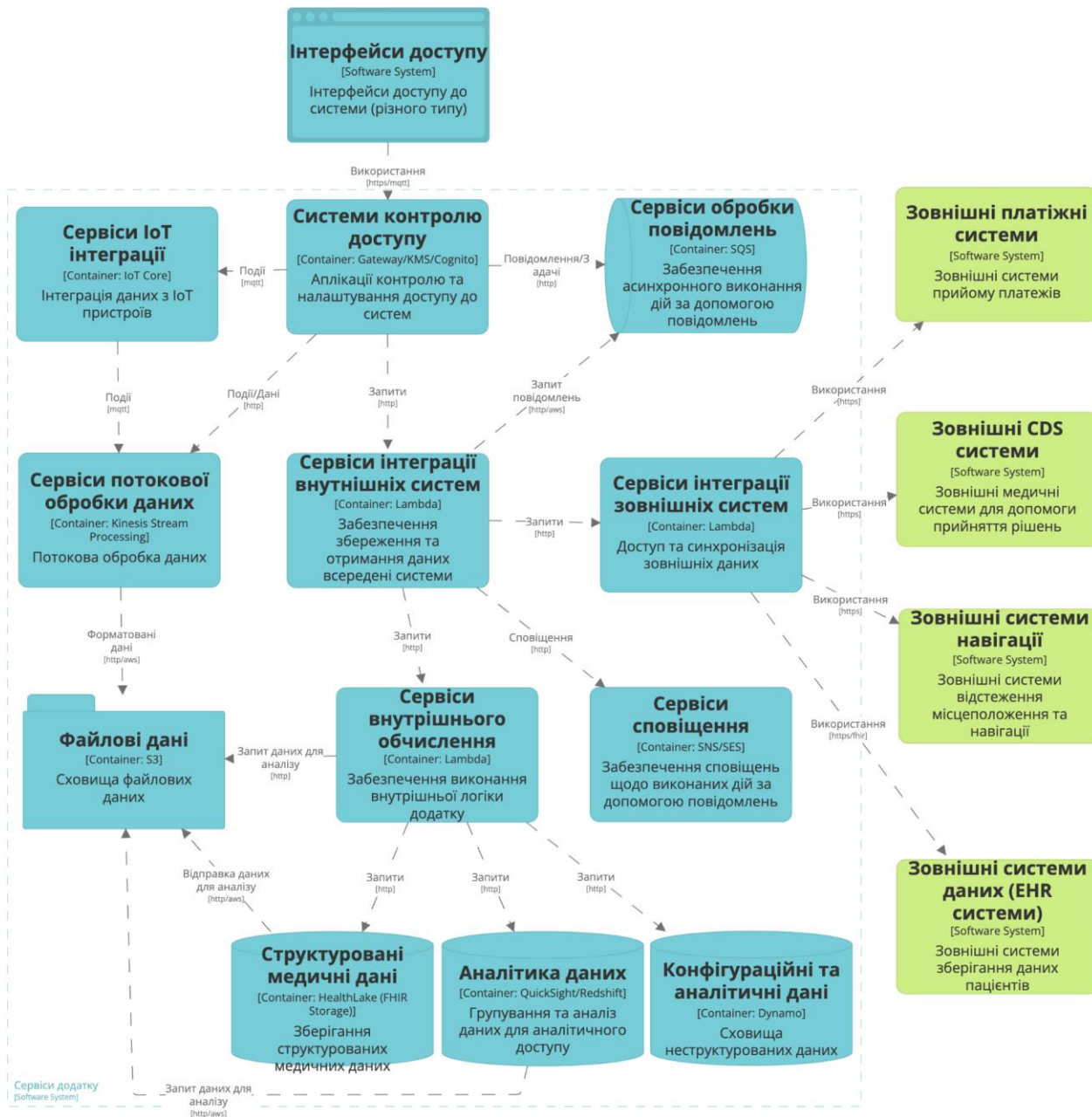


Рисунок 2.4 - Структура елементів сервісів додатку

Опис ключових елементів діаграми наведений в таблиці 2.7.

Таблиця 2.7 - Ключові елементи діаграми сервісів додатку

Компонент	Опис
Системи контролю доступу	Контролює доступ до різних частин системи, забезпечуючи безпеку та авторизацію. Зберігає ключі доступу до зовнішніх систем.
Сервіси IoT інтеграції	Відповідає за інтеграцію з інтернетом речей та медичними пристроями для збору та передачі даних пацієнтів.
Сервіси потокової обробки даних	Забезпечує обробку великих потоків даних у реальному часі для швидкого реагування на зміни в стані пацієнтів.
Сервіси інтеграції внутрішніх систем	Управляє взаємодією між різними внутрішніми модулями та сервісами додатку.
Сервіси інтеграції зовнішніх систем	Підключає додаток до зовнішніх систем, таких як EHR, CDS та інші, забезпечуючи обмін даними.
Сервіси обробки подій	Обробляє події всередині системи, такі як сповіщення користувачів та реакція на зміни.
Сервіси внутрішнього обчислення	Забезпечує підтримку внутрішньої інфраструктури та операційних процесів.
Сервіси сповіщення	Управляє розсилкою сповіщень та повідомлень користувачам системи.
Файлові дані	Сховище для зберігання та управління файловими даними, включаючи медичні зображення та документи.
Структуровані медичні дані	База даних для структурованого зберігання медичних записів та інформації.
Аналітика даних	Аналізує зібрані дані для підтримки клінічних рішень та управління операціями.
Конфігураційні та аналітичні дані	Зберігає налаштування параметрів системи та операційні дані додатку.

На представленій на рисунку 2.5 діаграмі інтеграцій ілюструється структура "Сервісів інтеграції зовнішніх систем" веб-додатку. Центральним компонентом цієї структури є оркестратор інтеграцій, який виступає як координатор між внутрішніми процесами додатку та зовнішніми системами. Рішення щодо створення оркестратора наведено в додатку А.3. Оркестратор запитів оптимізує взаємодію з зовнішніми системами, такими як зовнішні платіжні системи, системи допомоги клінічних рішень (CDS), навігаційні системи та електронні

медичні записи (EHR), спрощуючи процес інтеграції та забезпечуючи ефективність запитів.

В цілому, діаграма відображає стратегічний підхід до інтеграції зовнішніх систем, підкреслюючи важливість гнучкості, безпеки та ефективності у взаємодії з різноманітними зовнішніми сервісами та платформами, що є необхідними для сучасних медичних веб-додатків.

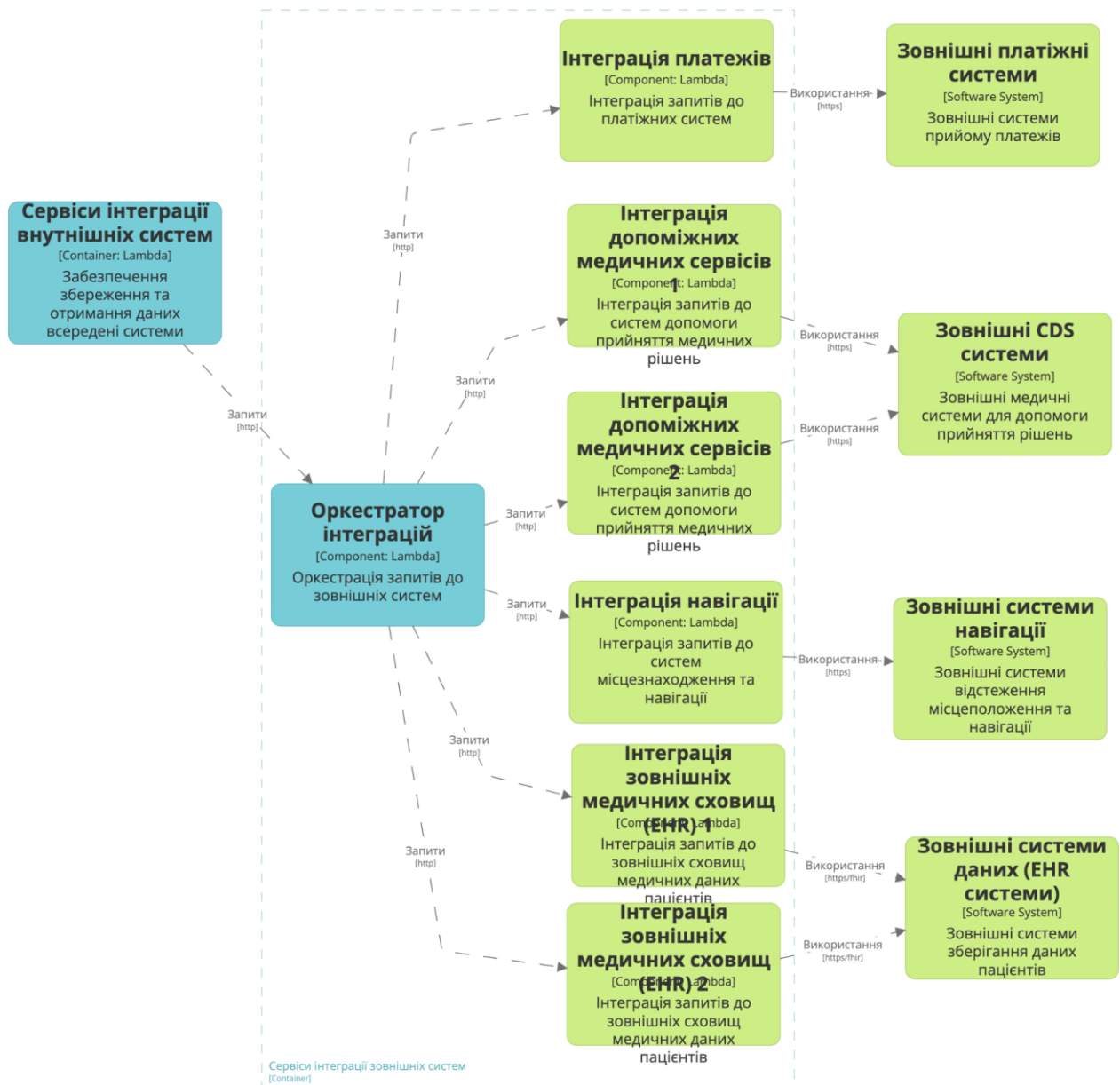


Рисунок 2.5 - Структура сервісів зовнішньої інтеграції

Опис ключових елементів діаграми наведений в таблиці 2.8.

Таблиця 2.8 - Ключові елементи діаграми сервісів зовнішньої інтеграції

Компонент	Опис
Оркестратор інтеграцій	Являє собою єдину точку доступу до зовнішніх систем. Забезпечує управління та оркестрацію запитів.
Інтеграція допоміжних медичних сервісів	Формує внутрішній інтерфейс взаємодії до зовнішніх допоміжних медичних сервісів
Інтеграція навігації	Формує внутрішній інтерфейс взаємодії із зовнішньою системою місцерозташування та навігації
Інтеграція зовнішніх медичних сховищ	Формує внутрішній інтерфейс взаємодії до зовнішніх сховищ медичних даних пацієнтів
Інтеграція платежів	Формує внутрішній інтерфейс взаємодії із зовнішньою платіжною системою

2.5. Ключові особливості взаємодії компонентів

Попередньо було розглянуто загальну структуру компонентів додатку та їх взаємодії. У даному підрозділі розглянуто кілька взаємодій (послідовностей викликів), які можуть потребувати додаткової уваги з огляду на загальну структуру взаємодії компонентів. Такий підхід дозволяє виокремити ключові взаємодії, що відіграють вирішальну роль у забезпеченні злагодженої роботи всіх компонентів, а також уникнути можливих проблем пов'язаних з масштабуванням і модифікацією системи.

Збір медичних даних. Для безперервного моніторингу стану пацієнтів їх медичні дані повинні бути передані у систему якомога швидше. Процес зберігання даних з медичних приладів та персональних медичних девайсів можна описати наступним чином [16]:

- Формування даних. Медичний працівник працює з медичними приладами, які мають вбудовані програмні інтерфейси взаємодії (SiMD);
- Збір даних. Медичні прилади та персональні медичні девайси (наприклад, монітори здоров'я, портативні датчики, розумні годинники) збирають різноманітні дані про стан здоров'я пацієнтів;
- Передача даних. Зібрані дані передаються через безпечні інтеграційні

шлюзи або API інтерфейси;

- Первинна обробка даних. Після отримання даних система може виконувати первинну обробку, таку як валідацію, нормалізацію та інші форми попереднього аналізу;

- Потокова обробка даних. Для даних, що потребують реального часу обробки, використовуються сервіси потокової обробки;

- Зберігання даних. Остаточні оброблені дані зберігаються у відповідних сховищах, таких як бази даних для структурованих медичних записів або системи для файлових даних (наприклад, зображення, PDF документи тощо).

- Доступ та аналіз. Дані доступні для подальшого аналізу, генерування звітів та використання різними користувачами системи, включно з медичними працівниками та адміністраторами.

Послідовність процесів збору медичних даних зазначена на рисунку 2.6.

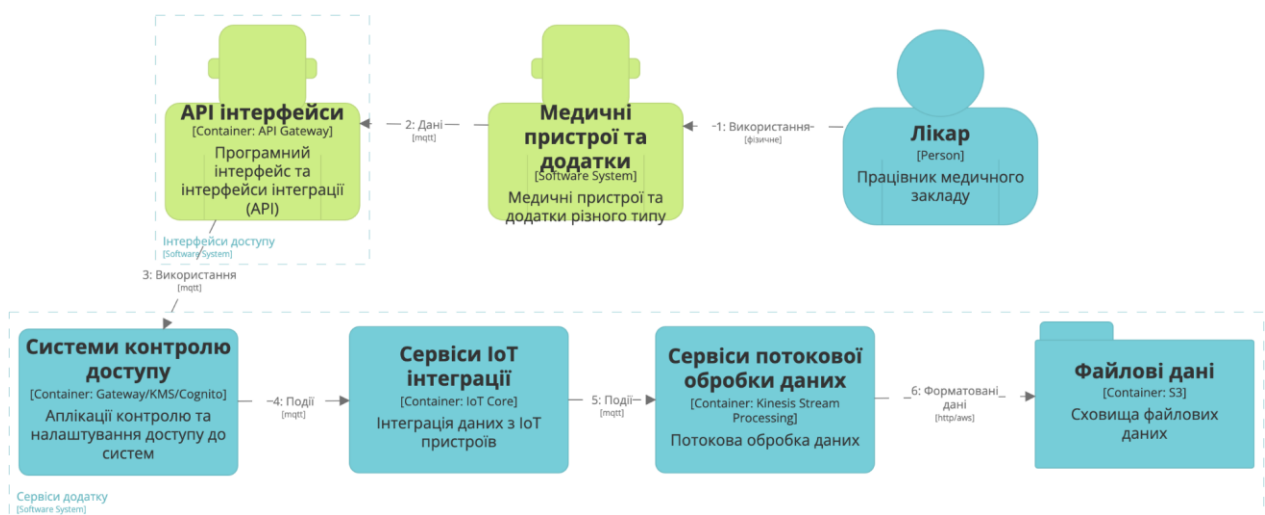


Рисунок 2.6 - Процес збору медичних даних пацієнтів

Попередня обробка даних. На рисунку 2.7 ілюструється спланований процес обробки медичних даних, що починається з первинного збору неструктурованих сирих даних у файловому сховищі, як зазначено на рисунку 2.6. Звідси дані направляються до сервісів внутрішнього обчислення, де вони піддаються процедурам структуризації та групування. Така попередня обробка є критичною для забезпечення ефективності та точності подальшого аналізу даних.



Рисунок 2.7 - Обробка даних пацієнтів

Обробка запитів щодо персональних даних. Крім безпосереднього захисту даних за допомогою протоколів та ролей доступу, а також застосування методів шифрування, вимоги HIPAA (та GDPR) охоплюють також право власника даних на перевірку наявності збережених даних додатками та отримання інформації про суть цих даних. Важливо, що власники даних мають право вимагати видалення своїх персональних даних з системи, якщо це не суперечить іншим зобов'язанням перед компанією, таким як оплата послуг чи інші контрактні угоди [14].

Це ставить перед розробниками завдання не лише забезпечити високий рівень безпеки даних, але й розробити механізми для ефективного управління цими даними, зокрема їх оновленням, видаленням чи перевіркою. Процес формування звітів про наявність персональних даних та перевірка можливості їх видалення може бути часозатратним, тому в поточному випадку застосовується асинхронний підхід до взаємодії. Це дозволяє ефективно обробляти запити

користувачів, мінімізуючи вплив на загальну продуктивність системи.

Окрім того, цей підхід передбачає розробку додаткових інтерфейсів та API для взаємодії з користувачем, що дозволяє власникам даних легко здійснювати перевірку та управління своїми даними. Важливим є також забезпечення прозорості та зрозумілості процесів для користувачів, що є ключовим аспектом у забезпеченні довіри та відповідності нормативним вимогам.

Діаграма, що наведена на рисунку 2.8, демонструє послідовність обробки запитів користувачів щодо їхніх персональних даних у відповідності до вимог захисту даних, зокрема HIPAA та GDPR. Запити можуть включати вимогу надання звіту про наявність персональних даних у системі, їх видалення або перевірки можливості видалення. Запити ініційовані користувачами спершу надходять до системи контролю доступу, де вони реєструються та ставляться у чергу на обробку.

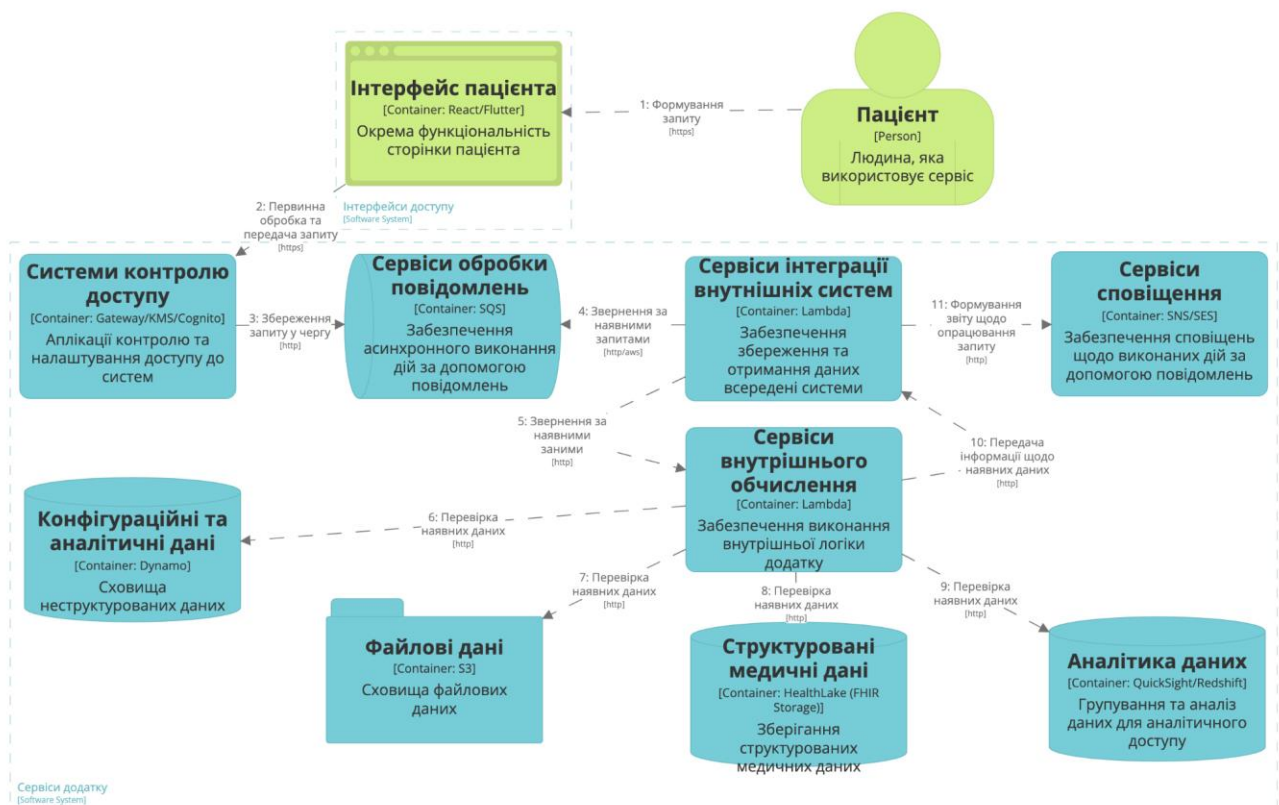


Рисунок 2.8 - Асинхронна обробка запитів щодо персональних даних

Після реєстрації запит проходить через сервіси обробки повідомлень, які

забезпечують зв'язок з внутрішніми системами для формування необхідної інформації. Усі сховища даних систематично опитуються для визначення наявності та локації персональних даних користувача. На основі отриманих даних формується звіт, який узагальнює інформацію про зберігання та обробку персональних даних в системі. Завершальним етапом є використання сервісу сповіщення для надсилання звіту користувачу.

2.6 Архітектура розгортання та інфраструктури

Діаграма розгортання, зазначена на рисунку 2.9, фокусується на безсерверних та хмарних рішеннях, відображає архітектуру розподілу системи, де компоненти та сервіси розміщуються в хмарному середовищі. Основною перевагою такого підходу є висока гнучкість та масштабованість ресурсів, що дає змогу оптимізувати витрати та підтримувати вимоги до продуктивності системи в реальному часі.

Ключові аспекти діаграми та такого варіанту розгортання:

- Безсерверні обчислення. Використання сервісів, які автоматично масштабуються відповідно до потреб додатку;
- Шлюз доступу. Централізована точка входу для управління запитами до різних сервісів;
- Інтеграція з хмарними сховищами. Використання хмарних сховищ для збереження та управління даними;
- Системи контролю доступу. Інтеграція з системами управління ідентифікацією та доступом;
- Сервіси обробки та аналітики даних. Інтеграція з потужними аналітичними та обробними сервісами;
- Сервіси сповіщення та звітності. Використання сервісів для автоматизації сповіщень та генерації звітів.

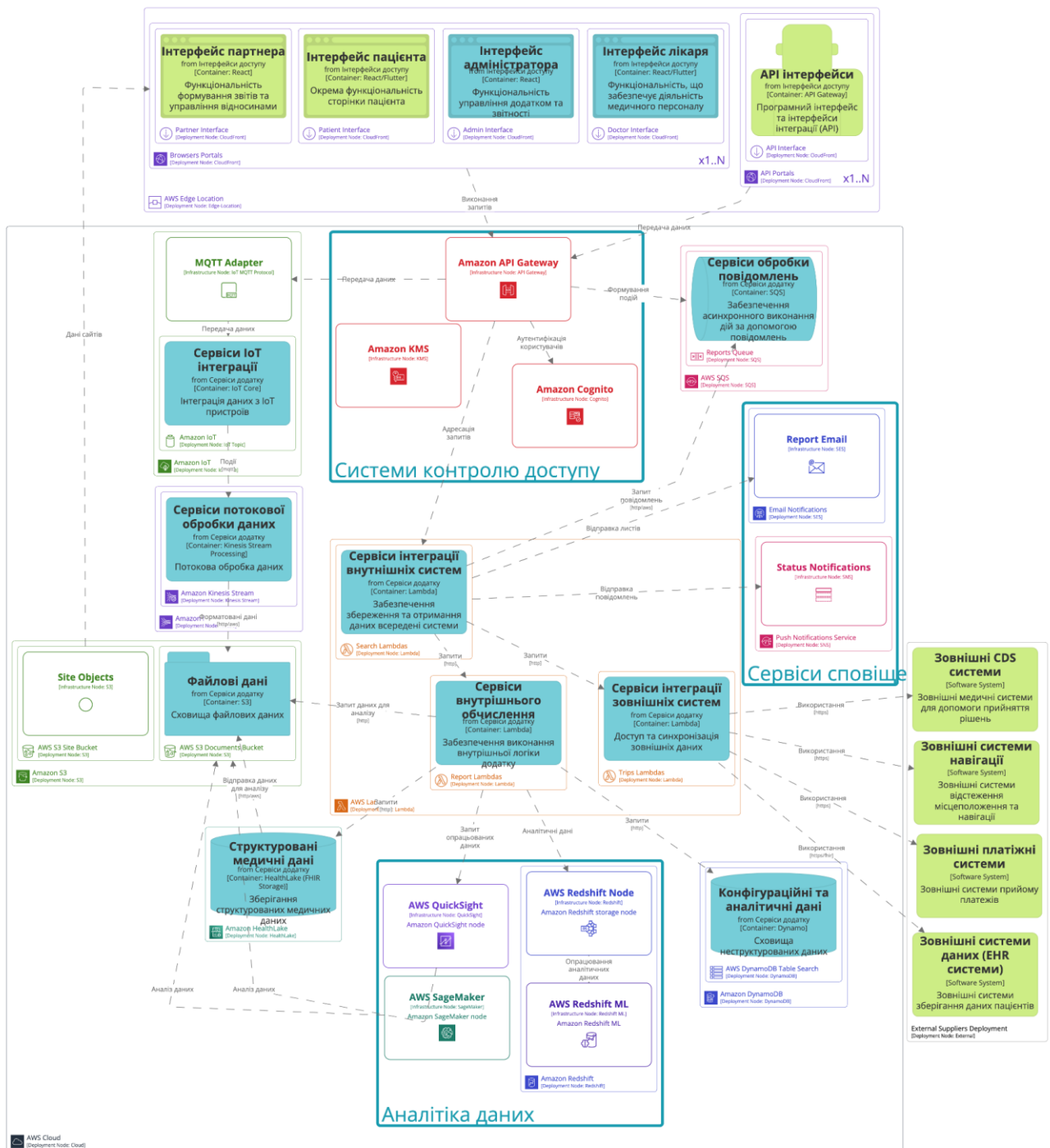


Рисунок 2.9 - Діаграма розгортання компонентів додатку

Ця модель розгортання підходить для систем, які потребують швидкої адаптації до змінюваного навантаження, високого рівня надійності та безпеки. Вона забезпечує ефективне управління ресурсами, оптимізацію витрат та можливість швидкого впровадження змін та оновлень у додатку.

Ключові елементи та сервіси, які використовуються на діаграмі описані в таблиці 2.9.

Таблиця 2.9 - Елементи діаграми розгортання додатку

Технологія	Компоненти	Опис
S3	Файлові дані	Amazon Simple Storage Service (Amazon S3) - це сервіс об'єктного зберігання, який пропонує провідну в галузі масштабованість, доступність даних, безпеку та продуктивність. Використовується Amazon S3 для розміщення статичного веб-сайту.
DynamoDB	Конфігураційні та аналітичні дані	Amazon DynamoDB - це повністю керована, безсерверна, ключово-значуща NoSQL база даних, розроблена для виконання додатків високої продуктивності будь-якого масштабу.
Redshift	Аналітика даних	Amazon Redshift використовує SQL для аналізу структурованих та напівструктурованих даних усередині даних складів, операційних баз даних та даних озер, використовуючи обладнання, розроблене AWS, та машинне навчання для надання найкращої ціни-продуктивності на будь-якому масштабі.
Lambda	Сервіси інтеграції внутрішніх/зовнішніх систем, Сервіси внутрішнього обчислення	AWS Lambda - це безсерверна, обробляна подіями обчислювальна служба, яка дозволяє вам запускати код для практично будь-якого типу додатку або бекенд-сервісу без необхідності виділення або керування серверами.
CloudFront	Інтерфейс лікаря/партнера/пацієнта/IT адміністратора, API Інтерфейси	Amazon CloudFront - це сервіс мережі доставки контенту (CDN), створений для високої продуктивності, безпеки та зручності розробників.
SQS	Сервіси обробки повідомлень	Amazon Simple Queue Service (SQS) дозволяє відправляти, зберігати та отримувати повідомлення між програмними компонентами будь-якого обсягу, не втрачаючи повідомлень або потребуючи доступності інших служб.
SES	Сервіси сповіщення	Amazon Simple Email Service (SES) дозволяє вам впевнено досягати клієнтів без необхідності використання локальної системи Simple Mail Transfer Protocol (SMTP).
SNS	Сервіси сповіщення	Amazon Simple Notification Service (SNS) надсилає сповіщення двома способами: A2A та A2P. A2A забезпечує високу пропускну здатність, відправлення на основі подій, багато-до-багатьох повідомлень між розподіленими системами.
API Gateway	Системи контролю доступу	Amazon API Gateway - це повністю керований сервіс, який полегшує розробникам створення, публікацію, підтримку, моніторинг та захист API будь-якого масштабу.

Продовження таблиці 2.9

Технологія	Компоненти	Опис
Cognito	Системи контролю доступу	Amazon Cognito надає сховище ідентифікації, яке масштабується до мільйонів користувачів, підтримує соціальну та корпоративну ідентифікацію федерації та пропонує передові функції безпеки для захисту споживачів та бізнесу.
KMS	Системи контролю доступу	AWS Key Management Service (KMS) - це керований сервіс, який забезпечує легке створення та керування криптографічними ключами, які використовуються для шифрування даних. KMS інтегрований з іншими сервісами AWS та додатками замовника, дозволяючи легко застосовувати шифрування для захисту даних.
IoT Core	Сервіси IoT інтеграції	AWS IoT Core - це керований хмарний сервіс, який дозволяє легко та безпечно підключати пристрої до хмарних додатків та інших пристроїв, обробляти та зберігати дані з інтернету речей (IoT) і виконувати дії на основі цих даних.
HealthLake	Структуровані медичні дані	AWS HealthLake - це масштабована, безсерверна платформа для зберігання, перетворення та аналізу здоров'я даних в облаках, яка дозволяє організаціям охорони здоров'я легко зберігати, змінювати, проаналізувати та вивчати великі обсяги здоров'я даних.
QuickSight	Аналітика даних	Amazon QuickSight - це повністю керований сервіс бізнес-аналітики, який забезпечує швидку та легку створення та публікацію інтерактивних дашбордів.
SageMaker	Аналітика даних	AWS SageMaker - це повністю керована служба, яка дозволяє розробникам та вченим швидко створювати, тренувати та розгортати моделі машинного навчання на будь-якому масштабі.
Redshift	Аналітика даних	Amazon Redshift використовує SQL для аналізу структурованих та напівструктурованих даних усередині даних складів, операційних баз даних та даних озер, використовуючи обладнання, розроблене AWS, та машинне навчання для надання найкращої ціни-продуктивності на будь-якому масштабі.

Діаграма розгортання інфраструктури, представлена на рисунку 2.10, ілюструє процеси автоматизації, моніторингу та управління інфраструктурою в хмарному середовищі AWS.

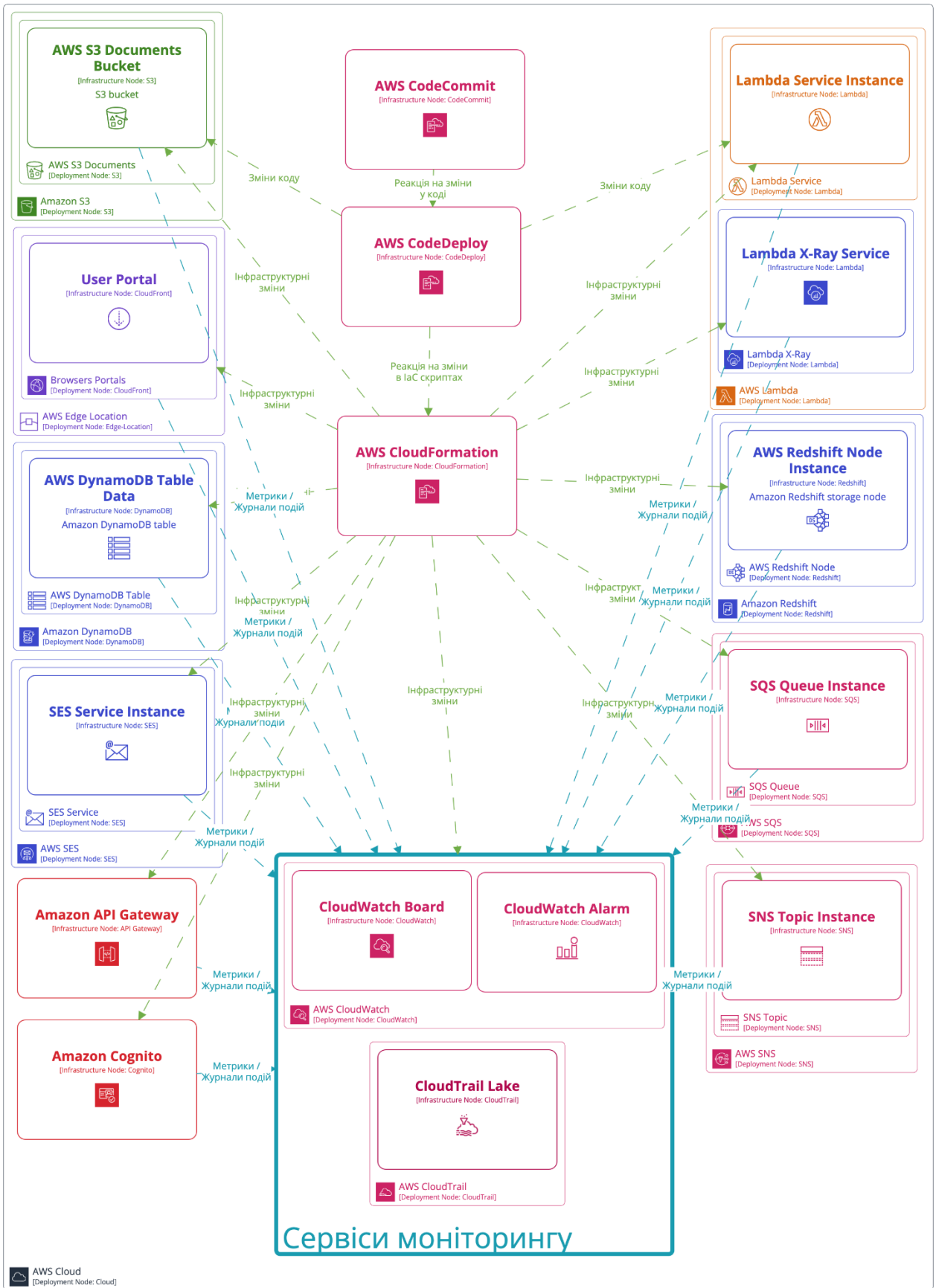


Рисунок 2.10 - Діаграма розгортання інфраструктури додатку

Така діаграма включає наступні ключові елементи:

- Управління кодом та інфраструктурою. Вихідний код додатку та скрипти інфраструктури як коду (Infrastructure as Code, IaC) зберігаються у сервісі AWS CodeCommit, який є безпечним репозиторієм для контролю версій.

- Автоматизація розгортання. AWS CodeDeploy автоматизує процес розгортання додатків на різні середовища, забезпечуючи стандартизацію, ефективність та надійність релізів.

- Оркестрація інфраструктури. AWS CloudFormation дозволяє розробникам використовувати IaC для швидкого та надійного моделювання та розгортання інфраструктури для додатків.

- Моніторинг та оповіщення. Сервіси, такі як Amazon CloudWatch та AWS CloudTrail, використовуються для моніторингу діяльності системи, відстеження подій та запису журналів для забезпечення безпеки та відповідності вимогам.

Цей підхід розгортання акцентує на безперервній інтеграції та доставці (CI/CD), автоматизації інфраструктури та високому рівні моніторингу, що є ключовими для сучасних хмарних додатків. Він забезпечує швидке впровадження змін, високу доступність сервісів та гнучкість управління масштабами та витратами.

На рисунку 2.10 не відображений повний перелік сервісів, які використовуються у загальній діаграмі розгортання, оскільки він має на мені відобразити загальний процес та взаємодію інфраструктурних компонентів.

Ключові інфраструктурні елементи та сервіси, які використовуються на діаграмі описані в таблиці 2.10.

Таблиця 2.10 - Ключові інфраструктурні елементи та сервіси додатку

Технологія	Мета	Опис
CodeCommit	Зберігання коду та скриптів IaC	AWS CodeCommit — це безпечний, високомасштабований, повністю керований сервіс контролю версій, який розміщує приватні репозиторії Git.
CodeDeploy	Організація автоматичного розгортання змін інфраструктури та	AWS CodeDeploy — це повністю керована служба розгортання, яка автоматизує програмні розгортання на різноманітні обчислювальні сервіси, такі як Amazon Elastic Compute Cloud

Продовження таблиці 2.10.

Технологія	Мета	Опис
	коду	(EC2), Amazon Elastic Container Service (ECS), AWS Lambda та ваші локальні сервери.
CloudFormation	Розгортання об'єктів інфраструктури та конфігурації за підходом IaC	AWS CloudFormation дозволяє моделювати, розгортати та керувати ресурсами AWS та сторонніми ресурсами, обробляючи інфраструктуру як код.
CloudWatch	Моніторинг метрик та журналів	Amazon CloudWatch збирає та візуалізує дані журналів, метрик та подій у реальному часі в автоматизованих інформаційних панелях для спрощення обслуговування інфраструктури та додатків.
CloudTrail	Відстеження діяльності користувачів та API	AWS CloudTrail — це служба, що забезпечує відстеження дій користувачів та звернень до API, збираючи інформацію про події, що відбуваються в AWS-обліковому записі для забезпечення безпеки та відповідності аудиту.

2.7 Забезпечення атрибутів якості

Забезпечення атрибутів якості системи є вирішальним фактором для успішного проектування та експлуатації програмного забезпечення, особливо в контексті сучасних вимог до швидкості, безпеки та гнучкості. Методологія ATAM надає систематичний підхід для оцінювання архітектури з точки зору атрибутів якості, які є критично важливими для бізнес-цілей та стейкхолдерів. ATAM допомагає ідентифікувати потенційні ризики, які можуть вплинути на виконання найважливіших атрибутів якості, та розробити стратегії їх мінімізації. Такий підхід дозволяє не тільки визначити, як атрибути якості підтримуються в поточній архітектурі, але й прогнозувати, як зміни в архітектурі можуть вплинути на ці атрибути у майбутньому. Врахування цих аспектів є ключовим для розробки стійкої та адаптивної системи, здатної задовольнити потреби користувачів та витримати виклики сучасного технологічного світу.

Доступність. Виходячи із запропонованої структури розгортання, можна покладатися на контракти на надання послуг AWS, на основі угод про рівень обслуговування, наданих AWS [17]. Без використання додаткових методів

підвищення доступності, можна досягти рівня — 99,5% доступності. Усі служби, будуть використовуватись, відповідають цій вимозі.

Крім того, стратегія хмарного розгортання дозволяє підвищити цінність доступності за рахунок дублювання послуг у різних регіонах або зонах доступності [18]. Такий підхід значно збільшує витрати, тому слід уникнути такого підходу на етапі розробки та впровадження.

Інтероперабельність. Основна інтеграція відбувається із зовнішніми постачальниками послуг. Щоб зробити можливим додавати нову інтеграцію простим способом, запропоновано розділити точки інтеграції на окремі виконувані процедури. Запит на пошук медичних даних пацієнта надає параметри запиту разом із відповідними бізнес-правилами та ключами, а оркестратор пошуку може вибрати список постачальників і параметри пошуку відповідно до вимог. Відповідь може бути представлена як потік, який може використовувати потік лямбда-відповідей, щоб не чекати відповідей від кожного постачальника та не блокувати потік користувача під час виконання пошукового запиту [19]. Послідовність дій інтеграції систем зазначена на рисунку 2.11.

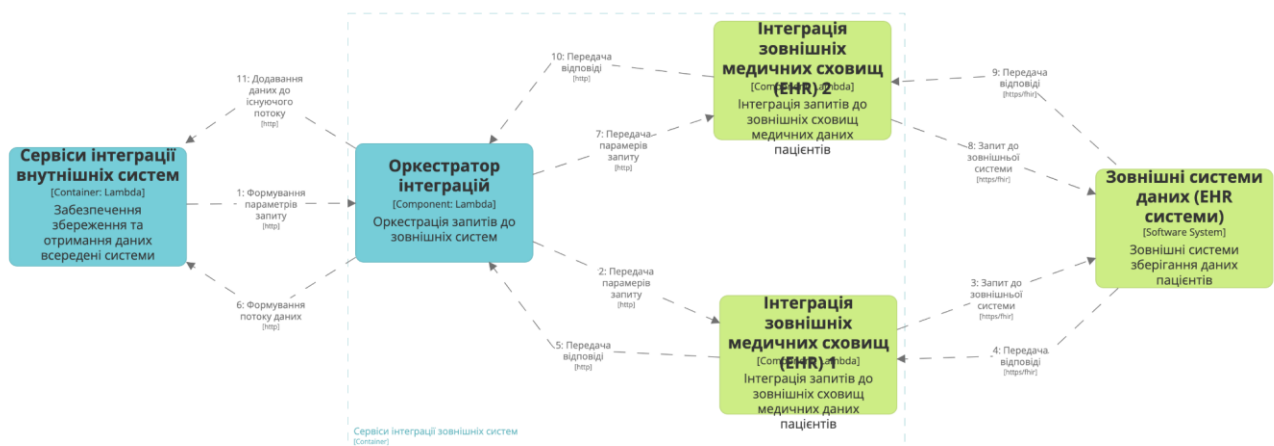


Рисунок 2.11 - Інтеграція зовнішніх систем додатку

Аудитопридатність. Керування даними має відповідати HIPAA. Це означає, що потрібно збирати журнали аудиту для всіх запитів на доступ до даних. За допомогою AWS CloudWatch та AWS CloudTrail можна збирати всі необхідні дані з усіх розгорнутих і налаштованих служб. Подальше архівування даних буде вимагати додаткових коштів та налаштувань, тож слід окремо розглядати

вимоги та впроваджувати додаткове архівування даних лише після впровадження системи в робоче середовище.

Масштабованість. Виходячи із запропонованої структури розгортання та фокусу даної роботи, запропоноване використання хмарних безсерверних технологій. Такий підхід забезпечує практично необмежену масштабованість. У той же час слід мати на увазі, що очікується обмежене робоче навантаження. У разі значного збільшення робочого навантаження запропоноване рішення може бути нерентабельним і потребуватиме архітектурних змін.

Швидкодія. Розділення користувальницьких порталів робить їх більш легкими, а розповсюдження за допомогою CDN забезпечує кращий час завантаження. У той же час, на стороні серверної частини, запропоновано зробити довготривалі операції (наприклад, створення звітів) асинхронними за допомогою сервісів обробки повідомлень.

Деякі запити стосуються пошуку даних у зовнішніх системах, що унеможлиблює прямий контроль над швидкодією. У такій ситуації запропоновано робити запити до різних систем постачальників поспіль або паралельно та об'єднувати відповіді в потоці. Зі сторони проекту, не можна вплинути на виконання пошукових запитів до кожного окремого постачальника, тому такий підхід дозволяє не чекати повної відповіді, а повертати результати частково.

Безпека. Загальний контроль доступу забезпечується за допомогою сервісу AWS API Gateway, який контролює неавторизовані запити до системи. За формування прав доступу та акаунтів користувачів відповідає сервіс AWS Cognito. Amazon API Gateway підтримує різні типи шифрування під час передавання, включаючи TLS 1.1. та 1.2. Крім того, за потреби можна налаштувати додаткові правила безпеки.

Шифрування медичних даних пацієнтів у хмарних сховищах є критично важливим аспектом забезпечення конфіденційності та захисту від несанкціонованого доступу чи витоків інформації [21]. Враховуючи чутливий характер особистої медичної інформації та строгі вимоги до дотримання норм,

таких як HIPAA у США та GDPR в Європі, шифрування на рівні сховища даних стає необхідним засобом для забезпечення цілісності та конфіденційності даних пацієнтів. Забезпечення внутрішньої безпеки та шифрування даних, що зберігаються, може бути досягнуто за допомогою таких сервісів AWS, як AWS IAM та AWS KMS. Використовуючи IAM, слід визначити ролі доступу для внутрішніх сервісів. З KMS можна керувати шифруванням даних, що зберігаються. На початковому етапі проекту можна використовувати шифрування на стороні сервера для всіх даних. Для даних, які передаються, використовуються протоколи шифрування HTTPS та TLS. Взаємодія з сервісами забезпечення внутрішньої безпеки проілюстровано на рисунку 2.12.

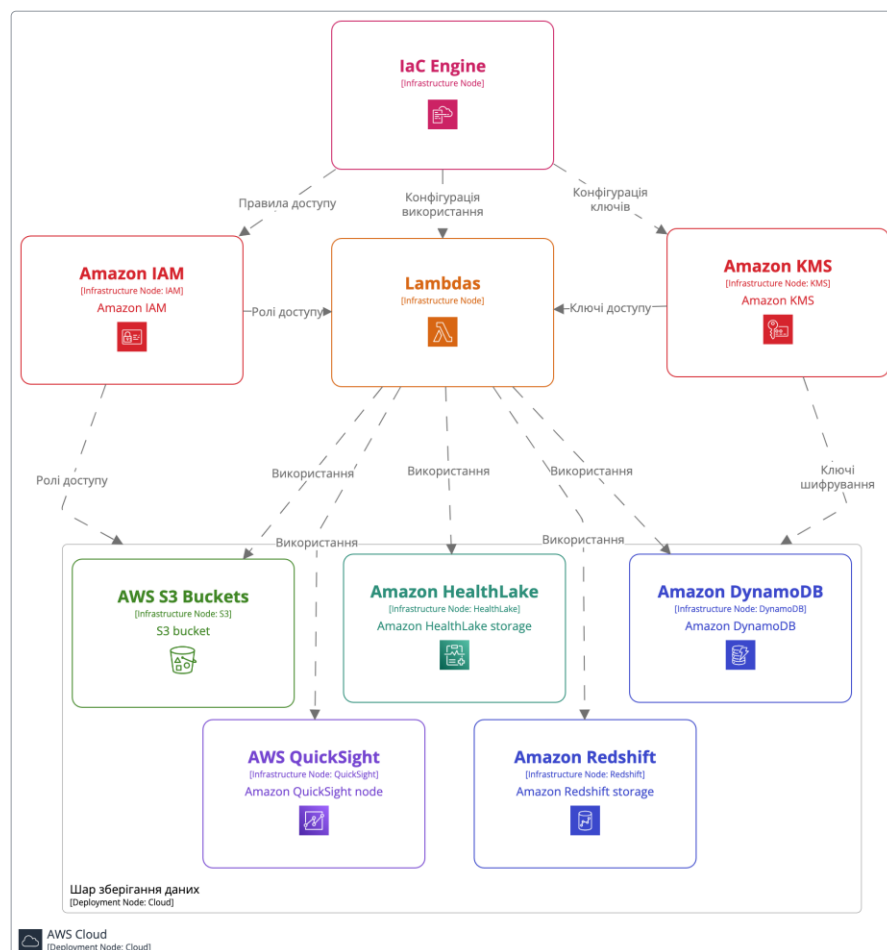


Рисунок 2.12 - Загальна структура забезпечення безпеки даних

Таблиця 2.11. надає додатковий опис елементів та сервісів, що забезпечують безпеку зберігання та передачі даних.

Таблиця 2.11 - Технології забезпечення безпеки сховищ даних

Технологія	Опис
Amazon IAM	За допомогою AWS Identity and Access Management (IAM) можна визначати, хто або що може мати доступ до сервісів та ресурсів AWS, централізовано управляти дрібнозернистими дозволами та аналізувати доступ для уточнення дозволів по всій хмарі AWS.
Amazon KMS	AWS Key Management Service (AWS KMS) дозволяє створювати, управляти та контролювати криптографічні ключі у додатках та сервісах AWS. За допомогою AWS KMS можна активувати шифрування на стороні сервера, використовуючи ключі KMS, якими можна керувати та управляти.

Діаграма, наведена на рисунку 2.13, демонструє інтегрований підхід до безпеки в хмарному розгортанні, використовуючи набір спеціалізованих сервісів AWS для забезпечення захисту додатку на рівні периметра, даних, що передаються, та внутрішніх процесів. Вона відображає глибоку інтеграцію між компонентами AWS для створення комплексного та еластичного середовища безпеки. Слід зазначити, що додаткові системи захисту потребують додаткового налаштування та витрат на обслуговування.

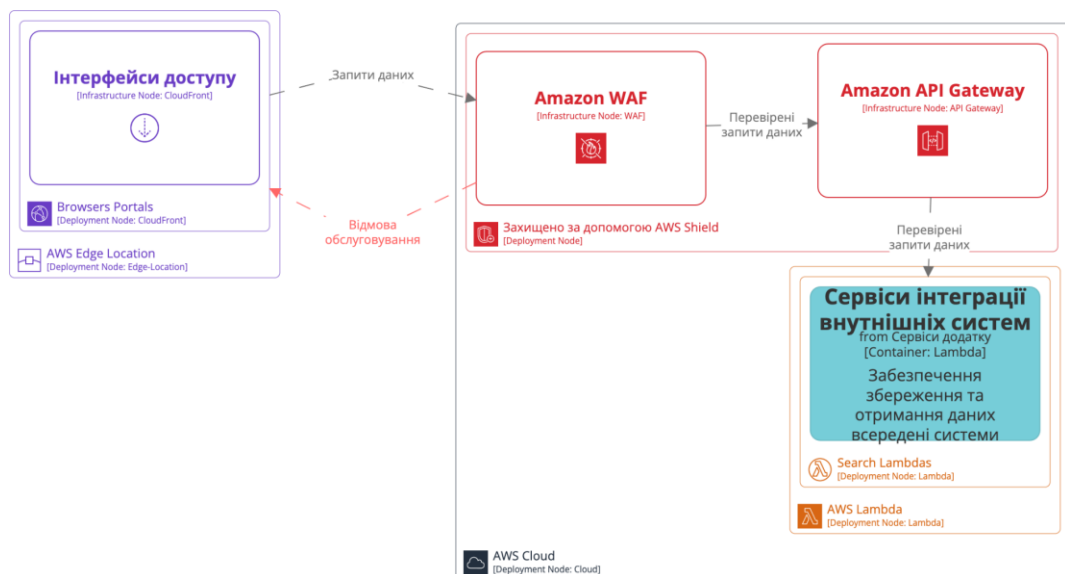


Рисунок 2.13 - Додаткові сервіси захисту доступу

У таблиці 2.12. описані сервіси захисту зовнішнього доступу до додатку.

Таблиця 2.12 - Додаткові сервіси захисту доступу

Технологія	Опис
AWS WAF	AWS Web Application Firewall (WAF) дозволяє контролювати трафік до веб-додатку, застосовуючи заздалегідь визначені правила фільтрації, які можуть запобігати спробам SQL ін'єкції, скриптів сайтів на стороні клієнта (XSS) та іншим загрозам безпеці.
AWS Shield	AWS Shield - це керована служба захисту від DDoS-атак, яка забезпечує захист додатків від атак розподіленого відмовлення в обслуговуванні на шарах 3, 4 та 7. AWS Shield надає автоматичні захисні заходи, які мінімізують збої та знижують шкоду від таких атак.

2.8 Аналіз ризиків та архітектурних обмежень

Основна орієнтація на безсерверні технології дозволяє уникнути багатьох несумісностей між версіями сховищ, шлюзів тощо. Крім того, завдяки модульності сервісів безсерверного обчислення архітектура майже не обмежена у виборі мови програмування, тобто різні частини додатку можуть бути написані на різних мовах, що жодним чином не вплине на роботу додатку в цілому.

Архітектурні обмеження проекту визначають рамки, у яких розробники та архітектори повинні працювати для досягнення бізнес-цілей, одночасно забезпечуючи ефективність, безпеку та масштабованість системи. Слід зазначити кілька можливих архітектурних обмежень для проекту:

- Частина пошуку даних пацієнтів та клінічних можна змінити з використанням асинхронних пошукових запитів, якщо буде інтегровано ще кількох провайдерів. Поточна архітектурна пропозиція передбачає синхронну відповідь, навіть якщо виконуються пошукові запити до кінцевих постачальників поспіль або паралельно. Якщо в системі багато постачальників, це може зайняти деякий час, що є нерозумним для Lambda Orchestrator.

- Оптимізація витрат на ресурси є важливою для економічності хмарних проектів, особливо при використанні безсерверних сервісів, як AWS Lambda. Такі сервіси з розрахунком оплати за фактичне використання можуть стати

дорогими при масштабуванні, що вимагає обачного планування. Перехід до контейнеризованих рішень, наприклад AWS ECS Fargate, може запропонувати більш вигідні умови витрат, забезпечуючи простіше управління ресурсами та стабільність для систем зі змінною активністю.

- При швидкому зростанні користувачів системи ідентифікації, витрати на AWS Cognito можуть стати значними, оскільки сервіс розрахований на помірне навантаження. У таких умовах більш економічно вигідними можуть бути хостингові рішення ідентифікації, які пропонують гнучке ціноутворення та легше масштабування для великої кількості користувачів. Використання відкритих стандартів автентифікації спрощує потенційну міграцію, забезпечуючи зниження витрат і покращення швидкодії.

Перегляд загальних ризиків проекту є фундаментальною частиною управління проектами, оскільки він допомагає ідентифікувати та аналізувати потенційні перешкоди та виклики, які можуть вплинути на успішне виконання проекту. Врахування ризиків, від фінансових і технологічних до правових та ринкових, дозволяє команді розробити ефективні стратегії їх мінімізації або усунення. Це забезпечує здатність проекту адаптуватися до змін, знижує імовірність непередбачених витрат і збоїв у роботі та підвищує загальну ймовірність успіху проекту.

Виходячи з потреб проекту та запропонованого архітектурного рішення, варто зосередитися на кількох загальних ризиках:

- Надійність обслуговування. оскільки система покладається на хмарні безсерверні технології, може виникнути ризик збоїв у роботі або збоїв у роботі. У даній роботі запропоновано обрати AWS як хмарного постачальника як це зазначено у додатку А.1. На основі загальнодоступного AWS SLA [17] маємо досить високий рівень доступності послуг і дуже високий рівень довговічності даних. Навіть у разі регіональних збоїв використання ІаС дозволяє відновити повну інфраструктуру та перенести дані після того, як регіон стане доступним. Крім того, можна тиражувати або створювати резервні копії даних у різних регіонах, щоб відновити всю інфраструктуру за короткий проміжок часу.

- Ризики безпеки даних. Неавторизований доступ, атаки проникнення. У розглянутій пропозиції пропонується використовувати контроль доступу на основі ролей. Сервіс Cognito забезпечує керування безпекою з можливістю налаштування. Але справжня гарантія безпеки повинна бути досягнута за допомогою автоматизованих тестів, які можуть перевіряти багато можливих сценаріїв з різними типами користувачів. Для пом'якшення атак проникнення розглянуто також додаткова інтеграція служб безпеки, таких як AWS Shield і AWS WAF.

- Ризики постачальника. Залежність від хмарного постачальника. Як описано вище, AWS SLA забезпечує високий рівень стабільності. Можливе впровадження багатохмарного рішення. Найроповсюджені хмарні провайдери надають сумісні послуги та сервіси [23]. Але такий підхід призводить до збільшення витрат і складності, тому слід такі варіанти розглядати у разі необхідності.

2.9 Розробка та підтримка проекту

Фокусування проекту на використанні хмарних та безсерверних рішень має потенціал істотно знизити часові та ресурсні витрати на розробку та підтримку інфраструктури. Хмарні сервіси, такі як AWS Lambda, дозволяють розробникам концентруватися на бізнес-логіці додатків, уникаючи складнощів з управлінням серверною інфраструктурою. Водночас, необхідно уважно стежити за витратами на хмарні сервіси, адже масштабування та інтенсивне використання можуть призвести до несподіваного зростання витрат.

Розробка. Розглянута архітектурна пропозиція не накладає обмежень на вибір мови програмування. Проте рекомендується використовувати скриптові мови, такі як Python, Go та JavaScript. Такий підхід може скоротити час запуску Lambda та використання пам'яті. Безумовно, слід використовувати Lambda Layers для уникнення дублювання спільних частин коду. Використання Lambda Layers дозволяє стандартизувати бібліотеки та залежності, спрощуючи

Таблиця 2.13 - Додаткові сервіси організації кількох оточень розробки

Сервіс	Опис
Cost Management	Група послуг, які дозволяють організовувати та звітувати про витрати та використання на основі методів, визначених користувачем, керувати виставленням рахунків і контролювати витрати
General IAM	За допомогою AWS Identity and Access Management (IAM) можна вказати, хто або що може отримувати доступ до служб і ресурсів в AWS, централізовано керувати детальними дозволами та аналізувати доступ для уточнення дозволів у AWS. На рівні організації можна встановити загальні правила для всієї компанії.

Технічне обслуговування. Завдяки використанню безсерверного підходу до розгортання, загальне технічне обслуговування інфраструктури є відповідальністю обраного хмарного провайдера.

Тестування. У поточному проекті пропонується зосередитися на end-to-end (e2e) тестуванні. З точки зору розробників:

- Юніт-тести є обов'язковими. Для юніт-тестів слід обрати відповідний інструмент, що базується на обраній мові програмування.

- Інтеграційні тести слід писати для тих частин, де це є доцільним. Технології AWS SAM [26] дозволяють створити інтеграційні тести для безсерверних обчислень.

Загальний пайплайн може включати різні обсяги e2e тестів (наприклад, димові тести, повні тести тощо). Також корисним може бути проведення тестів на тижневій або двотижневій основі.

Пайплайн розгортання повинен включати статичний аналіз коду, юніт-тести, розгортання, e2e тести. Пайплайн, який запускається для вищого середовища, можливий лише в разі успішного проходження на нижчому середовищі.

Використання зовнішніх сервісів тестування зображено на рисунку 2.15.

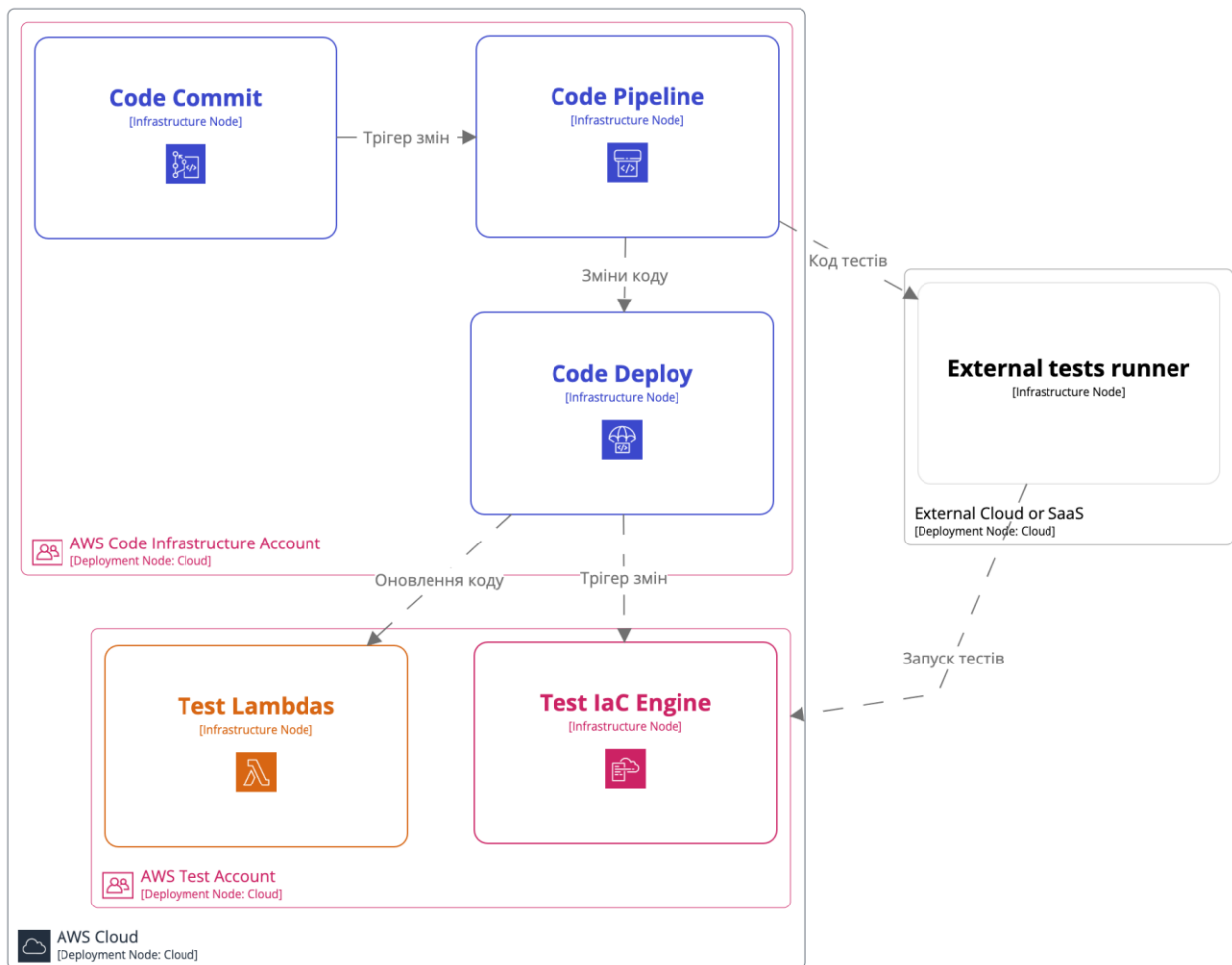


Рисунок 2.15 - Процес розгортання та тестування додатку

Таблиця 2.14 описує додаткові сервіси, які допомагають контролювати процес розробки та тестування.

Таблиця 2.14 - Додаткові сервіси розгортання додатку

Компонент	Технології	Опис
Code Pipeline	AWS Code Pipeline	AWS CodePipeline — це повністю керована служба безперервної доставки, яка допомагає вам автоматизувати конвеєри випусків для швидкого й надійного оновлення додатків та інфраструктури.
External tests runner	Можливі варіанти: - SauceLabs - BlazeMeter - інші	Система запуску тестів, яка підтримує різні види тестів, у тому числі ручний запис сценарію. Така система забезпечує налаштування інформаційної панелі та може працювати за розкладом.

Фокус на e2e тестах у цьому проекті зумовлений використанням сервісів, орієнтованих на безсерверні рішення, що передбачає виконання невеликих частин функціоналу в кодї для кожного окремого сервісу. Тому перевірка повної

функціональності системи є більш важливою, ніж тестування окремих компонентів. Такий підхід дозволяє забезпечити, що всі компоненти системи працюють разом ефективно та відповідають вимогам користувачів і бізнесу, особливо в контексті медичних додатків, де надійність та точність є критично важливими.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ РОЗГОРТАННЯ АРХІТЕКТУРИ

3.1. Обґрунтування частини проекту для практичної реалізації

У даному розділі особливу увагу варто приділити ролі та значенню мережі доставки контенту (CDN). Застосування CDN є критично важливим для підприємств медицини надзвичайних ситуацій, оскільки швидкість та доступність медичної інформації можуть мати безпосередній вплив на своєчасне надання допомоги пацієнтам. Важливість використання мережі доставки контенту (Content Delivery network - CDN) для підприємств, що спеціалізуються на медицині надзвичайних ситуацій, не можна недооцінювати. Використання CDN може значно покращити користувацький досвід завдяки зниженню затримок при завантаженні веб-сторінок та доступу до API, розподіляючи контент ближче до кінцевих користувачів. Додатковою перевагою є зниження вартості передачі даних та витрат на інфраструктуру, оскільки CDN здатні ефективно розподіляти навантаження, тим самим знижуючи потребу в дорогих центрах обробки даних та мережевій інфраструктурі [27]. Це особливо важливо для медичних установ, які прагнуть оптимізувати витрати, але при цьому не можуть допустити компромісу в питаннях надійності та продуктивності.

В якості технології для формування та опрацювання IaC скриптів обрано AWS CloudFormation як основний інструмент IaC для розгортання інфраструктури проекту, як це зазначено в додатку А.4.

3.2 Загальне розуміння CloudFront CDN

Amazon CloudFront є інтегральною частиною сучасної архітектури веб-додатків, особливо коли мова йде про медичні системи, де вимоги до швидкості обробки даних та їх доступності є надзвичайно високими. Без використання CDN, швидкість та ефективність веб-додатку стають суттєво обмеженими через географічні та інфраструктурні відстані, які можуть істотно впливати на час

завантаження ресурсів та відповіді сервера [28].

Рисунок 3.1 ілюструє запит користувачів до єдиної локації зберігання даних.

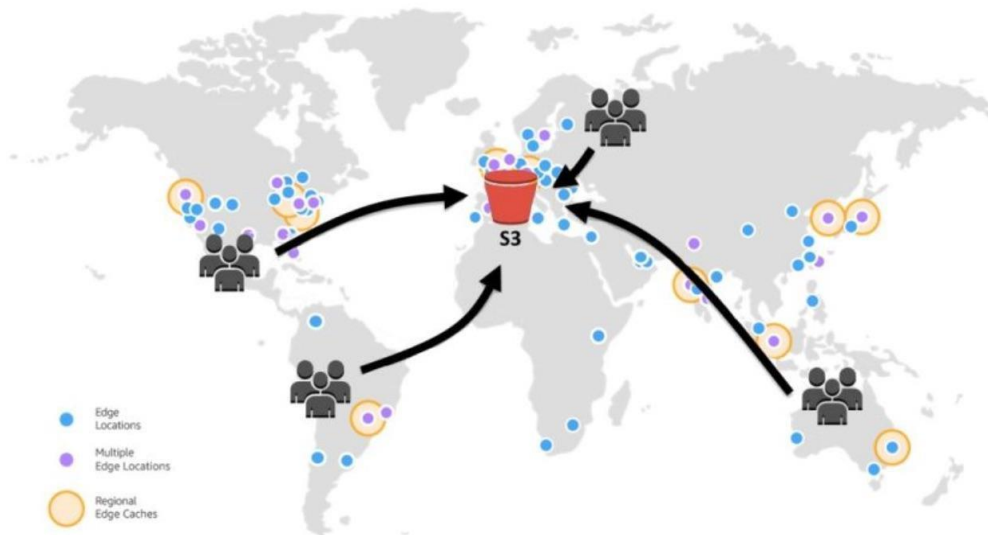


Рисунок 3.1 - Розташування точок доступу AWS

Коли користувач запитує вміст, який обслуговується за допомогою CloudFront, його запит спрямовується до найближчого розташування Edge. Якщо CloudFront має кешовану копію запитуваного файлу, CloudFront доставляє її користувачеві, надаючи швидку (з низькою затримкою) відповідь. Якщо запитуваний файл ще не кешовано, CloudFront отримує його з джерела, наприклад, із S3, де зберігається вміст. Рисунок 3.2 відображає запит користувачів до найближчої локації де дані зберігаються в локальному кеші.

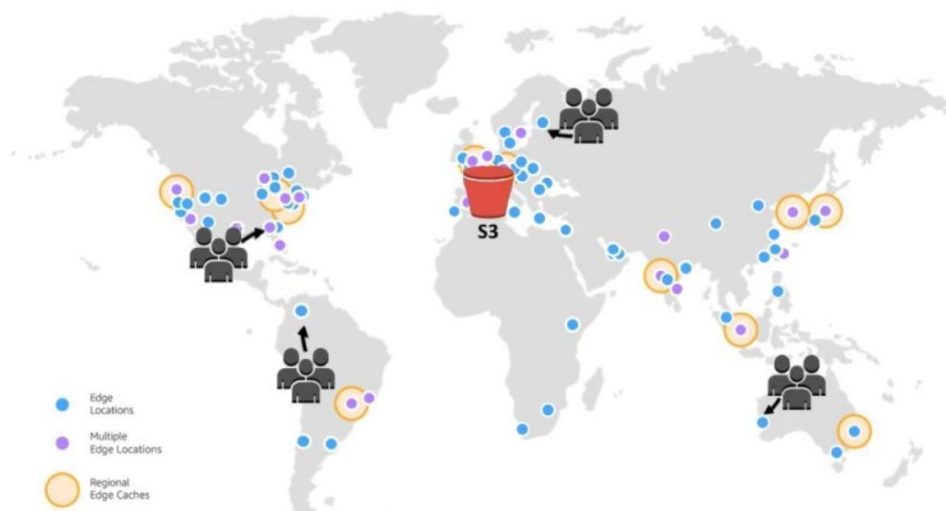


Рисунок 3.2 - Доступ користувачів до Edge Location

Навіть якщо немає потреби кешувати вміст, наприклад, оскільки обслуговується динамічний вміст, CloudFront покращує доставку вмісту, оскільки Edge Locations встановлює та підтримує з'єднання ближче до користувачів. CloudFront також використовує глобальну приватну мережу AWS, окрему магістраль в Інтернеті, яка допомагає обійти глобальні проблеми мережі, щоб забезпечити кращу продуктивність як для статичного, так і для динамічного вмісту. Рисунок 3.3 відображає результати тестів [28] щодо швидкодії кешу локацій.

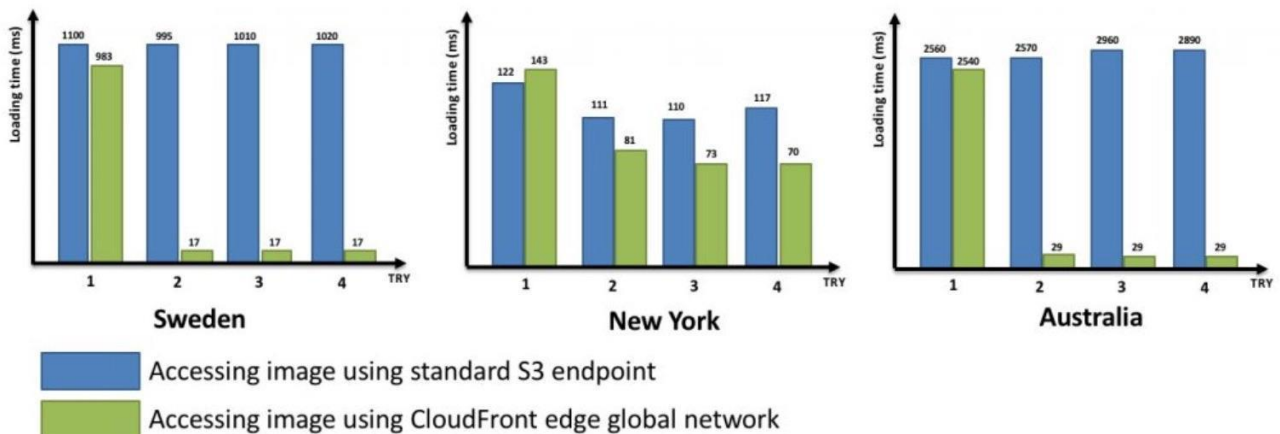


Рисунок 3.3 - Прискорення видачі контенту у різних регіонах (якщо оригінальний контент зберігається у США Північна Вірджинія)

Таким чином, розгортання CloudFront у проекті медичного веб-додатку не тільки покращує користувацький досвід за рахунок зниження часу завантаження та підвищення доступності, але й виступає як стратегічний крок у забезпеченні безпеки та надійності системи, що надзвичайно важливо для підтримки життєво важливих медичних процесів.

Для цілей проекту, розглянутого в розділі 2, вже була запропонована модель використання CDN та CloudFront. Для цілей цього розділу буде розширена діаграма використання CloudFront як зазначено на рисунку 3.4.

Діаграма на рисунку 3.4 демонструє гнучку та ефективну структуру розгортання, яка оптимізує взаємодію між користувачами та додатком, а також

у вигляді коду (Infrastructure as Code - IaC), що робить процес розгортання інфраструктури повторюваним, автоматизованим та з легкістю контрольованим [29].

Для керування дозволами на створення ресурсів використовується AWS Identity and Access Management (IAM). Окреме дослідження AWS IAM не є предметом цієї роботи, тому для цілей розгортання проекту в подальшому буде використаний користувач AWS з повним набором прав та привілеїв (root user).

Загальний процес створення стеку CloudFormation може бути проілюстрований рисунком 3.5. [30].



Рисунок 3.5 - Процес створення стеку CloudFormation

Для виклику створення, зміни або видалення стеку можна використовувати як автоматизовані тригери так і ручний запуск з використанням AWS CLI, AWS Console чи AWS SDK.

Якщо потрібно оновити ресурси стека, можна змінити шаблон стека. Для цього не потрібно створювати новий стек і видаляти старий. Щоб оновити стек, треба створити набір змін, надіславши змінену версію вихідного шаблону стеку, інші значення вхідних параметрів або обидва. CloudFormation порівнює змінений шаблон із вихідним шаблоном і створює набір змін. На рисунку 3.6. підсумовано робочий процес для оновлення стека.

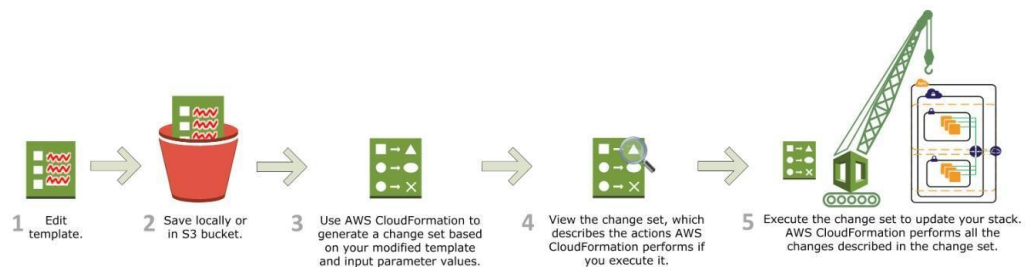


Рисунок 3.6 - Процес внесення змін в стек CloudFormation

Для видалення стеку необхідно вказати стек для видалення, а CloudFormation видалеє стек і всі ресурси в цьому стеку. Стеки можна видалити за допомогою консолі CloudFormation, API або AWS CLI.

AWS документація наводить низку прикладів сценаріїв, які можна використовувати, щоб зрозуміти, як декларувати різні частини шаблону AWS CloudFormation [31].

Для додаткової зручності розробки та аналізу скриптів розгортання інфраструктури створено сервіс Designer [32]. AWS CloudFormation Designer - це графічний інструмент, який дозволяє створювати, переглядати та модифікувати шаблони CloudFormation за допомогою дружнього до користувача інтерфейсу.

На рисунку 3.7 зображена візуалізація стеку сервісів, який потрібний для розгортання інфраструктурних об'єктів, що зображені на рисунку 3.4.

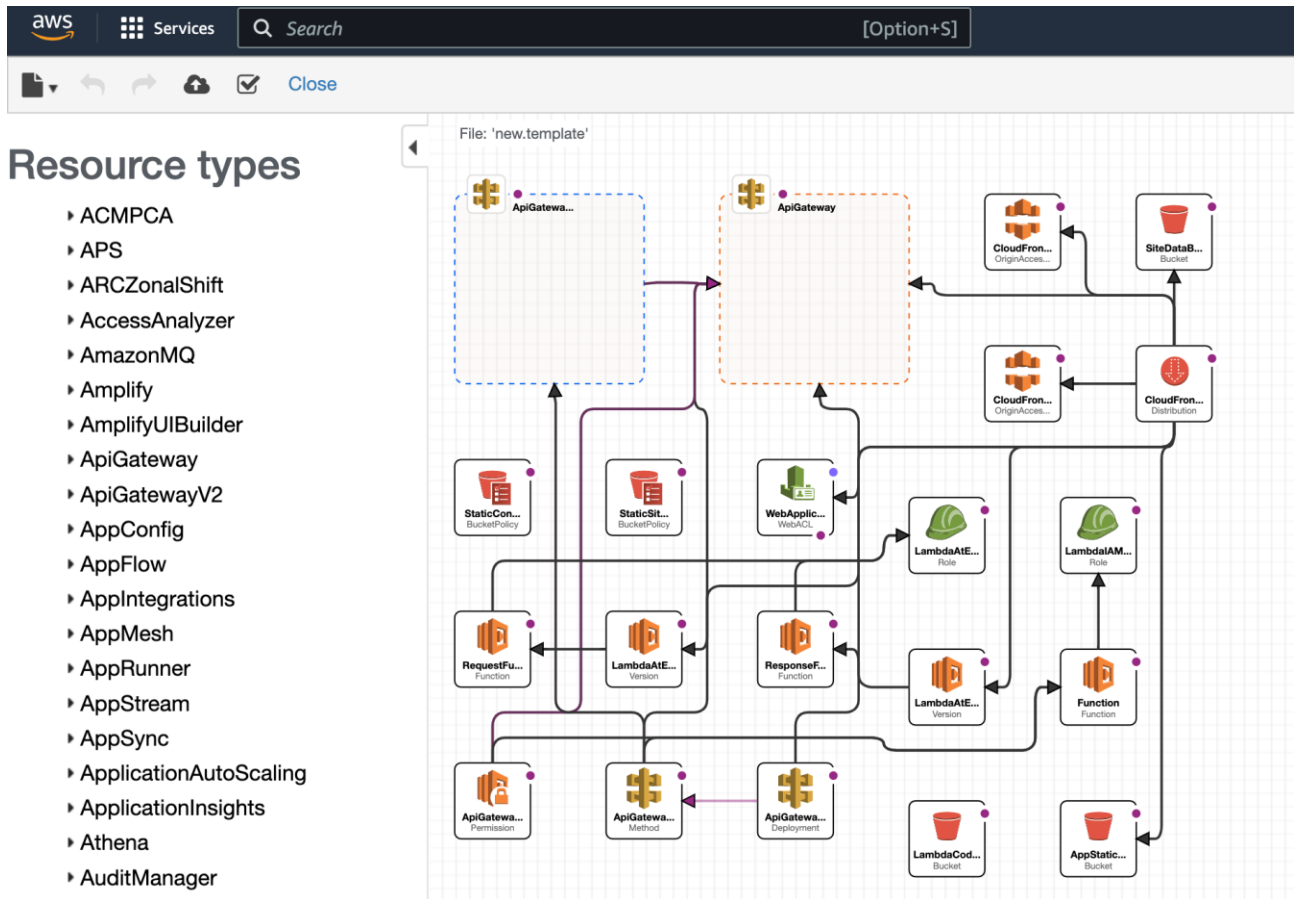


Рисунок 3.7 - Візуалізація CloudFormation стеку за допомогою Designer

3.4 Формування скриптів та параметрів розгортання

Для формування прикладів запуску команд до хмарного провайдера буде використано AWS CLI (Command Line Interface). Це потужний інструмент, що об'єднує можливості всіх індивідуальних інтерфейсів API сервісів AWS в один інтерфейс командного рядка, що робить управління хмарними ресурсами швидким та ефективним.

Оскільки даний розділ фокусується на розгортанні сервісів взаємодії з клієнтами, створюються невеликі сервіси, які будуть служити джерелом взаємодії. Так будуть потрібні наступні компоненти:

- S3 бакет, який буде зберігати дані сайту;
- S3 бакет, який буде зберігати статичний контент, що може наповнюватися (текстові документи, зображення тощо);

- S3 бакет, який буде зберігати код для Lambda сервісів;
- Lambda функція, яка формує невеликий API взаємодії;
- API Gateway, що обробляє вхідні запити.

У процесі підготовки до розгортання хмарної інфраструктури проекту, важливим етапом є формування стеків AWS CloudFormation. Засновуючись на рекомендаціях AWS [33], слід врахувати декілька ключових аспектів, які впливають на архітектуру та майбутнє масштабування системи. У світлі потенційної необхідності внесення змін у певні сегменти системи, підходить розробка індивідуальних шаблонів CloudFormation, що відображають різні логічні блоки проекту.

Водночас, надмірна деталізація шаблонів до рівня окремих сервісів чи їх компонентів може ускладнити процес управління та локалізації потенційних проблем. Таким чином, створений шаблон для розгортання S3 ресурсів.

```

AWSTemplateFormatVersion: '2010-09-09'
Description: Deploying S3 buckets

Resources:
  SiteDataBucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: emer-appl-simple-site-data-bucket

  AppStaticContentBucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: emer-appl-static-content-bucket

  LambdaCodeBucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: emer-appl-lambda-code-bucket

Outputs:
  SiteDataBucketDomainName:
    Description: The domain name of the site data bucket
    Value: !GetAtt SiteDataBucket.DomainName
    Export:
      Name: Emer-SiteDataBucketDomainName
  AppStaticContentBucketDomainName:
    Description: The domain name of the static content bucket
    Value: !GetAtt AppStaticContentBucket.DomainName
    Export:
      Name: Emer-AppStaticContentBucketDomainName
  LambdaCodeBucketDomainName:
    Description: The domain name of the lambda code bucket
    Value: !GetAtt LambdaCodeBucket.DomainName
    Export:
      Name: Emer-LambdaCodeBucketDomainName

```

Для фактичного запуску процесу створення ресурсів використовуємо AWS CLI команду “cloudformation”, яка дозволяє керувати сервісом AWS CloudFormation.

```
% aws cloudformation validate-template --template-body file://buckets.yaml
% aws cloudformation deploy --stack-name emer-s3-buckets --template-file
buckets.yaml
```

Команда “validate-template” дозволяє перевірити наявність синтаксичних помилок у шаблоні. Команда “deploy” запускає створення ресурсів.

Створимо мінімальний шаблон сторінки сайту та помістимо його в сховище для сайту. Подальше наповнення сторінки розглянь окремо.

```
% touch simple_site.html
% echo '<html><body><h1>Hello world!</h1></body></html>' > simple_site.html
% aws s3 cp ./simple_site.html s3://emer-appl-simple-site-data-
bucket/index.html
```

Також створений мінімальний код для відповіді на запити користувачів. У сховище помістимо заархівований код, оскільки його очікує процес розгортання функції.

```
% touch simple_lambda.js
% echo 'exports.handler = async (event) => { return { statusCode: 200,
body: "Hello world!"}};' > simple_lambda.js
% zip simple_lambda.zip simple_lambda.js
% aws s3 cp simple_lambda.zip s3://emer-appl-lambda-code-
bucket/simple_lambda.zip
```

У якості статичного контенту використаний стандартний рисунок, що відображає відсутність контенту. Прикладом такого рисунку є рисунок 3.8. Для цілей даного етапу цього буде достатньо.

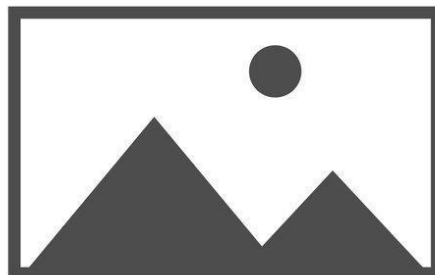


Рисунок 3.8 - Картинка для сайту

Скрипт для загрузки рисунку виглядає наступним чином.

```
% aws s3 cp placeholder.jpeg s3://emer-appl-static-content-
bucket/static/placeholder.jpeg
```

Надалі створена функція Lambda. Для того щоб функція могла виконувати дії, їй потрібно створити відповідну роль. Доки виконання коду не потребує ніяких додаткових доступів до інших сервісів та ресурсів, роль буде визначена у найпростіший спосіб.

```

AWSTemplateFormatVersion: '2010-09-09'
Description: The simple Lambda function
Resources:
  LambdaIAMRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Action:
              - sts:AssumeRole
            Effect: Allow
            Principal:
              Service:
                - lambda.amazonaws.com
  Function:
    Type: AWS::Lambda::Function
    Properties:
      Handler: simple_lambda.handler
      Role: !GetAtt LambdaIAMRole.Arn
      Code:
        S3Bucket: emer-appl-lambda-code-bucket
        S3Key: simple_lambda.zip
      Runtime: nodejs20.x
      Timeout: 5
  Outputs:
    SimpleLambdaFunctionArn:
      Description: The ARN of a simple lambda function
      Value: !GetAtt Function.Arn
      Export:
        Name: Emer-LambdaFunctionArn

```

Для створення стеку, що передбачає впровадження додаткових ролей доступу необхідно, щоб користувач мав відповідні права, а також вказати додатковий флаг “--capabilities CAPABILITY_IAM”.

```

% aws cloudformation deploy --stack-name emer-operational-lambdas --
template-file simple_lambda.yaml --capabilities CAPABILITY_IAM

```

Вихідним параметром шаблону є ARN (Amazon Resource Name) функції. ARN є унікальним ідентифікатором, який використовується для ідентифікації ресурсів у сервісах AWS.

Наступним кроком створимо API Gateway. Для формування точок доступу

потрібен опис методів доступу, а також посилання на Lambda функцію, яка буде виконувати обчислення. Фактично виконується зв'язок між API та виконанням функцій.

```

AWSTemplateFormatVersion: '2010-09-09'
Description: API Gateway for lambda calls
Resources:
  ApiGateway:
    Type: AWS::ApiGateway::RestApi
    Properties:
      Name: Emer-Api-Gateway
  ApiGatewayResource:
    Type: AWS::ApiGateway::Resource
    DependsOn:
      - ApiGateway
    Properties:
      ParentId: !GetAtt
        - ApiGateway
        - RootResourceId
      PathPart: hello
      RestApiId: !Ref ApiGateway
  ApiGatewayMethod:
    Type: 'AWS::ApiGateway::Method'
    DependsOn:
      - ApiGatewayResource
    Properties:
      RestApiId: !Ref ApiGateway
      ResourceId: !Ref ApiGatewayResource
      HttpMethod: ANY
      AuthorizationType: NONE
      Integration:
        Type: AWS_PROXY
        IntegrationHttpMethod: POST
        Uri: !Sub
          - "arn:aws:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/${LambdaFunctionArn}/invocations"
          - LambdaFunctionArn: !ImportValue Emer-LambdaFunctionArn

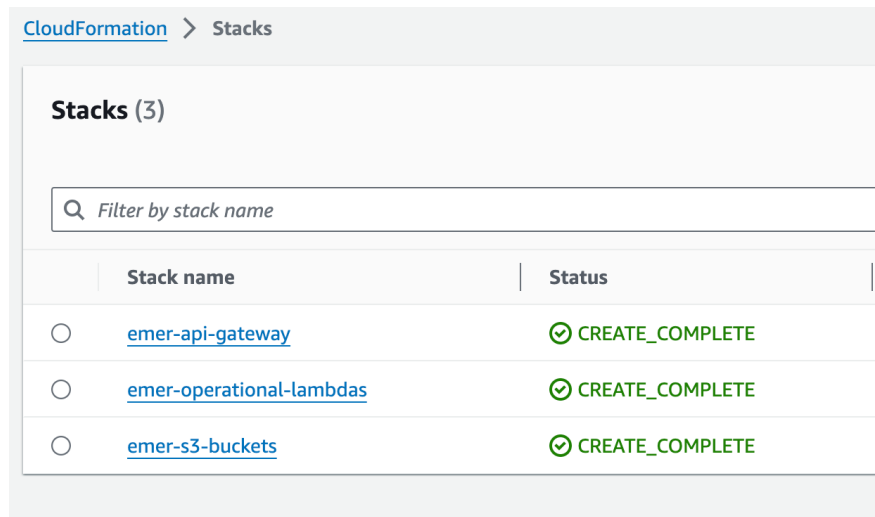
  ApiGatewayLambdaInvokePermission:
    Type: AWS::Lambda::Permission
    DependsOn:
      - ApiGateway
    Properties:
      Action: lambda:InvokeFunction
      FunctionName: !ImportValue Emer-LambdaFunctionArn
      Principal: apigateway.amazonaws.com
      SourceArn: !Sub
        - "arn:aws:execute-
api:${AWS::Region}:${AWS::AccountId}:${GatewayId}/*/*/hello"
        - GatewayId: !Ref ApiGateway
  ApiGatewayDeployment:
    Type: AWS::ApiGateway::Deployment
    DependsOn:
      - ApiGatewayMethod
    Properties:
      RestApiId: !Ref ApiGateway
      StageName: test
Outputs:
  ApiGatewayId:
    Description: The ID of a Emer API Gateway

```

```
Value: !Ref ApiGateway
Export:
  Name: Emer-ApiGatewayId
```

Вихідним параметром шаблону є унікальний ідентифікатор гейту, який потрібен для формування домену доступу при сполученні API Gateway з CloudFront.

Після виконання зазначених дій, отримане створене середовище комунікації. Так в нас підготовлений інтернет сайт, сховище статичного контенту та API взаємодії. Створені стеки ресурсів зазначені на рисунку 3.9.



The screenshot shows the AWS CloudFormation console 'Stacks' page. It displays a list of three stacks, all of which are in the 'CREATE_COMPLETE' state. The stacks are named 'emer-api-gateway', 'emer-operational-lambdas', and 'emer-s3-buckets'. Each stack has a radio button to its left and a green checkmark icon followed by the text 'CREATE_COMPLETE' to its right.

Stack name	Status
emer-api-gateway	✔ CREATE_COMPLETE
emer-operational-lambdas	✔ CREATE_COMPLETE
emer-s3-buckets	✔ CREATE_COMPLETE

Рисунок 3.9 - Стеки створення ресурсів на AWS

Тож слід створити можливість для користувачів комунікувати із зазначеною інфраструктурою.

В зазначеній у підрозділі 3.2 цільовій діаграмі розгортання зазначена Lambda функція, яка виконується в локації CDN. Для такої цілі використаний сервіс Lambda@Edge. Lambda@Edge - це функція AWS, яка дозволяє запускати функції Lambda для налаштування контенту, який доставляє CloudFront, виконуючи ці функції у розташуваннях AWS, які знаходяться ближче до кінцевого користувача.

Для цілей даної роботи розроблено дві функції. Одна функція буде обробляти запити, а інша - відповіді.

Далі наведений шаблон розгортання таких функцій. Загальний шаблон

зазначає створення ролі доступу, а також версій функцій.

```

AWSTemplateFormatVersion: '2010-09-09'
Description: The simple Lambda for edge purposes
Resources:
  LambdaIAMRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Action:
              - sts:AssumeRole
            Effect: Allow
            Principal:
              Service:
                - lambda.amazonaws.com
                - edgelambda.amazonaws.com
    ...

  LambdaAtEdgeResponseVersion:
    Type: 'AWS::Lambda::Version'
    Properties:
      FunctionName: !Ref ResponseFunction
  LambdaAtEdgeRequestVersion:
    Type: 'AWS::Lambda::Version'
    Properties:
      FunctionName: !Ref RequestFunction
Outputs:
  LambdaAtEdgeRequestVersionArn:
    Description: The ARN of a lambda for request edge function
    Value: !Ref LambdaAtEdgeRequestVersion
    Export:
      Name: Emer-LambdaAtEdgeRequestVersionArn
  LambdaAtEdgeResponseVersionArn:
    Description: The ARN of a lambda version for response edge function
    Value: !Ref LambdaAtEdgeResponseVersion
    Export:
      Name: Emer-LambdaAtEdgeResponseVersionArn

```

Головним вихідним значенням шаблону є ARN поточних версій функцій.

CloudFormation шаблони дозволяють визначити код функцій всередині шаблону. Це не є зручним механізмом для розробників, однак коли код не є великим на не передбачається його зміна в майбутньому такий підхід може стати в нагоді, оскільки не передбачає додаткових ресурсів та окремих команд для розгортання.

Тож можна визначити функції наступним чином:

...

```

ResponseFunction:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Role: !GetAtt LambdaIAMRole.Arn
    Code:
      ZipFile: |
        exports.handler = async (event, context, callback) => {
          const response = event.Records[0].cf.response;
          const headers = response.headers;
          headers['demonstration-header'] = [{key: 'Demonstration-Header', value: 'we-are-here'}];
          callback(null, response);
        };
    Runtime: nodejs20.x
    Timeout: 5
RequestFunction:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Role: !GetAtt LambdaIAMRole.Arn
    Code:
      ZipFile: |
        exports.handler = (event, context, callback) => {
          const request = event.Records[0].cf.request;
          request.uri = request.uri.replace(/^\/[\^\/]+\//, '/');
          return callback(null, request);
        };
    Runtime: nodejs20.x
    Timeout: 5
...

```

Маючи необхідні ресурси, можна створити шаблон розгортання CDN для доступу до них зі сторони користувачів.

Оскільки таке розгортання потребує додаткових прав для комунікації всередині системи, буде розглянутий шаблон розгортання частинами.

Розглянемо ресурс `OriginAccessControl`. Ця сутність створює нове джерело керування доступом у `CloudFront`. Створивши елемент керування доступом до джерела, слід додати його до джерела в розповсюдженні `CloudFront`, щоб `CloudFront` надсилав автентифіковані (підписані) запити до джерела. Таким чином, не зважаючи на те, що у сховищах `S3` не налаштовані окремі права доступу та вони є приватними, `CloudFront` матиме право розповсюджувати дані.

```

...
Resources:
  CloudFrontSiteOriginAccessControl:
    Type: AWS::CloudFront::OriginAccessControl
    Properties:
      OriginAccessControlConfig:
        Name: cloud-front-OAC-site
        OriginAccessControlOriginType: s3
        SigningBehavior: always
        SigningProtocol: sigv4
  CloudFrontDataOriginAccessControl:
    Type: AWS::CloudFront::OriginAccessControl
    Properties:
      OriginAccessControlConfig:
        Name: cloud-front-OAC-data
        OriginAccessControlOriginType: s3
        SigningBehavior: always
        SigningProtocol: sigv4
  WebApplicationFirewallCommonRules:
    Type: AWS::WAF::WebACL
    Properties:
      DefaultAction:
        Type: ALLOW
      MetricName: EmerWebACL
      Name: EmerWAF
...

```

Додатковою сутністю є також WAF (Web Application Firewall) WebACL (Web Access Control List). Завдяки конфігурації контролю доступу можна обмежити права користувачів виходячи з їх IP адрес, контролювати ін'єкції коду (такі як SQL, XSS та інші), створювати власні правила, та контролювати інші обмеження. У даному випадку не встановлено додаткових обмежень, але вони можуть бути налаштовані окремо.

Розглянемо розгортання саме CDN. CloudFront дозволяє налаштувати загальні правила доступу, регіональні обмеження, загальні правила кешування та багато інших параметрів. Також треба зазначити асоціації з WAF. Для поведінки кешу за замовчуванням також визначена асоціація з Lambda, що обробляє відповіді.

```

...
Resources:
  CloudFrontDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        WebACLId:

```

```

    Ref: "WebApplicationFirewallCommonRules"
    Origins:
    ...
    Enabled: "true"
    HttpVersion: http1.1
    PriceClass: PriceClass_100
    DefaultRootObject: index.html
    DefaultCacheBehavior:
      AllowedMethods:
        - GET
        - HEAD
        - OPTIONS
      ForwardedValues:
        QueryString: "false"
        Cookies:
          Forward: none
      ViewerProtocolPolicy: redirect-to-https
      TargetOriginId: static-site-hosting
      LambdaFunctionAssociations:
        - EventType: viewer-response
          IncludeBody: false
          LambdaFunctionARN: !ImportValue Emer-
LambdaAtEdgeResponseVersionArn
      CacheBehaviors:
    ...
    Restrictions:
      GeoRestriction:
        RestrictionType: whitelist
        Locations:
          - AQ
          - UA
    ...

```

Окремо слід розглянути визначення джерел даних та правил кешування. Так джерелом можуть бути як сховища даних S3 так і інші сервіси (в даному випадку API Gateway). Різні типи джерел конфігуруються окремими параметрами. Для розуміння яке саме джерело використовується для поточного запиту використовується параметр “OriginPath”. Таким чином для URI, що починаються з зазначеного шляху, буде обрано відповідне джерело.

```

...
...
    Origins:
      - DomainName: !ImportValue Emer-SiteDataBucketDomainName
        Id: static-site-hosting
        OriginAccessControlId: !GetAtt
CloudFrontSiteOriginAccessControl.Id
        S3OriginConfig:
          OriginAccessIdentity: ""
      - DomainName: !ImportValue Emer-AppStaticContentBucketDomainName
        Id: static-data-hosting

```

```

OriginPath: /static
OriginAccessControlId: !GetAtt
CloudFrontDataOriginAccessControl.Id
S3OriginConfig:
  OriginAccessIdentity: ""
- DomainName: !Sub
  - "${GatewayId}.execute-api.${AWS::Region}.amazonaws.com"
  - GatewayId: !ImportValue Emer-ApiGatewayId
Id: operational-api
OriginPath: /api
CustomOriginConfig:
  HTTPSPort: 443
  OriginProtocolPolicy: https-only
...

```

Для того, щоб CloudFront міг обслуговувати об'єкти з усіх джерел, необхідно створити принаймні стільки поведінок кешу (включаючи поведінку кешу за замовчуванням), скільки існує джерел. Слід зазначити, що AWS надає можливість обрати існуючі типи поведінки, які відповідають середньостатистичним вимогам різних типів проектів. Так, наприклад, для API Gateway, зазвичай не потрібно використання кешу, для чого створена відповідна стратегія кешування з назвою “CachingDisabled” та ідентифікатором “4135ea2d-6df8-44a3-9df3-4b5a84be39ad”.

Тож поведінка кешу для проекту сконфігурована наступним чином:

```

...
CacheBehaviors:
- AllowedMethods:
  - GET
  - HEAD
  - OPTIONS
  ForwardedValues:
    QueryString: "false"
    Cookies:
      Forward: none
  ViewerProtocolPolicy: redirect-to-https
  TargetOriginId: static-data-hosting
  PathPattern: /static/*
  LambdaFunctionAssociations:
  - EventType: viewer-request
    IncludeBody: false
    LambdaFunctionARN: !ImportValue Emer-
LambdaAtEdgeRequestVersionArn
- AllowedMethods:
  - GET
  - HEAD
  - OPTIONS
  - PUT
  - PATCH
  - POST

```

```

- DELETE
ViewerProtocolPolicy: redirect-to-https
TargetOriginId: operational-api
PathPattern: /test/*
CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad
OriginRequestPolicyId: b689b0a8-53d0-40ab-baf2-68738e2966ac
ResponseHeadersPolicyId: 60669652-455b-4ae9-85a4-c4c02393f86c
LambdaFunctionAssociations:
  - EventType: viewer-request
    IncludeBody: false
    LambdaFunctionARN: !ImportValue Emer-
LambdaAtEdgeRequestVersionArn

```

Вихідним параметром шаблону є доменна адреса CDN дистрибуції. За цією адресою будуть доступні ресурси додатку. Оскільки усі джерела розповсюджуються через єдину мережу дистрибуції, такий підхід дозволяє уникнути додаткових проблем з конфігурацією CORS (Cross Origin Resource Sharing).

Як вже зазначалося раніше, різні джерела даних доступні за різними початковими шляхами URI (Uniform Resource Identifier). Таким чином, маючи загальну доменну адресу та конфігурацію джерел ресурсів за URI, можна отримати доступ до різних частин додатку.

```

...
Outputs:
  CloudFrontDomainName:
    Description: Domain name of a DistributionConfig
    Value: !GetAtt CloudFrontDistribution.DomainName
    Export:
      Name: Emer-CloudFrontDomainName

```

3.5 Підсумки та висновки щодо практичної реалізації

Головним вихідним параметром практичної реалізації є доменне ім'я дистрибуції ресурсів проекту як зазначено на рисунку 3.10. За цим ім'ям доступні всі ресурси, які забезпечують взаємодію з користувачем.

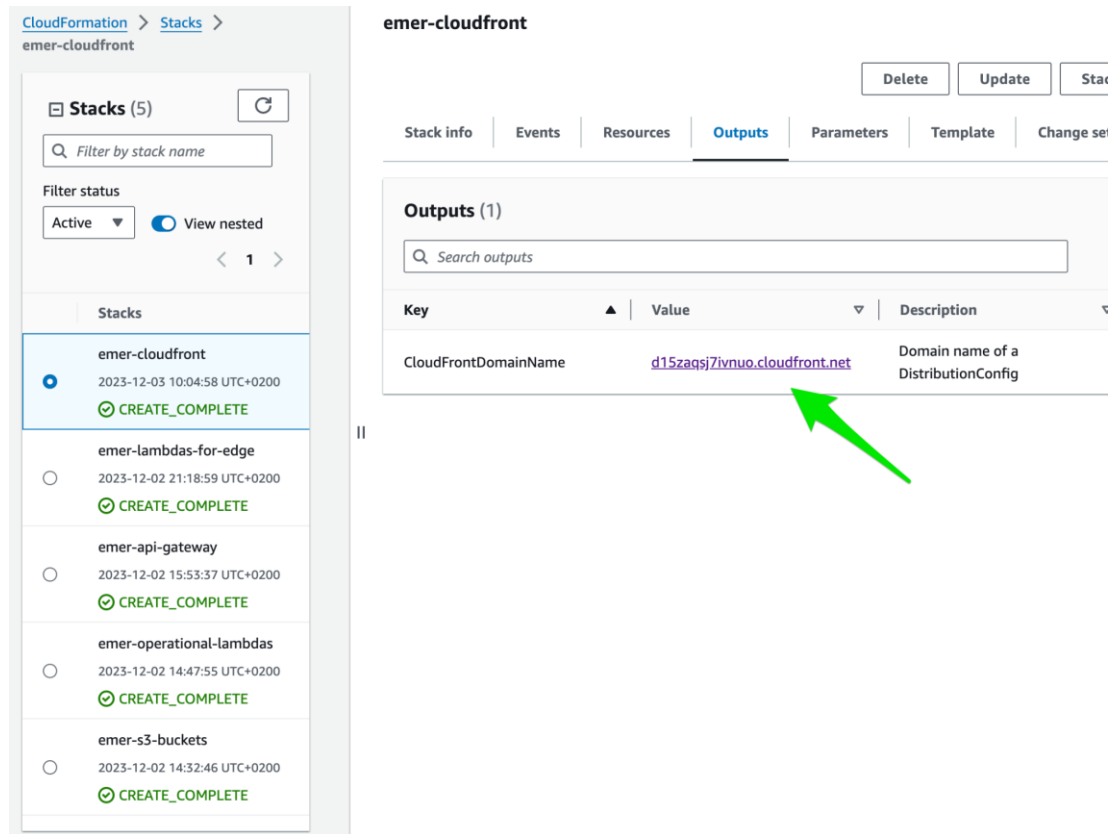


Рисунок 3.10 - Загальний стек ресурсів практичної реалізації CDN

Для перевірки доступності ресурсів, модифікуємо сторінку сайту та надамо їй вигляд який включає статичний контент, а також виклик до API. Модифікований код сайту може мати наступний вигляд.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Emergency Medicine Enterprise</title>
  </head>
  <script>
    async function callApi() {
      try {
        const response = await fetch('/test/hello');
        const data = await response.text();
        document.getElementById('apiResponse').innerText = data;
      } catch (error) {
        console.error('Error fetching API:', error);
        document.getElementById('apiResponse').innerText = 'Failed to fetch
API';
      }
    }
    async function changeImage() {
      const image = document.getElementById('staticImage');
      const source = image.src;
      if (source.endsWith("placeholder.jpeg")) {
        image.src = "/static/HealthCare.png";
      } else {
```

```

        image.src = "/static/placeholder.jpeg";
    }
}
</script>
</head>
<body>
    <h1>Welcome to the Emergency Medicine Enterprise Website</h1>
    
    <div><button onclick="callApi()">Call API</button>&nbsp;<button
onclick="changeImage()">Change Image</button></div>
    <p id="apiResponse"></p>
</body>
</html>

```

Для внесення змін на сайт достатньо завантажити новий код до S3.

```
% aws s3 cp ./emergency_medicine_website.html s3://emer-appl-simple-site-data-bucket/index.html
```

Оновлені дані стануть доступні через деякий після інвалідації кеша CloudFront. У разі коли треба оновити дані негайно, можна викликати інвалідацію достроково.

```
% aws cloudfront create-invalidation --distribution-id E55PP24G2BLKP --paths '/*'
```

На рисунку 3.11 бачимо, що сторінка доступна за посиланням на домен CloudFront дистрибуції. Також доступний статичний контент зі сховища на S3.

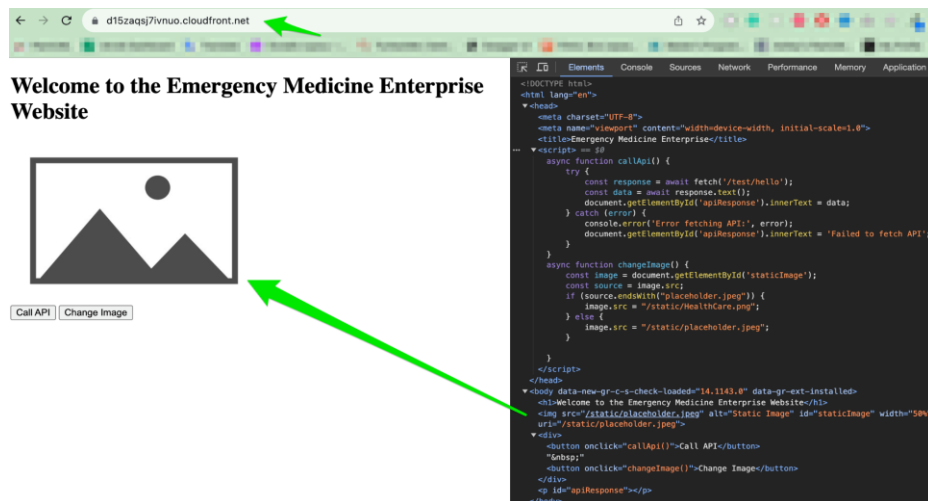


Рисунок 3.11 - Завантаження сайту з CDN

Рисунок 3.12 відображає можливість доступу до сховищ, а також можливості виклику API.

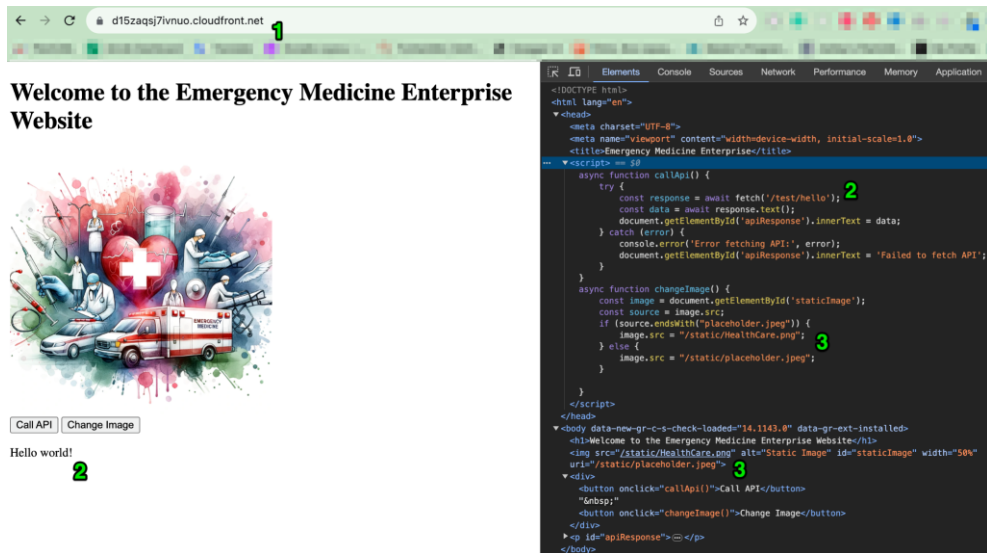


Рисунок 3.12 - Динамічна зміна контенту

Додатково на рисунку 3.12. слід зазначити:

- Первинне завантаження сторінки відбувається за посиланням на доменне ім'я дистрибуції;
- Виклик API відбувається через відносне ім'я ресурсу;
- Статичний контент завантажується через відносне ім'я ресурсу.

Практичне застосування скриптів IaC в рамках AWS CloudFormation відкриває широкі можливості для повторного використання коду та спрощення процесу внесення змін в інфраструктуру. Скрипти CloudFormation дозволяють архітекторам систем створювати модульні та параметризовані шаблони, які можуть бути легко адаптовані під нові вимоги проекту або повторно застосовані в різних середовищах розгортання. Завдяки декларативному підходу, внесення змін у вже існуючу інфраструктуру відбувається швидко та контрольовано, що знижує ризики при оновленні та масштабуванні систем.

4 ВИСНОВКИ ТА ПЕРСПЕКТИВИ РОЗВИТКУ ПРОГРАМНИХ РІШЕНЬ ДЛЯ МЕДИЦИНИ ТА МЕДИЦИНИ НАДЗВИЧАЙНИХ СИТУАЦІЙ

Хмарні технології роблять медичні послуги більш доступними, особливо у віддалених або недостатньо обслуговуваних регіонах. Вони дозволяють медичним працівникам здійснювати віддалений моніторинг стану пацієнтів та надавати телемедичні консультації, що знижує потребу у фізичному відвідуванні лікарень. Зберігання та обробка медичних даних у хмарі вимагає високого рівня безпеки, особливо з огляду на конфіденційність та захист персональних даних. Хмарні провайдери зазвичай пропонують розширені функції безпеки, включаючи шифрування даних, механізми аутентифікації та авторизації, що допомагає забезпечити захист від несанкціонованого доступу та інших кіберзагроз.

4.1 Розвиток хмарних технологій в світі

Згідно висновків Deloitte [34], хмарні рішення є рушієм інновацій та прогресу як для окремих підприємств так і для держав в цілому. Висновки Deloitte зокрема підкреслюють, що увага до хмарних технологій стає ще більш актуальною у всьому світі. Тенденції розвитку хмарних рішень зазначають також видатні видання світу. Аналітики Gartner зазначають [35], що більше ніж 85% організацій до 2025 року дотримуватимуться принципу "cloud-first" і не зможуть повноцінно реалізувати свої цифрові стратегії без використання архітектур та технологій, орієнтованих на хмарні обчислення. Застосування хмарно-орієнтованих платформ означає, що цифрові або продуктові команди використовуватимуть архітектурні принципи та можливості, які максимально використовують властиві можливості хмарного середовища. Нові робочі навантаження, розгорнуті у хмарно-орієнтованому середовищі, стануть повсюдними, а не просто популярними, і все, що не є хмарним, буде розглядатися як застаріле.

Згідно з прогнозом Gartner [36], світові витрати кінцевих користувачів на публічні хмарні сервіси збільшаться на 20,4% і складуть 678,8 мільярда доларів у 2024 році, порівняно з 563,6 мільярда доларів у 2023 році. Одним з ключових трендів, що спонукає витрати на хмарні сервіси, є стійке зростання платформ індустріальних хмар. Індустріальні хмарні платформи спрямовані на досягнення бізнес-результатів, актуальних для певної галузі, шляхом комбінування базових послуг програмного забезпечення як сервісу (SaaS), платформ як сервісу (PaaS) та інфраструктури як сервісу (IaaS) в єдину продуктову пропозицію з можливістю комбонування функцій. Gartner прогнозує [36], що до 2027 року більше ніж 70% підприємств використовуватимуть індустріальні хмарні платформи для прискорення своїх бізнес-ініціатив, у порівнянні з менш ніж 15% у 2023 році.

Також, Gartner надає передбачення [35], що до 2025 року понад 95% нових цифрових робочих навантажень будуть розгорнуті на хмарно-орієнтованих платформах, що є значним зростанням з 30% у 2021 році. Змінюючись, модель управління організації зміниться на орієнтовану на продукти, де весь потік цінностей бізнесу та ІТ буде узгоджено за продуктами. Це створить нові ролі та відповідальності, такі як інженери надійності сайтів, менеджери продуктів чи спільноти практиків.

Експерти у сфері хмарних обчислень в Україні також підкреслюють невідворотність розгортання хмарних рішень у всіх секторах економіки, включаючи приватний та державний сегменти. Спеціалісти GigaCloud вважають [37], що на основі практичного осмислення переваг, які надає хмарна інфраструктура, державні інституції, імовірно, втратять зацікавленість у поверненні до використання власних серверних ресурсів. Організації, які наразі покладаються на фізичне серверне обладнання, можуть у зв'язку зі збільшенням потенційних ризиків, з часом перейти до використання віртуальних дата-центрів.

Аналітики MarketsAndMarkets прогнозують, що глобальний ринок хмарних обчислень зросте з \$626,4 мільярда в 2023 році до \$1,266,4 мільярда до 2028 року, зі щорічним зростанням на 15,1% протягом прогнозованого періоду [38].

Тенденції розвитку хмарних технологій підтверджуються також дослідженнями Data Bridge. Слід зазначити деякі з них:

- згідно з аналізом компанії Data Bridge Market Research, обсяг ринку персональних хмарних послуг у 2021 році був оцінений в 37,36 мільярда доларів США і очікується, що до 2029 року його вартість досягне 235,84 мільярда доларів США за середньорічним темпом зростання (CAGR) 25,90% у прогнозований період з 2022 по 2029 рік [39];

- аналітичний звіт компанії Data Bridge Market Research вказує, що ринок державних хмарних рішень, обсяг якого в 2021 році становив 27,32 мільярда доларів США, очікується зросте до 77,39 мільярда доларів США до 2029 року, з середньорічним темпом зростання (CAGR) 13,90% у прогнозований період з 2022 по 2029 рік [40];

- ринок гібридних хмар був оцінений у 47,67 мільярда доларів США у 2021 році і, за прогнозами, досягне 135,98 мільярда доларів США до 2029 року, з реєстрацією середньорічного темпу зростання (CAGR) на рівні 14,00% у прогнозований період з 2022 по 2029 рік [41].

Різноманітність аналітичних підходів та дослідницьких методологій, застосованих різними аналітичними та дослідницькими агенціями, призводить до розбіжностей у висновках щодо поточного стану ринків хмарних послуг. Незважаючи на методологічні відмінності, є консенсус серед провідних аналітичних компаній, який вказує на стабільне зростання ринку хмарних послуг. Це зростання оцінюється не менше ніж у 15% річно, що є істотним показником активізації інтересу як з боку споживачів послуг, так і суспільства загалом. Такий рівень зростання сигналізує про високу динаміку розвитку галузі та потенціал для інвестування і інновацій.

4.2 Загальносвітові тенденції розвитку програмних рішень в медицині

Глобальна ініціатива з цифрового здоров'я (GIDH) [42] є мережею, керованою ВООЗ, створеною для сприяння впровадженню Глобальної стратегії

цифрового здоров'я на 2020-2025 роки та інших норм і стандартів ВООЗ для трансформації систем цифрового здоров'я [43]. Ініціатива слугуватиме платформою, яка дозволить глобальній екосистемі колективно працювати над підвищенням потенціалу країн та зміцненням міжнародної співпраці в області цифрового здоров'я.

Ініціатива Всесвітньої організації охорони здоров'я "Ініціатива геолокації даних медичних установ" (Geolocated Health Facilities Data initiative - GHFD) [44] є одним з напрямів роботи ВООЗ, спрямованого на покращення доступності та якості медичних послуг шляхом використання геопросторових технологій. Основні аспекти цієї ініціативи включають:

- Збір та Аналіз Даних. Ініціатива передбачає збір та аналіз даних про розташування медичних установ по всьому світу;

- Геолокація та Картографування. Основна ідея полягає у використанні географічної інформаційної системи (ГІС) для точного картографування розташування медичних установ;

- Покращення Реагування на Надзвичайні Ситуації. Ініціатива має на меті підвищити ефективність реагування на надзвичайні ситуації, такі як епідемії, природні катастрофи або конфлікти;

- Підтримка Політики та Планування. Надаючи точні геопросторові дані, ця ініціатива допомагає урядам та міжнародним організаціям у прийнятті рішень щодо планування охорони здоров'я, розподілу ресурсів та розробки політик.

- Співпраця та Партнерство. Ініціатива може включати співпрацю з урядами країн, міжнародними організаціями, неприбутковими організаціями та приватним сектором для збору та аналізу даних.

Крім того, маркетингові дослідження провідних корпорацій вказують на глибокий інтерес суспільства до цифровізації медичних послуг, який підсилюється прогнозами значного зростання ринку цифрових медичних рішень. Наприклад, аналітики з MarketsAndMarkets надають прогнози щодо розвитку ринку цифрових медичних послуг. Слід зазначити деякі з них:

- глобальний ринок цифрових аптек за доходами було оцінено в \$96.5

мільярдів у 2022 році та прогнозується досягти \$211.9 мільярдів до 2027 року, з річним зростанням на 17.0% в період з 2022 по 2027 рік [45];

- розмір світового ринку Інтернету речей (IoT) у сфері охорони здоров'я прогнозується зросте з 127.7 мільярда доларів США в 2023 році до 289.2 мільярда доларів США до 2028 року, що відповідає середньорічному зростанню на 17.8% протягом прогнозованого періоду [46];

- глобальний ринок цифрових двійників у сфері охорони здоров'я (створення деталізованих віртуальних моделей, які точно відтворюють фізіологічні та біологічні характеристики пацієнтів або медичні процедури), виражений у доходах, був оцінений приблизно в 1,6 мільярда доларів США у 2023 році і, як передбачається, досягне 21,1 мільярда доларів до 2028 року, зростаючи зі середньорічними темпами 67,0% у період з 2023 по 2028 рік [47];

- глобальний ринок телемедицини та телездоров'я за доходами було оцінено приблизно в 120,4 мільярда доларів у 2023 році та очікується зростання до 285,7 мільярда доларів до 2028 року, з річним зростанням на 23,2% у період з 2023 по 2028 рік [48].

У контексті глобального переходу до цифрової економіки, особливу увагу привертають зусилля провідних цифрових корпорацій, таких як Amazon, Microsoft, Google, Oracle та інших, спрямовані на розробку та вдосконалення хмарних рішень для сектора охорони здоров'я.

Цифрові хмарні додатки та системи такі як Apple HealthKit [49], AWS for e-Health [50], Google Health [51], Microsoft eHealth Information Management [52], European Health Data Cloud [53], Dell Healthcare Cloud Computing Solutions [54] та інші, продовжують набирати обертів, задаючи нові стандарти в області медичного обслуговування та досліджень.

4.3 Розвиток програмних рішень для медицини в Україні

Україна робить значні кроки у напрямку цифровізації медицини, особливо з огляду на необхідність модернізації системи охорони здоров'я та підвищення

якості медичних послуг. Згідно Розпорядження Кабінету Міністрів України від 17.01.2018 р. № 67-р [55] “Цифровізація медицини є життєвою необхідністю її розвитку та ефективного надання медичних послуг. Медична інформація, що міститься в медичних паперових картках, повинна залишитися в минулому.”.

Надалі розглянемо деякі з ключових ініціатив.

Електронні Медичні Записи (ЕМЗ) (EHR). Україна активно працює над впровадженням та розвитком системи електронних медичних записів. Це включає створення централізованих баз даних, які містять медичну інформацію про пацієнтів, включаючи анамнез, діагнози, призначення та історію лікування.

За даними Національної служби здоров'я України [56] з початку розвитку програми (2017 рік) було створено 2.4 мільярди записів. Тенденція розвитку електронних записів в Україні відображена на рисунку 4.5.

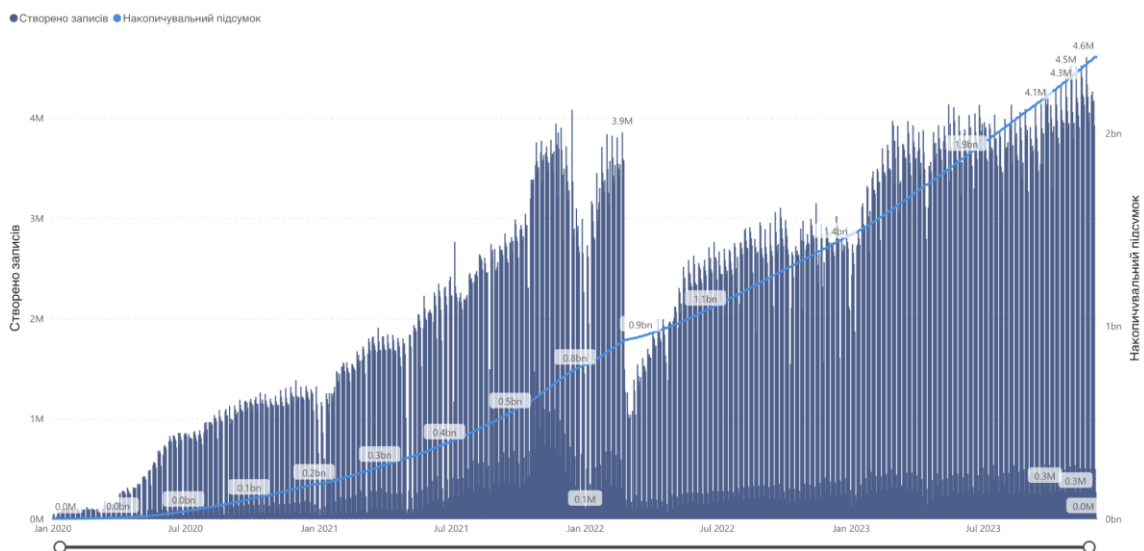


Рисунок 4.5 - Статистика створення Електронних медичних записів в Україні

Електронні лікарняні листи. Україна розпочала процес цифровізації лікарняних листів, що дозволяє пацієнтам отримувати офіційні документи про тимчасову непрацездатність в електронному форматі.

З 1 жовтня 2021 року усі заклади охорони здоров'я України перейшли на оформлення електронних лікарняних [57]. Працівники, захворівши, мають повідомляти роботодавцю лише номер електронного лікарняного.

З початку дії програми в Україні створено більше 23 мільйонів записів [56],

як наведено на рисунку 4.6.

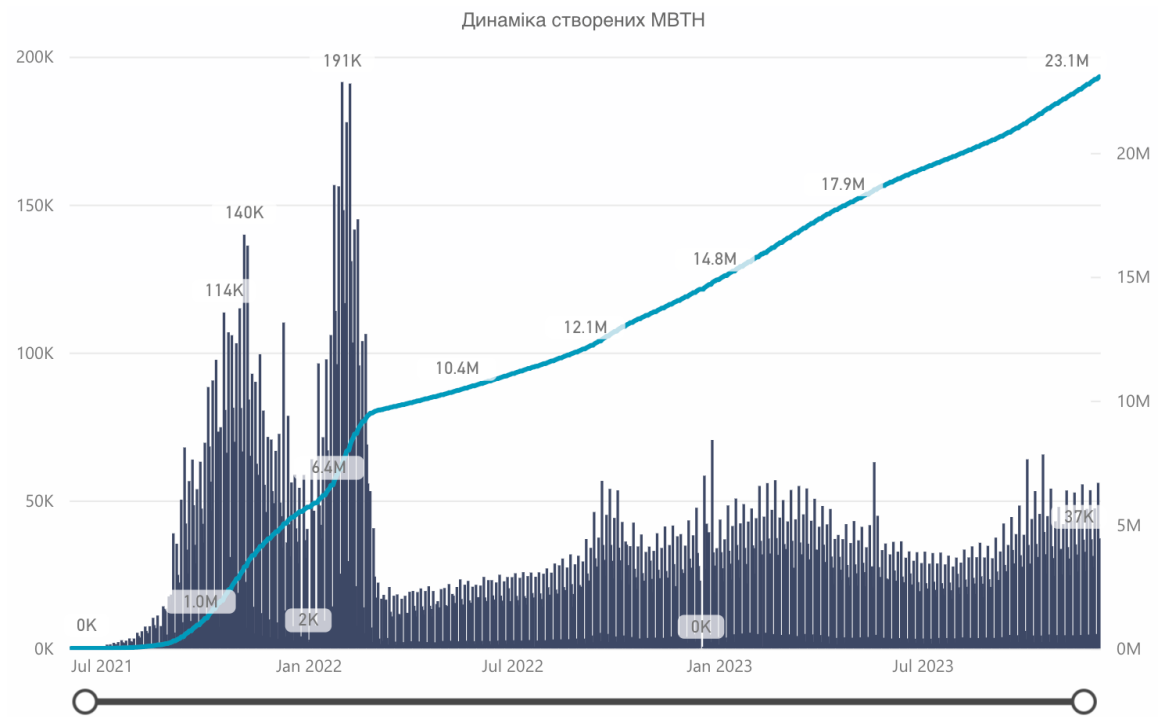


Рисунок 4.6 - Статистика створених медичних виписок про тимчасову непрацездатність

Система "Електронний рецепт". Ця система дозволяє лікарям виписувати рецепти електронно, що полегшує процес отримання медикаментів для пацієнтів та підвищує контроль за виписуванням ліків.

В Україні е-рецепт запрацював у квітні 2019 року в рамках відпуску препаратів за програмою реімбурсації "Доступні ліки". Так, пацієнти з серцево-судинними захворюваннями, бронхіальною астмою чи діабетом II типу можуть отримати ліки безоплатно або з незначною доплатою за електронним рецептом. Починаючи з 1 квітня 2023 року е-рецепт поширюється на всі зареєстровані в Україні лікарські засоби, що підлягають відпуску за рецептом лікаря. З початку дії програми в Україні виписано понад 66 мільйонів електронних рецептів [56]. Розподіл за регіонами відображений на рисунку 4.7.

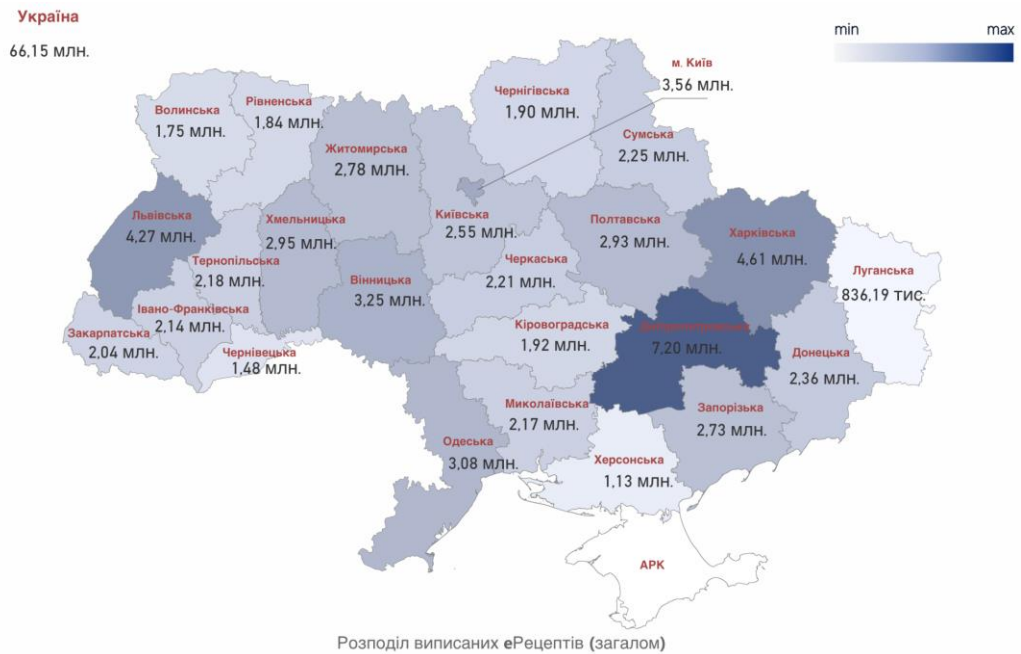


Рисунок 4.7 - Розподіл виписаних електронних рецептів за регіонами України

Мобільний додаток "Дія". Уряд України інтегрував медичні послуги в мобільний додаток "Дія" [58], що дозволяє користувачам отримувати доступ до своїх медичних даних, записуватися на прийом до лікаря та отримувати інші медичні послуги онлайн.

Телемедицина. Розвиток та впровадження телемедичних послуг на рівні держави [59][60], особливо важливих для віддалених та сільських районів, де доступ до медичних спеціалістів може бути обмежений.

Напрямок телемедицини на рівні держави є в активній розробці та впровадження телемедичних рішень в Україні відбувається за технічної підтримки проекту USAID «Сталий розвиток національних систем охорони здоров'я в Україні» (LHSS) [61].

Навчання медичних працівників. Програми з підвищення кваліфікації та навчання медичних працівників на рівні держави з використанням цифрових технологій та систем. Платформа Академії НСЗУ (Національна Служба Здоров'я України) [62] – електронна веб платформа для розміщення онлайн курсів, вебінарів, новин та іншої корисної інформації стосовно підвищення кваліфікації лікарів, медичного та адміністративного персоналу закладів охорони здоров'я, розвитку їх професійних навичок та особистісних якостей, а також

інформування пацієнтів про медичні послуги за програмою медичних гарантій.

Також, згідно з постановою Кабінету Міністрів України №126 [63] та Ліцензійних умов провадження господарської діяльності з медичної практики [64] медичні установи України зобов'язані “дотримуватися вимог до ведення форм первинної облікової документації, затверджених МОЗ, а також вимог, передбачених Порядком функціонування електронної системи охорони здоров'я”.

Ключовою державною системою щодо консолідації медичних даних є програмний комплекс eHealth [65][66].

Українська державна система eHealth, яка впроваджується Державним підприємством “Електронне Здоров'я”, є ключовою частиною реформи охорони здоров'я в Україні, спрямованої на цифровізацію медичних послуг і підвищення якості та доступності медичного обслуговування. Зазначені напрями розвитку цифрових технологій для медицини є не повним переліком функціоналу системи, який є значно ширшим.

Health24 [67], впроваджена Компанією «Здоров'я 24», яка входить до складу групи компаній E-Tech, є однією з ключових цифрових платформ у сфері охорони здоров'я в Україні. Ця система була розроблена з метою поліпшення доступу до медичних послуг та підвищення ефективності роботи медичних установ. Health24 розроблена за міжнародними стандартами електронного медичного документообігу, відповідає вимогам чинного українського законодавства та є акредитованим оператором підключення до системи eHealth (Наказ ДП “Електронне здоров'я” №6-ГД від 02.04.2018 не є публічним). Система об'єднує інформацію різних медичних установ та інформацію eHealth, що відображено на рисунку 4.8.

КЛЮЧОВІ РОЛІ В СИСТЕМІ ЕЛЕКТРОННОГО ЗДОРОВ'Я



Рисунок 4.8 - Ключові ролі в системі електронного здоров'я в Україні

Значні досягнення України на шляху цифровізації медичних послуг обумовлені зокрема використанням хмарних технологій. Так Закон “Про хмарні послуги” [68] врегулював правові відносини, що виникають при наданні хмарних послуг, та встановив особливості використання хмарних послуг органами державної влади. Таким чином надавши державним цифровим ініціативам доступ до сучасних потужностей хмарних сервісів та рішень.

Такі хмарні провайдери України як GigaCloud [69] зробили важливий внесок у розвиток технологічних можливостей медицини України забезпечуючи стабільний сервіс з одночасним виконанням вимог щодо безпеки даних у відповідності з державною нормативною базою.

Додатково, Постановою КМУ №263 [70] та Законом України №2130-IX [71] врегульовано питання можливості тимчасового розміщення даних в закордонних датацентрах та використання хмарних провайдерів інших держав.

Ці законодавчі ініціативи забезпечили основу для інтеграції хмарних технологій у сферу охорони здоров'я, що привело до підвищення масштабованості та гнучкості, зниження оперативних витрат, покращення доступності та безпеки, відповідності міжнародним стандартам.

За останні роки Україна досягла значного прогресу у розвитку цифрових технологій в медицині. Однак, згідно Стратегії розвитку системи охорони здоров'я до 2030 року Міністерства охорони здоров'я України [72], недостатня цифровізація все ще формує низку проблемних ситуацій, які можуть бути вирішені завдяки подальшій роботі над програмними продуктами для медицини.

Також зазначається, що “Одним із ключових принципів розвитку даних сфери охорони здоров'я є пріоритет цифрової форми. При проектуванні нових процесів та сервісів у системі охорони здоров'я слід переосмислювати існуючу роботу з даними й інформацією та виходити не від наявних паперових форм медичної звітності, а від переліку даних, які потрібні для подальшого ухвалення рішень або автоматизованої роботи цифрових компонентів.”

4.4 Висновки щодо подальшого розвитку використання хмарних рішень в медицині та медицині надзвичайних ситуацій

У сфері медицини надзвичайних ситуацій, хмарні рішення можуть відіграти ключову роль у покращенні координації між медичними установами, оптимізації розподілу ресурсів та швидкому доступі до критично важливої інформації. Це сприятиме більш ефективному реагуванню на надзвичайні ситуації, зменшенню часу реагування та покращенню результатів лікування.

Так дослідники з Медичної школи Луїсвільського університету з колегами задаються питанням щодо майбутнього медицини надзвичайних ситуацій [73]. Одним з ключових технологічних аспектів розвитку медицини надзвичайних ситуацій зазначається телемедицина. Зазначається, що ця галузь продовжуватиме інтегрувати нові технології, як у клінічній практиці, так і в процесі навчання майбутніх лікарів. Телемедицина забезпечує кращий доступ до медичних послуг та швидко набуває популярності в невідкладних відділень США.

Дослідження Deloitte [6] звертає увагу на майбутнє електронних медичних записів (ЕМЗ). Зазначається, що у майбутньому електронні медичні записи

можуть включати системи амбієнтного прослуховування, голограми та концепцію лікарень у домашніх умовах. Існує думка, що насувається період змін, який призведе до виникнення нового покоління ЕМЗ. Майбутні рішення ЕМЗ мають бути розроблені так, щоб враховувати потреби всіх користувачів, особливо потреби споживачів та медичних працівників. Також вони мають використовувати найсучасніші технології, такі як хмарні рішення та екосистемні платформи, для покращення здоров'я та добробуту споживачів.

Приклад Українського інституту майбутнього [75] зазначає стрімкий розвиток цифровізації України в цілому та медицини в особливості. Також зазначається необхідність державної підтримки таких трансформацій. Завдяки активній участі держави в трансформації медицини вважається [75], що вдалося досягти “цифрового стрибка” в цифровізації, що проілюстровано рисунком 4.9.



Рисунок 4.9 - Приклад цифрового розвитку медицини

Також, слід зазначити зусилля міжнародних коаліцій компаній таких як Transform Health [76]. Transform Health — це коаліція організацій, які виступають за справедливу цифрову трансформацію систем охорони здоров'я — для досягнення здоров'я для всіх. Ініціатива Transform Health була заснована у 2019 році сімома організаціями, які визнали необхідність злагоджених та довгострокових зусиль для використання перетворювальної сили цифрових

технологій та даних з метою розширення первинної медичної допомоги та досягнення цілі сталого розвитку щодо Всеосяжного Медичного Забезпечення до 2030 року. Спільними зусиллями компанії розробляють загальні стратегії та рекомендації щодо напрямів інвестицій та розвитку у сфері цифровізації медицини [77].

Важливим аспектом є також забезпечення безпеки та конфіденційності даних, особливо у контексті зростаючих вимог до захисту персональної інформації пацієнтів. Розвиток стандартів та протоколів безпеки, а також впровадження передових методів шифрування та контролю доступу є критично важливими для підтримки довіри та надійності хмарних медичних рішень. Відповідний графік наведено на рисунку 4.10.

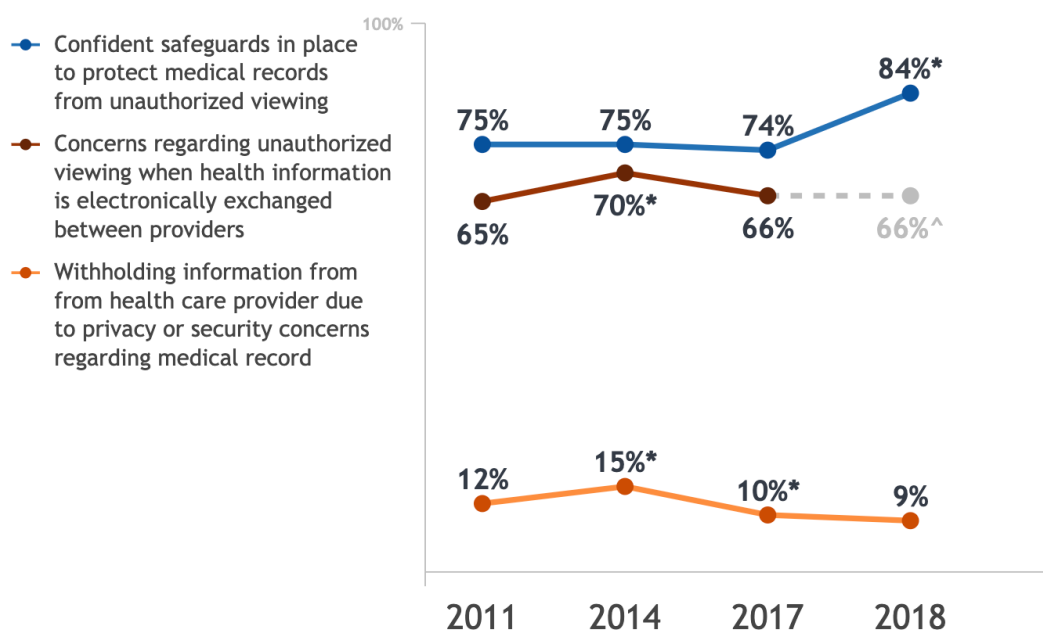


Рисунок 4.10 - Статистика довіри до електронних медичних записів

За даними HealthIT [78] щодо ринку цифрових медичних записів в США, більшість людей (84%) переконані у безпеці своїх медичних записів від несанкціонованого доступу, однак висловлюють занепокоєння (66%), коли медична інформація обмінюється в електронному вигляді. В порівнянні з 2017 роком, зараз більше осіб впевнені у захищеності своїх записів від несанкціонованого перегляду. Менше десяти відсотків респондентів повідомили про утримання інформації від свого медичного працівника через побоювання

щодо приватності та безпеки своїх медичних записів. Це зниження з 15% осіб у 2014 році.

Безперечно, слід зазначити розвиток штучного інтелекту. Як зазначалося раніше, штучний інтелект відіграє важливу роль у розвитку медицини в цілому та медицини надзвичайних ситуацій в особливості. За допомогою штучного інтелекту сучасні медичні додатки створюють та аналізують як набори медичних даних, лабораторних аналізів, медичних знімків, так і створюють для подальшого аналізу цифрові копії окремих органів, вірусів та бактерій, ліків. У своїй роботі [79] Доктор Фред Мілґрім (Fred Milgrim) розбирає питання щодо можливостей штучного інтелекту та чат-ботів на його основі. Загальним висновком є те, що штучний інтелект розкриває додаткові можливості та може стати гарним помічником лікарю, але не його заміною.

Особливу увагу слід приділити долученню держав до цифровізації медицини в цілому. Робота держави як законодавчого інституту повинна йти поруч з розвитком технологій. Крім загальних ініціатив ВООЗ [42] слід також звертати увагу на досвід окремих країн [80][81][82], які можуть надати розуміння деяких перешкод та проблем на шляху цифровізації держави та її інституцій.

Загалом, подальша цифровізація та подальший розвиток хмарних технологій у медицині та медицині надзвичайних ситуацій відкриває широкі можливості для підвищення ефективності, доступності та якості медичних послуг, сприяючи розвитку інноваційних підходів у охороні здоров'я.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Hoover C. Electronic Medical Record (EMR) System Review. Parsons Institute for Information Mapping. URL: http://piim.newschool.edu/_media/pdfs/PIIM-RESEARCH_CommercialEMRSystemsReview.pdf (date of access: 30.10.2023).
2. Narrative Review for Exploring Barriers to Readiness of Electronic Health Record Implementation in Primary Health Care / S. Hakiem Afrizal et al. Healthcare Informatics Research. URL: <https://e-hir.org/journal/view.php?id=10.4258/hir.2019.25.3.141> (date of access: 30.10.2023).
3. Ishigure Y. Trends, Standardization, and Interoperability of Healthcare Information | NTT Technical Review. NTT Technical Review. URL: <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201104gls.html> (date of access: 30.10.2023).
4. Quickstats | HealthIT.gov. ONC | Office of the National Coordinator for Health Information Technology. URL: <https://www.healthit.gov/data/quickstats> (date of access: 18.10.2023).
5. Electronic Medical Records (EMR) Market Growth & Global Trends to 2029. Market Research Business Consulting and Strategy Planning Firm | Data Bridge Market Research Private Ltd. URL: <https://www.databridgemarketresearch.com/reports/global-electronic-medical-records-emr-market> (date of access: 17.10.2023).
6. Preparing for the next generation of electronic health records / L. Sterrett et al. Deloitte Insights. URL: <https://www2.deloitte.com/us/en/insights/industry/health-care/ehr-systems-the-future-of-electronic-health-records.html> (date of access: 17.10.2023).
7. Architecture Design of Healthcare Software-as-aService Platform for Cloud-Based Clinical Decision Support Service / S. Oh et al. *CiteSeerX*. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b75d7b2b95873c858a9efe1ca8a4c70b2478703b> (date of access: 10.10.2023).

8. Index - FHIR v5.0.0. Health Level Seven International - Homepage | HL7 International. URL: <https://www.hl7.org/fhir/> (date of access: 23.10.2023).
9. Barnett G. O. History of the Development of Medical Information Systems at the Laboratory of Computer Science at Massachusetts General Hospital. *Sea Island Systems Home*. URL: <http://www.seaislandsystems.com/Hardhats/HistoricDocs/OctoBarnett-History.pdf> (date of access: 10.10.2023).
10. Historically Important Electronic Medical Record Systems - Clinfowiki. *Clinfowiki*. URL: https://www.clinfowiki.org/wiki/index.php/Historically_Important_Electronic_Medical_Record_Systems (date of access: 10.10.2023).
11. Western Cape Government. | B401 Solution Architecture Overview. URL: https://www.westerncape.gov.za/other/2010/2/b401_solution_architecture_v1.1.pdf (date of access: 25.10.2023).
12. Blog Index (Homepage) | CMU Software Engineering Institute. URL: https://insights.sei.cmu.edu/documents/629/2000_005_001_13706.pdf (дата звернення: 12.11.2023).
13. The C4 model for visualising software architecture. The C4 model for visualising software architecture. URL: <https://c4model.com/> (date of access: 12.11.2023).
14. HIPAA compliance checklist from HIPAA journal. HIPAA Journal. URL: <https://www.hipaajournal.com/hipaa-compliance-checklist/> (date of access: 11.10.2023).
15. HIPAA eligible services reference - amazon web services (AWS). Amazon Web Services, Inc. URL: <https://aws.amazon.com/compliance/hipaa-eligible-services-reference/> (date of access: 11.10.2023).
16. Serverless Electrocardiogram Stream Processing in Federated Clouds With Lambda Architecture / S. Ristov et al. *Computer*. 2023. Vol. 56, no. 9. P. 18–27. URL: <https://doi.org/10.1109/mc.2023.3281873> (date of access: 16.11.2023).
17. AWS Service Level Agreements. Amazon Web Services, Inc. URL: <https://aws.amazon.com/legal/service-level-agreements> (date of access: 22.11.2023).
18. Multi-region API Gateway with CloudFront - Multi-region API Gateway

with CloudFront. URL: <https://docs.aws.amazon.com/architecture-diagrams/latest/multi-region-api-gateway-with-cloudfront/multi-region-api-gateway-with-cloudfront.html> (date of access: 22.11.2023).

19. Introducing AWS Lambda response streaming | Amazon Web Services. Amazon Web Services. URL: <https://aws.amazon.com/blogs/compute/introducing-aws-lambda-response-streaming/> (date of access: 22.11.2023).

20. Zhao N. Improvement of Cloud Computing Medical Data Protection Technology Based on Symmetric Encryption Algorithm. *Journal of Testing and Evaluation*. 2022. Vol. 51, no. 1. P. 20210456. URL: <https://doi.org/10.1520/jte20210456> (date of access: 23.11.2023).

21. Zhao N. Improvement of Cloud Computing Medical Data Protection Technology Based on Symmetric Encryption Algorithm. *Journal of Testing and Evaluation*. 2022. Vol. 51, no. 1. P. 20210456. URL: <https://doi.org/10.1520/jte20210456> (date of access: 23.11.2023).

22. Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures / M. Villamizar et al. *Service Oriented Computing and Applications*. 2017. Vol. 11, no. 2. P. 233–247. URL: <https://doi.org/10.1007/s11761-017-0208-y> (date of access: 23.11.2023).

23. Compare AWS and Azure services to Google Cloud | Documentation. Google Cloud. URL: <https://cloud.google.com/docs/get-started/aws-azure-gcp-service-comparison> (date of access: 23.11.2023).

24. What is AWS Organizations? - AWS Organizations. URL: https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html (date of access: 23.11.2023).

25. Implementing Canary Deployments of AWS Lambda Functions with Alias Traffic Shifting | Amazon Web Services. Amazon Web Services. URL: <https://aws.amazon.com/blogs/compute/implementing-canary-deployments-of-aws-lambda-functions-with-alias-traffic-shifting/> (date of access: 23.11.2023).

26. Testing and debugging serverless applications - AWS Serverless Application Model. URL: <https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-test-and-debug.html> (date of access: 23.11.2023).

27. Sasikumar M., Jayarin P. J., Vinnarasi F. S. F. A Hierarchical Optimized Resource Utilization based Content Placement (HORCP) model for cloud Content Delivery Networks (CDNs). *Journal of Cloud Computing*. 2023. Vol. 12, no. 1. URL: <https://doi.org/10.1186/s13677-023-00519-2> (date of access: 25.11.2023).

28. Amazon S3 + Amazon CloudFront: A Match Made in the Cloud | Amazon Web Services. Amazon Web Services. URL: <https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon->

cloudfront-a-match-made-in-the-cloud/ (date of access: 26.11.2023).

29. Infrastructure As Code Provisioning Tool - AWS CloudFormation - AWS. Amazon Web Services, Inc. URL: <https://aws.amazon.com/cloudformation/> (date of access: 26.11.2023).

30. How does AWS CloudFormation work? - AWS CloudFormation. URL: <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-what-is-howdoesitwork.html> (date of access: 26.11.2023).

31. Template snippets - AWS CloudFormation. URL: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/CHAP_TemplateQuickRef.html (date of access: 26.11.2023).

32. What is AWS CloudFormation Designer? - AWS CloudFormation. URL: <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/working-with-templates-cfn-designer.html> (date of access: 26.11.2023).

33. AWS CloudFormation best practices - AWS CloudFormation. URL: <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/best-practices.html> (date of access: 27.11.2023).

34. Cloud as innovation driver. Deloitte Insights. URL: <https://www2.deloitte.com/us/en/insights/industry/public-sector/government-trends/2020/government-cloud-innovation.html> (date of access: 11.12.2023).

35. Gartner says cloud will be the centerpiece of new digital experiences. Gartner. URL: <https://www.gartner.com/en/newsroom/press-releases/2021-11-10-gartner-says-cloud-will-be-the-centerpiece-of-new-digital-experiences> (date of access: 11.12.2023).

36. Gartner forecasts worldwide public cloud end-user spending to reach \$679 billion in 2024. Gartner. URL: <https://www.gartner.com/en/newsroom/press-releases/11-13-2023-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-679-billion-in-2024> (date of access: 17.12.2023).

37. G-Cloud — як держави використовують хмару. GigaCloud: Хмарні Технології та Хмарний Сервіс для Бізнесу. URL: <https://gigacloud.ua/blog/navchannja/g-cloud-jak-derzhavi-vikoristovujut-hmaru> (дата звернення: 17.12.2023).

38. Cloud computing market size, share, growth drivers, opportunities & statistics. MarketsandMarkets. URL: <https://www.marketsandmarkets.com/Market-Reports/cloud-computing-market-234.html> (date of access: 17.12.2023).

39. Personal cloud market trends, size, & growth insight. Market Research Business Consulting and Strategy Planning Firm | Data Bridge Market Research Private Ltd. URL: <https://www.databridgemarketresearch.com/reports/global-personal-cloud-market> (date of access: 17.12.2023).

40. Government cloud market size, future growth, & research report. Market

Research Business Consulting and Strategy Planning Firm | Data Bridge Market Research Private Ltd. URL: <https://www.databridgemarketresearch.com/reports/global-government-cloud-market> (date of access: 17.12.2023).

41. Hybrid cloud market players, size, share, segmentation, & forecast trends by 2029. Market Research Business Consulting and Strategy Planning Firm | Data Bridge Market Research Private Ltd. URL: <https://www.databridgemarketresearch.com/reports/global-hybrid-cloud-market> (date of access: 17.12.2023).

42. Global initiative on digital health. World Health Organization CDN. URL: https://cdn.who.int/media/docs/default-source/digital-health-documents/global-initiative-on-digital-health_executive-summary-31072023.pdf (date of access: 17.12.2023).

43. WHO launches a new global initiative on digital health supported by the G20 presidency. World Health Organization (WHO). URL: <https://www.who.int/news/item/19-08-2023-who-launches-a-new-global-initiative-on-digital-health-at-the-g20-summit-in-india> (date of access: 17.12.2023).

44. Geolocated health facilities data initiative. World Health Organization (WHO). URL: <https://www.who.int/data/GIS/GHFD> (date of access: 17.12.2023).

45. Digital pharmacy market revenue trends and growth drivers | MarketsandMarkets. MarketsandMarkets. URL: <https://www.marketsandmarkets.com/Market-Reports/digital-pharmacy-market-134868253.html> (date of access: 17.12.2023).

46. IoT healthcare market size, statistics, growth analysis & trends [2030]. MarketsandMarkets. URL: <https://www.marketsandmarkets.com/Market-Reports/iot-healthcare-market-160082804.html> (date of access: 17.12.2023).

47. Digital twins in healthcare market size, share, trends and revenue forecast, 2028 | marketsandmarkets. MarketsandMarkets. URL: <https://www.marketsandmarkets.com/Market-Reports/digital-twins-in-healthcare-market-74014375.html> (date of access: 17.12.2023).

48. Telehealth & telemedicine market size, share, trends and revenue forecast [latest]. MarketsandMarkets. URL: <https://www.marketsandmarkets.com/Market-Reports/telehealth-market-201868927.html> (date of access: 17.12.2023).

49. IOS - health. Apple (Україна). URL: <https://www.apple.com/ua/ios/health/> (date of access: 17.12.2023).

50. AWS for Healthcare & Life Sciences | AWS. Amazon Web Services, Inc. URL: <https://aws.amazon.com/health/> (date of access: 17.12.2023).

51. What Is Google Health? - Google Health. What Is Google Health? - Google Health. URL: <https://health.google/> (date of access: 17.12.2023).

52. Digital Health Tech – eHealth Information Management | Microsoft Industry. Digital Health Tech – eHealth Information Management | Microsoft Industry. URL: <https://www.microsoft.com/mea/industry/health/> (date of access: 17.12.2023).

53. Health Data Cloud. Health Data Cloud. URL: <https://www.healthdatacloud.eu/> (date of access: 17.12.2023).

54. Dell Data Storage Solutions for Healthcare and Life Sciences | Dell USA. Dell Technologies. URL: <https://www.dell.com/en-us/dt/solutions/healthcare-and-life-sciences.htm> (date of access: 17.12.2023).

55. Про схвалення Концепції розвитку цифрової економіки та суспільства України на 2018-2020 роки та затвердження плану заходів щодо її реалізації : Розпорядж. Каб. Міністрів України від 17.01.2018 р. № 67-р : станом на 17 верес. 2020 р. URL: <https://zakon.rada.gov.ua/laws/show/67-2018-p#Text> (дата звернення: 06.12.2023).

56. Аналітичні панелі (дашборди). Національна служба здоров'я України. URL: <https://edata.e-health.gov.ua/e-data/dashboard> (дата звернення: 06.12.2023).

57. Е-лікарняний: важливі роз'яснення - Пенсійний фонд України. Пенсійний фонд України. URL: <https://www.pfu.gov.ua/2148877-e-likarnyanuj-vazhlyvi-roz-yasnennya/> (дата звернення: 07.12.2023).

58. Перевірка медичних висновків. Державні послуги онлайн | Дія. URL: <https://diia.gov.ua/services/perevirka-medichnih-visnovkiv> (дата звернення: 06.12.2023).

59. Про затвердження нормативних документів щодо застосування телемедицини у сфері охорони здоров'я : Наказ МОЗ України від 19.10.2015 р. № 681 : станом на 21 жовт. 2022 р. URL: <https://zakon.rada.gov.ua/laws/show/z1400-15#Text> (дата звернення: 06.12.2023).

60. Відтепер більше українців зможе скористатися телемедичними послугами. Електронна система охорони здоров'я в Україні. URL: <https://ehealth.gov.ua/2023/11/30/vidteper-bilshe-ukrayintsiv-zmozhe-skorystatsya-telemedychnymu-poslugamy/> (дата звернення: 06.12.2023).

61. Охорона здоров'я | Fact Sheet | Ukraine | U.S. Agency for International Development. U.S. Agency for International Development. URL: <https://www.usaid.gov/uk/ukraine/fact-sheets/aug-03-2022-health> (дата звернення: 07.12.2023).

62. Академія. Національна служба здоров'я України. URL: <https://academy.nszu.gov.ua> (дата звернення: 06.12.2023).

63. Деякі питання провадження господарської діяльності з медичної практики : Постанова Каб. Міністрів України від 16.02.2022 р. № 126 : станом на 31 груд. 2022 р. URL: <https://zakon.rada.gov.ua/laws/show/126-2022-p#Text> (дата звернення: 07.12.2023).

64. Про затвердження Ліцензійних умов провадження господарської діяльності з медичної практики. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/285-2016-п#n8> (дата звернення: 07.12.2023).

65. Електронна система охорони здоров'я в Україні. URL: <https://ehealth.gov.ua/> (дата звернення: 06.12.2023).

66. eHealth - що це та як підключитися? - Блог Health24. Блог Health24. URL: <https://blog.h24.ua/uk/ehealth-shho-tse-ta-yak-pidklyuchytysya/> (дата звернення: 06.12.2023).

67. Health24. URL: <https://h24.ua/> (дата звернення: 06.12.2023).

68. Про хмарні послуги : Закон України від 17.02.2022 р. № 2075-IX. URL: <https://zakon.rada.gov.ua/laws/show/2075-20#Text> (дата звернення: 06.12.2023).

69. Хмара для Сфери Охорони Здоров'я: Технологічні Рішення | GigaCloud. GigaCloud: Хмарні Технології та Хмарний Сервіс для Бізнесу. URL: <https://gigacloud.ua/services/hmarni-tehnologii-dlja-mediciny> (дата звернення: 07.12.2023).

70. Деякі питання забезпечення функціонування інформаційно-комунікаційних систем, електронних комунікаційних систем, публічних електронних реєстрів в умовах воєнного стану : Постанова Каб. Міністрів України від 12.03.2022 р. № 263 : станом на 23 трав. 2023 р. URL: <https://zakon.rada.gov.ua/laws/show/263-2022-п#Text> (дата звернення: 06.12.2023).

71. Про внесення змін до деяких законів України щодо забезпечення функціонування інформаційно-комунікаційних систем, електронних комунікаційних систем, публічних електронних реєстрів : Закон України від 15.03.2022 р. № 2130-IX : станом на 29 лип. 2023 р. URL: <https://zakon.rada.gov.ua/laws/show/2130-20#Text> (дата звернення: 06.12.2023).

72. Стратегія розвитку системи охорони здоров'я до 2030 року. Міністерство охорони здоров'я України. URL: <https://moz.gov.ua/uploads/ckeditor/Стратегія/UKR%20Health%20Strategy%20Feb%2024.2022.pdf> (дата звернення: 07.12.2023).

73. Emergency Medicine History and Expansion into the Future: A Narrative Review / M. Huecker et al. *Western Journal of Emergency Medicine*. 2022. Vol. 23, no. 3. P. 418–423. URL: <https://doi.org/10.5811/westjem.2022.2.55108> (date of access: 02.12.2023).

74. Milgrim F. Artificial Intelligence's Leap Into Emergency Medicine: The Promise and Pitfalls of ChatGPT. *Annals of Emergency Medicine*. 2023. Vol. 82, no. 4. P. A13–A16. URL: <https://doi.org/10.1016/j.annemergmed.2023.08.378> (date of access: 17.12.2023).

75. Український інститут майбутнього. Економічна стратегія України 2030 >> Український інститут майбутнього. URL: <https://strategy.uifuture.org/kraina-z>

rozvnutoyu-cifrovoyu-ekonomikoyu.html (дата звернення: 17.12.2023).

76. Home | Transform Health. Transform Health. URL: <https://transformhealthcoalition.org/> (date of access: 17.12.2023).

77. More and better funding for the digital transformation of health. Home | Transform Health. URL: <https://transformhealthcoalition.org/wp-content/uploads/2023/04/Private-Sector-Policy-Brief-v2.pdf> (date of access: 17.12.2023).

78. Individuals' Perceptions of the Privacy and Security of Medical Records and Health Information Exchange | HealthIT.gov. ONC | Office of the National Coordinator for Health Information Technology. URL: <https://www.healthit.gov/data/quickstats/individuals-perceptions-privacy-and-security-medical-records-and-health-information> (date of access: 17.12.2023).

79. Milgrim F. Artificial Intelligence's Leap Into Emergency Medicine: The Promise and Pitfalls of ChatGPT. *Annals of Emergency Medicine*. 2023. Vol. 82, no. 4. P. A13–A16. URL: <https://doi.org/10.1016/j.annemergmed.2023.08.378> (date of access: 17.12.2023).

80. Ali O., Osmanaj V. The role of government regulations in the adoption of cloud computing: A case study of local government. *Computer Law & Security Review*. 2020. Vol. 36. P. 105396. URL: <https://doi.org/10.1016/j.clsr.2020.105396> (date of access: 17.12.2023).

81. Pang M.-S., Tanriverdi H. Strategic roles of IT modernization and cloud migration in reducing cybersecurity risks of organizations: The case of U.S. federal government. *The Journal of Strategic Information Systems*. 2022. Vol. 31, no. 1. P. 101707. URL: <https://doi.org/10.1016/j.jsis.2022.101707> (date of access: 17.12.2023).

82. The effects of e-Government cloud assimilation on public value creation: An empirical study of China / Y. Liang et al. *Government Information Quarterly*. 2019. Vol. 36, no. 4. P. 101397. URL: <https://doi.org/10.1016/j.giq.2019.101397> (date of access: 17.12.2023).