НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

«На правах рукопису»

УДК 004.75, 004.738

«До захисту допущено»
Завідувач кафедри
_____ Сергій КРАВЧУК
«___»_____2021 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Інженерія та програмування інфокомунікацій»
зі спеціальності 172 «Телекомунікації та радіотехніка»
на тему: «Аналіз методів побудови аналітичних платформ для Інтернету речей»

Виконав:
студент ІІ курсу, групи ТЗ-01мп
Коробчук Олександр Миколайович _____

Керівник:
доцент кафедри ТК, к.т.н., с.н.с.
Міночкін Дмитро Анатолійович _____

Рецензент:
Професор каф. ІКТС ІТС, д.т.н., с.н.с.
Скулиш Марія Анатоліївна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2021 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій КРАВЧУК
«___»_____2021 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Коробчуку Олександру Миколайовичу

1. Тема дисертації «Аналіз методів побудови аналітичних платформ для Інтернету речей», науковий керівник дисертації Міночкін Дмитро Анатолійович, кандидат технічних наук, старший науковий співробітник, затверджені наказом по університету від «04» листопада 2021 р. № 3672-с.

2. Термін подання студентом дисертації 10.12.2021 р.

3. Об’єкт дослідження: аналітична платформа для Інтернету речей

4. Предмет дослідження: методи побудови аналітичних платформ для Інтернету речей

5. Перелік завдань, які потрібно розробити:

- провести аналітичний огляд архітектури Інтернету речей, її основних рівнів, протоколів;
- проаналізувати методи побудови аналітичних платформ для Інтернету речей, визначити ключові функції та дослідити можливі варіанти використання;

- моделювання роботи аналітичної платформи для Інтернету речей.

6. Орієнтовний перелік ілюстративного матеріалу: Основні етапи архітектури Інтернету речей; Перша частина IoT архітектури (датчики та приводи); Друга частина IoT архітектури (шлюзи та збір даних); Третя частина IoT архітектури (концепція граничних обчислень); Четверта частина IoT архітектури (дата центри, хмара); Групи протоколів в клієнт-серверній архітектурі; Схематичне зображення принципу дії протоколу MQTT; Схематичне зображення принципу дії протоколу LwM2M; Структурна схема принципу роботи аналітичної IoT-платформи; Модель використання AWS IoT Analytics; Архітектура Microsoft Azure IoT; Архітектура Google IoT Platform; Створення таблиці даних про погоду; Процес створення Cloud Functions; Програмний код функції в Cloud Functions; Встановлення пакетів конфігурації; Статус функції в Cloud Functions; Параметри деталей роботи; Статус роботи Dataflow; Заповнена таблиця BigQuery з опрацьованими даними; Графічне зображення отриманих результатів.

7. Дата видачі завдання “01” вересня 2019 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Ґрунтовне ознайомлення з предметною галуззю	07.09.2020	виконано
2	Визначення структури магістерської дисертації; пошук та опрацювання джерел інформації	09.10.2020	виконано
3	Робота над першим розділом магістерської дисертації	25.11.2020	виконано
4	Проведення наукового дослідження; робота над другим розділом магістерської дисертації	01.03.2021	виконано
5	Робота над третім розділом магістерської дисертації; виконання практичної частини дослідження	01.09.2021	виконано
6	Розробка стартап проекту	29.10.2021	виконано

7	Завершення роботи над основною частиною магістерської дисертації; підготовка ілюстративного матеріалу	15.11.2021	виконано
8	Оформлення текстової і графічної частини магістерської дисертації	19.11.2021	виконано
9	Попередній розгляд магістерської дисертації	08.12.2021	виконано

Студент Олександр КОРОБЧУК

Науковий керівник дисертації Дмитро МІНОЧКІН

РЕФЕРАТ

Робота містить: 81 сторінку, 21 рисунок та 2 таблиці. Було використано 15 джерел інформації.

Актуальність теми. Обсяги даних, які доводиться аналізувати сучасному бізнесу, зростають нелінійно і все більше компаній автоматизують роботу з Big Data. Рішення з урахуванням інструментів бізнес-аналітики дедалі більше захоплюють міжнародний ринок. Використовуються різні підходи, будуються теорії, обробляються великі масиви даних. Агрегації, консолідовані набори інформації дозволяють приймати стратегічні рішення, здійснювати планування діяльності та бізнес-процесів у різних галузях та сферах життєдіяльності суспільства. За допомогою сучасних методів аналізу великих даних, безпосередньо використовуючи аналітичні платформи, компанії оптимізують виробничі процеси, прогнозують відмови обладнання та відтік клієнтів, здійснюють кредитний скоринг та розумну маршрутизацію дзвінків, виявляють шахрайські дії тощо.

Об’єктом дослідження є аналітична платформа для Інтернету речей.

Предметом дослідження є методи побудови аналітичних платформ для Інтернету речей.

Мета роботи: полягає у проведені аналізу методів побудови аналітичних платформ для Інтернету речей. Розуміння основних принципів побудови та можливостей застосування.

Наукова новизна:

1. Визначено ключові функції, які повинні виконувати аналітичні платформи для Інтернету речей, щоб результати отримані під час роботи платформи були корисні та могли бути використанні в подальшому.
2. З’ясовані аспекти, які потрібно враховувати при побудові аналітичних платформ, щоб робота платформи була коректною та відповідала поставленим задачам.

Практична цінність полягає у тому, що було отримано модель аналітичної платформи зі створення багатого контексту. Маючи це на місці, можна вжити найкращих заходів для вирішення ситуації – чи то зниження операційних ризиків/витрат, отримання нових доходів або підвищення ефективності роботи.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів та висновків.

У *вступі* подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень.

У *першому розділі* наведено базову структуру Інтернету речей та розглянуто стек протоколів, що використовуються.

У *другому розділі* було проаналізовано методи побудови аналітичних платформ для Інтернету речей та їх ключові функції.

У *третьому розділі* проведено моделювання роботи аналітичної платформи для Інтернету речей.

У *четвертому розділі* розроблено стартап проект по темі магістерської дисертації.

У *висновках* представлені результати проведеної роботи.

Ключові слова: Інтернет речей, аналітична платформа, MQTT, HTTP.

ABSTRACT

The work contains: 81 pages, 21 figures and 2 tables. 15 sources of information were used.

Actuality of theme. The amount of data that today's business has to analyze is growing non-linearly and more and more companies are automating Big Data. Business analytics tools are increasingly gaining ground in the international marketplace. Different approaches are used, theories are built, large data sets are processed. Aggregations, consolidated sets of information allow you to make strategic decisions, plan activities and business processes in various sectors and areas of society. Using modern methods of big data analysis, directly using analytical platforms, companies optimize production processes, predict equipment failures and customer outflows, perform credit scoring and smart call routing, detect fraud and more.

The object of research is an analytical platform for the Internet of Things.

The subject of research is methods of building analytical platforms for the Internet of Things.

Purpose: is to analyze the methods of building analytical platforms for the Internet of Things. Understanding the basic principles of construction and application possibilities.

Scientific novelty:

1. The key functions that analytical platforms for the Internet of Things must perform in order for the results obtained during the operation of the platform to be useful and can be used in the future.

2. Clarified aspects that need to be considered when building analytical platforms, so that the work of the platform was correct and consistent with the objectives.

The practical value is that a model of an analytical platform for creating a rich context was obtained. With this in place, the best measures can be taken to address the

situation - whether to reduce operational risks / costs, generate new income or increase efficiency.

Structure and scope of work. The master's dissertation consists of an introduction, four chapters and conclusions.

The introduction presents a general description of the work, assesses the current state of the problem, substantiates the relevance of research.

The first section presents the basic structure of the Internet of Things and discusses the stack of protocols used.

The second section analyzes the methods of building analytical platforms for the Internet of Things and their key functions.

The third section simulates the work of the analytical platform for the Internet of Things.

In the fourth section, a startup project on the topic of a master's dissertation is developed.

The conclusions present the results of the work.

Keywords: Internet of Things, analytical platform, MQTT, HTTP.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

IoT	Internet of Things
LAN	Local Area Network
PAN	Personal Area Network
CC	Cloud Computing
D2D	Device to Device
D2S	Device to Service
S2S	Service to Service
M2M	Machine to Machine
DCPS	Data Centric Publish-Subscribe
DRLR	Data-Local Reconstruction Layer
DDS	Data-Distribution Service
LwM2M	Lightweight Machine to Machine
XMPP	Extensible Messaging and Presence Protocol
CoAP	Constrained Application Protocol
MQTT	Message Queuing Telemetry Transport
HTTP	HyperText Transfer Protokol
ML	Machine Learning

ВСТУП

Інтернет речей (IoT), великі дані та аналітика – це нові сфери зростання та перспективи. Хоча ринкова вартість і потенціал високі, а варіанти використання здаються очевидними, компанії прагнуть покращити реальні бізнес-результати та вартість, створену в проектах IoT. Потрібні нові види аналітичних платформ та інструментів, які допоможуть їм швидко досягти поставлених цілей.

IoT несе виклики іншого рівня. В Інтернеті речей ми матимемо справу з величезною кількістю даних, які мають високий ступінь дисперсії щодо швидкостей, каналів і циклів даних. Коли ми бачимо під'єднані мільйони й мільярди пристроїв в Інтернеті речей, щомиті ми бачимо величезну кількість нових даних, які генеруються, які можуть принести більше уявлень. Операційні менеджери хотіли б використовувати ці дані для виявлення аномалій, раннього прогнозування проблем, пом'якшення будь-яких збоїв у роботі сервісу та надання нових послуг клієнтам.

На додаток до вибуху даних, бізнес-середовище та умови змінюються швидше. У цьому новому середовищі необхідні прийняття рішень у режимі реального часу та швидке реагування на конкурентні та операційні виклики.

Організаціям потрібно вжити заходів і швидко реагувати на навколишнє середовище та вирішувати проблеми IoT, щоб знайти розуміння та цінність.

Щоб подолати ці тенденції, менеджери повинні вирішити потребу в швидшій аналітиці в цілому всіх типів даних — швидких, повільних і проміжних — через природу та різноманітність типів даних в Інтернеті речей. Щоб подолати проблеми IoT, знадобиться:

• Швидші інструменти й послуги аналітики необхідні для вирішення критичних часу проблем IoT. Нові форми уніфікованих аналітичних платформ

також необхідні для роботи з різноманітністю та обсягом великих даних у масовому масштабі в рамках IoT.

• Аналітики та команди розробників додатків мають бути наділені інструментами самообслуговування та автоматизацією розробки на основі моделі, щоб вони могли швидше створювати та створювати аналітичні додатки.

• Уніфікація історичної, потокової та прогнозної аналітики в реальному часі, щоб забезпечити ширший контекст «загальної картини» для менеджерів з бізнес-операцій, щоб швидко діяти. Важливо використовувати досягнення в усіх типах аналітики, щоб прискорити ланцюг створення цінності аналітики та мати найбільший вплив на вартість бізнесу.

Ця швидша уніфікована аналітика надає ширший контекст, який, у свою чергу, надасть своєчасну інформацію, яка дозволить діяти більш розумно та покращувати результати бізнесу. Найбільшу цінність несе створення багатого контексту з використанням усіх типів аналітики, щоб зрозуміти всю складність будь-якого сценарію IoT. Маючи це на місці, можна вжити наступних найкращих заходів для вирішення ситуації – чи то зниження операційних ризиків/витрат, отримання нових доходів або підвищення ефективності роботи.

РОЗДІЛ 1
АРХІТЕКТУРА ІНТЕРНЕТУ РЕЧЕЙ

1.1. Загальні відомості Інтернету речей

Кевін Аштон казав, що сьогодні комп'ютери, а значить, і Інтернет, майже повністю залежать від людини для отримання інформації. Майже весь об'єм даних, наявних в Інтернеті, був вперше зафіксований та створений людьми – шляхом набору, натискання кнопки запису, фотографування або сканування штрих-коду. Звичайні діаграми Інтернету включають сервери та маршрутизатори, тощо, але вони залишають без уваги найчисленніші та найважливіші маршрутизатори з усіх: люди. Проблема полягає в тому, що люди мають обмежений час, увагу та акуратність – все це означає, що вони не дуже хороші в збиранні даних про речі в реальному світі. Ми фізичні, як і наше оточення. Наша економіка, суспільство та виживання не базуються на ідеях чи інформації – вони ґрунтуються на речах. Ідеї та інформація важливі, але речі мають значення набагато більше. Однак сьогоднішня інформаційна технологія настільки залежить від даних, які походять від людей, що наші комп'ютери знають більше про ідеї, ніж про речі [1].

Інтернет речей – це сукупність взаємодіючих технічних систем і комплексів, що складаються з мікропроцесорів, сенсорів, пристроїв, систем передачі даних, локальних і/або розподілених обчислювальних ресурсів і програмних засобів, зокрема програм штучного інтелекту, призначених для здійснення суспільних відносин, в тому числі, пов'язаних з наданням послуг і проведенням робіт за безпосередньої участі або без участі суб'єктів (юридичних або фізичних осіб) на основі використання великих даних і мережі Інтернет [2]. Інтернет речей ґрунтується на трьох базових принципах. По-перше, повсюдно поширену комунікаційну інфраструктуру, по-друге, глобальну ідентифікацію кожного об'єкта і, по-третє, можливість кожного об'єкта відправляти і отримувати дані

за допомогою персональної мережі або мережі Інтернет, до якої він підключений [3].

Джон Доу вважав, що Інтернет речей має силу змінювати та покращувати наше повсякденне життя разом із способами, в яких ми функціонуємо як суспільство, він також може трансформувати спосіб ведення бізнесу і, зрештою, те як ми сприймаємо практичо кожен аспект нашого світу[15].

IoT допомагає автоматизувати більшість операцій за рахунок:

- автоматичного збору інформації про об’єкти (механізмах, обладнанні, пристроях, приміщеннях, транспортних засобах) для відслідковування статусу або поведінки;
- використання цієї інформації для контролю і управління, що допомагає оптимізувати процеси і використання ресурсів, а також покращити процес прийняття рішень.

Одним з двох найбільших викликів, що стоять перд IoT є фрагментація (інший – безпека). Це основа Інтернету речей через різноманітний характер речей, які він має на мені поєднати. Для роботи будь-якої системи IoT потрібне використання ресурсів, апаратного забезпечення, програмного забезпечення та систем, їх потрібно поєднати в єдине ціле, якими б вони не були різноманітними, щоб сформувати комплексне, надійне та економічно вигідне рішення. Простіше кажучи, для кожного розгортання IoT потрібна міцна архітектура IoT, щоб мати можливість виконувати призначене для цього завдання; результативна ефективність та застосованість системи багато в чому залежить від якості розробленої інфраструктури[15].

IoT розвинувся в зв’язку з розвитком M2M, тобто машин, що з’єднувались один з одним через мережу без взаємодії людини. M2M відноситься до підключення пристрою до хмари, управління ним та збору даних. Піднімаючи

M2M на наступний рівень, IoT – це сенсорна мережа з мільярдів розумних пристроїв, яка з'єднує людей, системи та інші програми для збору та обміну даними. В якості основи M2M пропонує підключення, що дозволяє IoT [5].

Хоча кожна система IoT різна, основа для кожної архітектури Інтернету речей, а також загальний потік його даних приблизно однаковий. Перш за все, він складається з Речей, що є об'єктами, підключеними до Інтернету, які за допомогою вбудованих датчиків та виконавчих механізмів здатні аналізувати навколишнє середовище та збирати інформацію, яка потім передається на шлюзи IoT. Наступний етап складається із систем збору даних Інтернету речей та шлюзів, які збирають велику масу необроблених даних, перетворюють їх у цифрові потоки, фільтрують та попередньо обробляють, щоб вони були готові до аналізу. Третій етап представлений граничними пристроями, відповідальними за подальшу обробку та розширений аналіз даних. На цьому етапі також можуть бути технології візуалізації та машинного навчання. Після цього дані передаються в центри обробки даних, які можуть бути або хмарними, або встановленими локально [4].

1.2. Архітектура Інтернету речей

IoT архітектура складається з 4-х частин [4]:

- датчики та приводи;
- шлюзи та збір даних;
- Edge IT (концепція граничних обчислень);
- дата центри, хмара.

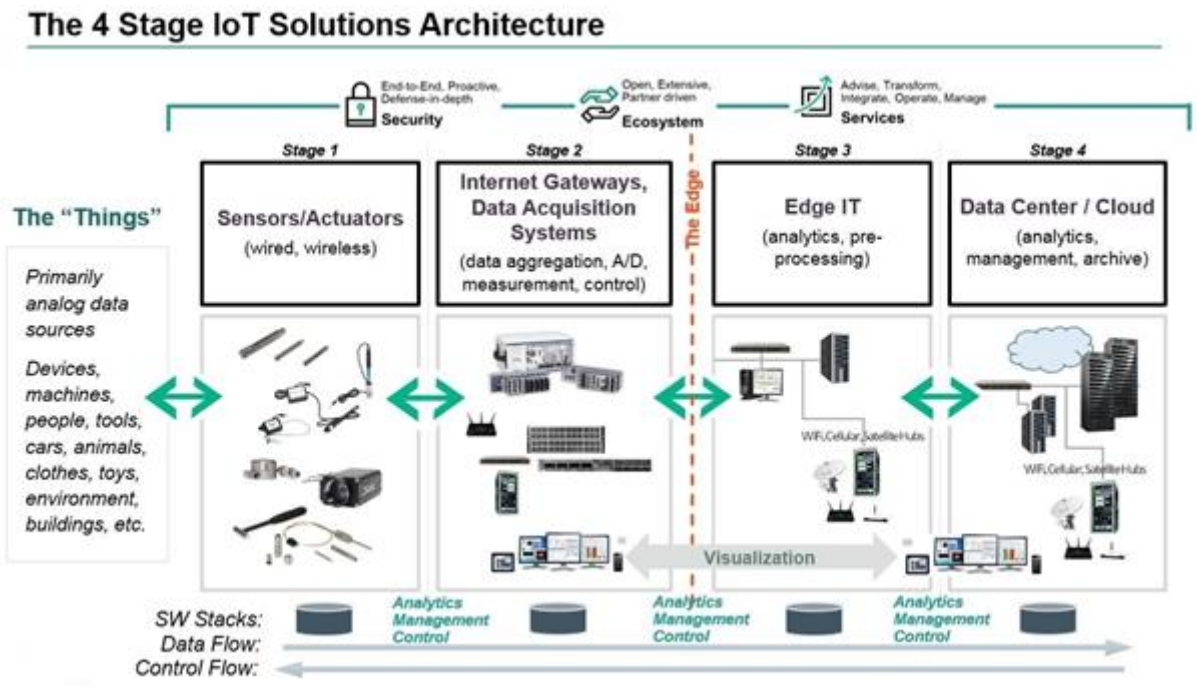

Рис. 1.1. Основні етапи архітектури Інтернету речей

1.2.1 Датчики та приводи

Як основа для кожної системи IoT, підключені пристрої відповідають за забезпечення сутності Інтернету речей, що є даними. Щоб зібрати фізичні параметри у зовнішньому світі чи всередині самого об'єкта, їм потрібні датчики. Вони можуть бути або вбудовані в самі пристрої, або реалізовані як окремі об'єкти для вимірювання та збору даних телеметрії. Так само, як наше зір, слух, запах, дотик та смак дозволяють нам, людям, осмислити світ, так датчики дозволяють машинам осягнути світ. Для прикладу, подумайте про датчики сільського господарства, завдання яких – вимірювати такі параметри, як температура повітря та ґрунту, вологість, рівень Рн ґрунту та багато іншого. Більшість датчиків потребує підключення через шлюзи датчиків. Підключення датчиків може здійснюватися через локальну мережу (LAN) або персональну мережу (PAN) [15].

Ще один незамінний елемент цього шару – це приводи. Перебуваючи у тісній співпраці з датчиками, вони можуть перетворювати дані, що генеруються розумними об’єктами, у фізичну дію. Давайте уявимо розумну систему поливу з усіма необхідними датчиками. На основі вхідних даних з датчиків система аналізує ситуацію в режимі реального часу і командує приводами відкривати вибрані клапани води, розташовані в місцях, де вологість ґрунту нижче встановленого значення. Клапани тримаються відкритими, поки датчики не повідомлять про відновлення значень за замовчуванням. Очевидно, все це відбувається без єдиного втручання людини[15].

Також важливим є те, що підключені об’єкти повинні бути не лише здатними спілкуватися в двосторонній формі з відповідними шлюзами або системами збору даних, але й бути в змозі розпізнавати та говорити один з одним, щоб збирати та обмінюватися інформацією та співпрацювати в режимі реального часу, щоб використовувати значення всього розгортання. Зокрема, для пристроїв з обмеженими ресурсами та акумуляторами, досягти цього непросте завдання, оскільки таке спілкування вимагає великої кількості обчислювальної потужності та споживає дорогоцінну енергію та пропускну здатність. Отже, надійна архітектура може дозволяти ефективно керувати пристроями лише тоді, коли використовуються підходящі за призначенням безпечні та легкі протоколи зв’язку, такі як Lightweight M2M, який став провідним стандартним протоколом управління легкими пристроями малої потужності, типовими для багатьох випадків використання IoT[15].

Рис. 1.2. Перша частина IoT архітектури (датчики та приводи)

1.2.2 Шлюзи та збір даних

Хоча цей шар все ще функціонує в безпосередній близькості від датчиків і виконавчих механізмів на даних пристроях, важливо описати його як окремий етап архітектури IoT, оскільки він має вирішальне значення для процесів збору даних, фільтрації та передачі до граничної інфраструктури та хмарних платформ. Зважаючи на величезний обсяг введення та виведення, який можуть генерувати розгортання на мільйонних пристроях, можливості для агрегації, вибору та транспортування даних повинні бути в центрі уваги. Як посередники між підключеними речами, хмарою та аналітикою, шлюзи та системи збору даних забезпечують необхідну точку з’єднання, яка зв’язує решта шарів разом[15].

Будучи на межі світу експлуатаційних та інформаційних технологій, шлюзи полегшують зв’язок між датчиками та іншою частиною системи, перетворюючи дані сенсорів у формати, які легко передаються та використовуються для інших компонентів системи. Більше того, вони здатні контролювати, фільтрувати та відбирати дані, щоб мінімізувати обсяг інформації, яку потрібно переслати в хмару, що позитивно впливає на

витрати на передачу даних в мережі та час відповіді. Таким чином, шлюзи забезпечують місце для локальної попередньої обробки даних датчиків, які видавлюються в корисні пакети, готові для подальшої обробки[15].

Оскільки велика кількість даних отримується за допомогою датчиків, то для передачі даних приводи потребують високошвидкісних шлюзів та мереж. Ця мережа може мати тип локальної мережі (наприклад, Wi-Fi, Ethernet тощо), широкосмугової мережі (WAN, таких як GSM, 5G тощо).

Ще один аспект, який підтримують шлюзи – це безпека. Оскільки шлюзи несуть відповідальність за управління потоком інформації в обох напрямках, за допомогою належного інструменту шифрування та захисту вони можуть запобігти витоку хмарних даних IoT, а також зменшити ризик шкідливих зовнішніх атак на пристрої IoT[15].

Рис. 1.3. Друга частина IoT архітектури (шлюзи та збір даних)

1.2.3 Edge IT (концепція граничних обчислень)

Edge IT – це апаратні та програмні шлюзи, які аналізують та попередньо обробляють дані перед передачею їх у хмару. Якщо дані, отримані з датчиків і шлюзів, не змінені від попереднього значення зчитування, вони не передаються через хмару, це зберігає використовувані дані[15].

Будучи невід'ємною складовою кожної архітектури IoT, граничні пристрої можуть принести значні переваги, особливо великомасштабним проектам IoT. В умовах обмеженої доступності та швидкості передачі даних хмарних платформ IoT граничні системи можуть забезпечити швидший час реагування та більшу гнучкість в обробці та аналізі даних IoT. Оскільки швидкість аналізу даних є ключовою в деяких індустріальних додатках для Інтернету речей, останнім часом крайові обчислення спостерігають різке зростання популярності серед екосистем Індустріального Інтернету Речей[15].

Оскільки гранична інфраструктура може бути розташована ближче до джерела даних у фізичному відношенні, то IT-служба може простіше та швидше працювати з матеріалам IoT в режимі реального часу та забезпечувати результат у вигляді миттєвої інформації. У цьому випадку сюди пересилаються лише більші фрагменти даних, які дійсно потребують потужності Хмари для обробки. За рахунок мінімізації впливу на мережу безпека може бути значно підвищена, тоді як зменшення споживання електроенергії та пропускної здатності сприяє більш ефективному використанню бізнес-ресурсів[15].

Рис. 1.4. Третя частина IoT архітектури (концепція граничних обчислень)

1.2.4 Дата центри, хмара

Якщо датчики є нейронами, а шлюз є основою IoT, то хмара – це мозок в тілі Інтернету речей і належать до служб управління. На відміну від граничних рішень, центр обробки даних або хмарна система призначений для зберігання, обробки та аналізу величезних обсягів даних для більш глибокого розуміння з використанням потужних механізмів аналізу даних та механізмів машинного навчання, які граничні системи ніколи не зможуть підтримати[15].

За останні кілька років збільшилось впровадження хмарних обчислень (особливо в промисловій архітектурі IoT). Вони сприяють підвищенню темпів виробництва, скороченню незапланованого простою та споживання енергії та багатьох інших корисних речей для бізнесу[15].

Якщо наявні відповідні рішення для користувацьких додатків, хмара може надавати бізнес-аналіз та варіанти презентації, які допомагають людям взаємодіяти із системою, контролювати та відстежувати її і приймати обґрунтовані рішення на основі звітів, інформаційних панелей та даних, що переглядаються в режимі реального часу[15].

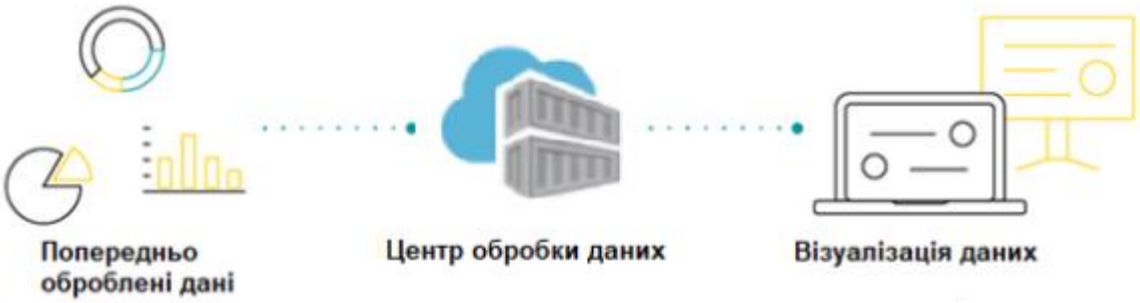

Рис. 1.5. Четверта частина IoT архітектури (дата центри, хмара)

1.3 Взаємодія IoT з перспективними інфокомунікаційними технологіями

Великі дані (Big Data). Інтернет речей радикальним чином збільшує обсяг зібраних даних, що є наслідком величезного кількості джерел інформації[15].

Гігантські сенсорні мережі вже зараз виробляють величезні потоки даних, які треба вміти не тільки зберігати, але й обробляти, робити по ним висновки, приймати рішення – і все це з урахуванням неточності як оригінальних даних, так і процедур обробки. В кінці 2000-х років для обробки великого обсягу даних сформувався підхід, який називається «великі дані» - це серія інструментів і методів обробки структурованих і неструктурованих даних величезних обсягів значного різноманіття для отримання необхідних результатів обробки[15].

Основна відмінність великих даних від «звичайних» полягає в тому, що ці дані неможливо обробити традиційними системами управління базами даних (СУБД) і рішеннями класу Business Inteligence через їх великий обсяг і різноманітний склад. Інша важлива їх властивість – феноменальне прискорення накопичення даних і постійна зміна. Такі популярні завдання, як зведення даних, отриманих з різних джерел, вимагають особливих методів аналізу в разі неточних даних, особливо даних величезних розмірів. У зв’язку з цим і був розроблений набір інструментів, які отримали назву «великі дані», що дозволяють працювати з даними незалежно від їх типу і обсягу [3].

Методи і техніки аналізу великих даних:

- методи класу отримання даних Data Mining – навчання асоціативним правилами, класифікація, кластерний аналіз, регресивний аналіз;

- краудсорсинг – категоризація та збагачення даних силами широкого, невизначеного кола осіб, залучених на підставі публічної оферти, без вступу в трудові відносини;

- змішання і інтеграція даних – набір технік, що дозволяють інтегрувати різнорідні дані з різноманітних джерел для можливості глибинного аналізу;

- машинне навчання – використання моделей, побудованих на базі статичного аналізу або машинного навчання для отримання комплексних прогнозів на основі базових моделей; штучне нейронні мережі, мережевий аналіз, оптимізація, в тому числі генетичний алгоритм;

- розпізнавання образів;

- прогнозна аналітика;

- імітаційне моделювання;

- просторовий аналіз – клас методів, які використовують топологічну, геометричну і географічну інформацію в даних;

- статистичний аналіз (наприклад, A / B-тестування, аналіз часових рядів);

- візуалізація аналітичних даних – подання інформації у вигляді малюнків діаграм, з використанням інтерактивних можливостей та анімації, як для отримання результатів, так і при використання в якості вихідних даних для подальшого аналізу [3].

Хмарні обчислення (Cloud Computing).

Так як інтернет речей породжує «великі дані», тому виникає закономірне питання: де їх зберігати і чим обробляти? Відповіддю на це питання є перспективна інфокомунікаційна технологія – хмарні обчислення (CC, Cloud Computing). Хмарні обчислення являють собою оренду послуг і ресурсів для зберігання і опрацювання даних в глобальній мережі замість власної інфраструктури. У систем CC повинні бути п’ять основних характеристик: самообслуговування на вимогу, широкосмуговий мережевий доступ, пул ресурсів, можливість швидкого переналаштування або розширення і вимірне обслуговування[15].

Існують чотири моделі розгортання хмарної інфраструктури (так званих «хмар») [3]:

1. Приватна хмара (англ. Private cloud) – призначена для використання однією організацією, що містить кілька споживачів (Наприклад, підрозділ однієї організації), можливо також клієнтами і підрядниками даної організації, так і третьої сторони, і може існувати як всередині, так і поза юрисдикцією власника.

2. Публічна хмара (англ. Public cloud) – призначена для вільного використання широкою публікою. Публічною хмарою можуть користуватись комерційні, наукові і урядові організації. Публічне хмара фізично існує в юрисдикції власника – постачальника послуг.

3. Гібридна хмара – (англ. Hybrid cloud) – комбінація декількох різних типів хмар (приватних, публічних або суспільних), що є унікальними об’єктами, але пов’язані між собою стандартизованими або приватними технологіями передачі даних і програм.

4. Громадська хмара (англ. Community cloud) – призначена для користування конкретною спільнотою споживачів з організацій, що мають спільні цілі (наприклад, місії, вимог безпеки, політики). Громадська хмара може бути в

корпоративній (спільній) власності, управлінні в експлуатації однієї або більше організацій спільноти або третьої сторони, і воно може фізично існувати як всередині, так і поза юрисдикцією власника.

Різні послуги СС, що позначаються в загальному випадку як XaaS (Xas a Service), можна віднести до трьох основних класів[15]:

- «інфраструктура, як послуга» (IaaS, Infrastructure as a service) – оренда потужності серверів і ємності систем поранення центрів обробки даних (ЦОД);

- «програмне забезпечення, як послуга» (SaaS, Software as a Service) – оренда програмного забезпечення (ПО), яке запускається з «хмари»;

- «платформа, як послуга» (PaaS, Platform as a Service) – оренда платформи розробки ПО колективними або індивідуальними розробниками.

Всі інші послуги систем СС (наприклад, BpaaS - «бізнес-процес, як послуга або VsaaS - «відеоспостереження, як послуга»), можна, так чи інакше, віднести до трьох вищевказаних класів звичайних послуг.

Для роботи технологій Інтернету речей можна використовувати і туманні обчислення (Fog Computing). Мається на увазі наближення «Хмари до землі», в даному випадку «туман» - це різновид хмарних сервісів, розташованих не десь в недоступних висотах, а в навколишньому середовищі. Інакше кажучи, Fog computing не альтернатива, а доповнення до Cloud Computing, і можуть виникнути ситуації їх спільної дії (наприклад, виконання аналітичного додатку), і в такому випадку Cloud надасть послугу Fog.

Туманні обчислення доповнюють хмарні обчислення і забезпечують взаємодію розумних речей між собою і хмарними ЦОД в вигляді трирівневої ієрархічної структури. Верхній рівень займають тисячі хмарних ЦОД, що надають ресурси, необхідні для виконання серйозних, наприклад аналітичних,

програмних додатків IoT. Рівнем нижче розташовуються десятки тисяч розподілених керуючих ЦОД, в яких міститься «інтелект» Fog Computing, а на нижньому рівні знаходяться мільйони обчислювальних пристроїв розумних речей[15].

Fog Computing можна визначити як в максимальному ступені віртуалізовану платформу, що підтримує три основних типи сервісів, утворюючих міжмашинні комунікації M2M: обчислення, зберігання та мережу. Завдання Fog Computing полягає в забезпеченні взаємодії мільярдів пристроїв між собою і з хмарними ЦОД[15].

Парадигма Fog Computing відрізняється від Cloud Computing за цілою низкою параметрів[15]:

1. Розподіл обчислювальної потужності і реальний час. Значні обчислювальні ресурси можуть бути розміщені на периферії мережі, причому не повинно бути залежності від координат того місця, де знаходиться пристрій, і при тому робота в режимі реального часу передбачає низький рівень затримок при обміні даними, до того ж може статися конвергенція двох існуючих, довгий час автономно, одна від одної систем – управління бізнесом і технологічними системами.

2. Географічне розміщення компонентів. Модель розподілу сервісів в Fog Computing менш централізована, ніж для хмар, а в окремі пристрої можуть бути пов’язані між собою потоками даних і надавати один одному «важкі» сервіси.

3. Великий об’єм зовнішніх даних. Пристрої, оснащені численними сенсорами, можуть в реальному часі генерувати гігантські обсяги даних.

4. Складна топологія. Мільйони географічно розподілених вузлів можуть створювати різноманітні і недетерміновані заздалегідь зв’язки.

5. Мобільність і гетерогенність. Мобільність пристроїв зажадає використання альтернативних протоколів, наприклад протоколу маршрутизації

LISP (Locator / ID Separation Protocol), який дозволяє розділити функціональність IP-адрес на 2 частини: ідентифікатори хостів і локатори маршрутизації. Концепція передбачає установку тунельних маршрутизаторів, які будуть додавати LISP-заголовки в інформаційні пакети по мірі їх руху по мережі.

Повсюдна комп'ютеризація (Ubiquitous Computing).

У 1991 році дослідник лабораторії Xerox PARC Марк Вейзер висунув концепцію майбутнього світу, «багато і непомітно насиченого сенсорами, дисплеями і обчислювальними елементами, з'єднані в єдину мережу, які є невід'ємними предметами побуту». Фізична можливість здійснення цієї концепції з'явилася до кінця 2000-х років у міру тотального поширення дешевих і мініатюрних обчислювальних мобільних пристроїв, бездротових мереж і супутникової навігації.

Повсюдне обчислення означає створення інтелектуальних інформаційних систем. В тотальному обчисленні головним є користувач, який повинен мати доступ до обчислювального обслуговування безперервно, цілодобово, кожного дня, причому обслуговування різного виду – від інтелектуальних обчислень до управління домашніми приладами. Так, наприклад, система персональної допомоги (Personal Assistance System, PAS) допомагає людям похилого віку в самообслуговуванні з використання бездротової мережі, який об'єднує RFID-рідери, медтехніку з Bluetooth-інтерфейсом, програмне забезпечення і апаратуру, які відстежують переміщення людини в житлі, датчики падіння, систем безпеки і т.п.

Можна виділити чотири основні характеристики повсюдного комп'ютингу[15]:

1) максимальне використання особистого розумного простору, що оточує нас на роботі, в транспорті, вдома пристрої з комп'ютерним управлінням, необхідними датчиками і виконавчими механізмами;

2) невидимість – мінімізація звернення уваги користувача на керування речами, що оточуючими його;

3) локальна масштабованість – будь-яка точка особистого розумного простору має бути настільки обчислювально «потужною», наскільки цього потребує користувач;

4) маскування неоднорідностей – під неоднорідністю розуміється відмінності як в технічному плані, так і не технічні – організаційні структури, бізнес-процеси, економічні фактори.

До цього списку можна віднести знання контексту, тобто користувач існує в персональному розумному просторі не «наосліп», а представляючи собі, усвідомлюючи контекст. В деякому відношенні це суперечить властивості невидимості, однак, насправді, повинен існувати розумний баланс між невидимістю і знанням контексту.

Взаємодія Інтернету речей та зазначених вище трьох перспективних інфокомунікаційних технологій, якщо розглядати на прикладі транспортних додатків, то дані, зібрані за допомогою різних датчиків і сенсорів, розташованих як на транспортних засобах, так і в дорожній інфраструктурі (елементи інфраструктури повсюдної комп'ютеризації), обробляються з використанням спеціалізованого хмарного транспортного додатку в реальному часі, яке для своєї роботи задіє ресурси систем великих даних.

Таким чином, користувач, який звернувся за відповідною послугою до хмари «транспорт», отримає всю необхідну інформацію про транспортне обслуговування швидко і ефективно.

1.4 Аналіз протоколів Інтернету речей

Протоколи Інтернету речей є важливою частиною стека технологій Інтернету речей, без них обладнання буде звичайним залізом і буде безкорисним, оскільки протоколи Інтернету речей дозволяють IT-підрозділу обмінюватися даними. З цих переданих фрагментів даних можна витягти корисну інформацію для кінцевого користувача, і завдяки цьому розгортання мережі стає економічно вигідним[15].

Існують різні підходи до класифікації протоколів Інтернету. Один з них запропонував об’єднати протоколи по межі застосування в клієнт-серверній архітектурі мереж:

- D2D (пристрій до пристрою) – протокольні взаємодії навколишніх пристроїв між собою;
- D2S (пристрій до серверу) – для передачі даних, зібраних пристроями на сервери для обробки;
- S2S (сервер до сервера) – протоколи взаємодії серверів друг з іншими.

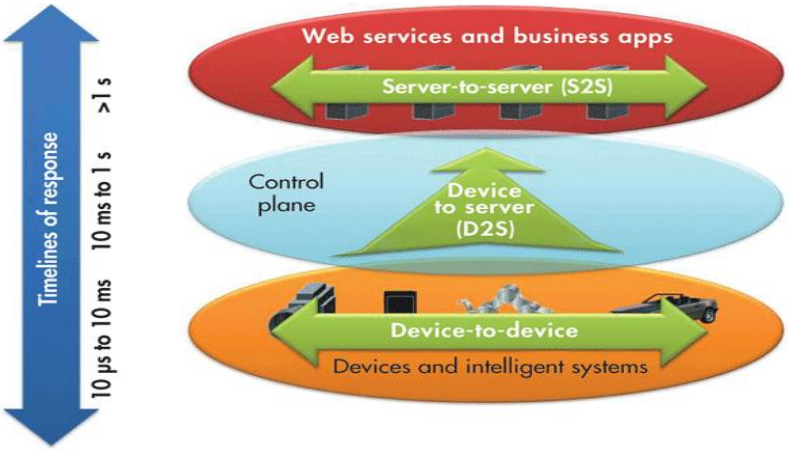

Рис. 1.6. Групи протоколів в клієнт-серверній архітектурі

Інший має таку структуру: замість того, щоб намагатися встановити всі протоколи IoT поверх існуючих моделей архітектури, таких як OSI, було

розбито протоколи на наступні групи, щоб забезпечити певний рівень організації:

1) Інфраструктурна (6LowPAN, Ipv4 / Ipv6, RPL);
2) Ідентифікація (EPC, uCode, Ipv6, URI);
3) Комунікації / Транспорт (Wifi, Bluetooth, LPWAN);
4) Відкриття (Physical Web, Mdns, DNS-SD);
5) Протоколи даних (MQTT, CoAP, AMQP, Websocket, Node);
6) Управління пристроями (TR-069, OMA-DM);
7) Семантичний (JSON-LD, Web Thing Model);
8) Багаторівневі фрейморки (Alljoyn, IoTivity, Weave, Homekit).

1.4.1 Message Queuing Telemetry Transport (MQTT)

MQTT – це протокол обміну повідомленнями з шаблоном публікації-підписки (pub / sub) і призначений для зв’язку комп’ютеризованих пристроїв, підключених до локальної або глобальної мережі, між собою і різними громадськими чи приватними веб-сервісами. Його завдання – замінити приватні технології, використовувані різними компаніями і стати таким же стандартом обміну даними в мережі Інтернет, як протокол HTTP. [3]

Протокол MQTT спочатку був створений для датчиків, які відстежують стан труб, проте пізніше сфера його діяльності була розширена і він знайшов своє застосування у безлічі вбудованих рішень, в тому числі в смартфонах. Так соціальна мережа Facebook застосовує цей протокол для обміну повідомленнями (Facebook Messenger).

У мережі на базі протоколу MQTT розрізняють 3 об’єкти:

1) видавець (publisher) – MQTT-клієнт, який при виникненні певних подій передає інформацію про них в брокер;

2) брокер (broker) – MQTT-сервер, який приймає інформацію від видавців і передає її відповідним підписникам, в складних системах може виконувати також різні операції, пов’язані з аналізом і обробкою даних, що надійшли;
3) підписник (subscriber) – MQTT-клієнт, який після підписки у відповідного брокера більшу частину часу «слухає» його і постійно готовий до приймання та обробки вхідного повідомлення від брокера.

Протокол знайшов широке застосування в таких пристроях IoT, як електролічильники, транспортні засоби, сповіщувачі, промислове чи санітарне обладнання, MQTT добре відповідає наступним потребам [6]:

- мінімальне використання пропускної здатності;
- робота з бездротовими мережами;
- низьке споживання енергії;
- висока надійність при необхідності;
- використовує мало ресурсів для обробки та пам’яті.

Незважаючи на свої характеристики, MQTT може бути проблематичним для деяких дуже обмежуючих пристроїв, через факт передачі повідомлень через TCP та керування назви довгих тем. Це вирішено за допомогою варіанту MQTT-SN, який використовує UDP та підтримує індексацію назв теми. Однак, незважаючи на широке прийняття, MQTT не підтримує чітко визначену модель представлення даних та структуру управління пристроями, що робить реалізацію своїх можливостей управління даними та можливостями управління пристроями повністю орієнтованими на платформу чи постачальника[15].

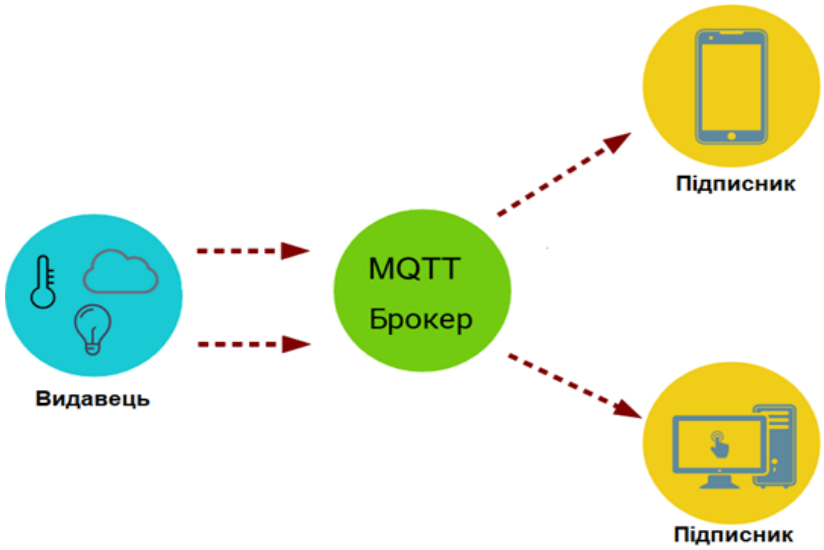

Рис. 1.7. Схематичне зображення принципу дії протоколу MQTT

1.4.2 Constrained Application Protocol (CoAP)

Продовжуючи тему протоколів Інтернету речей розглянемо другий за поширеними після протоколу MQTT – CoAP.

CoAP (протокол обмеженого застосування) був розроблений для вирішення завдань взаємодії пристроїв з обмеженими ресурсами. Практично всі кінцеві пристрої Інтернету речі можна віднести до цього визначення. І саме такі пристрої виконують основну роботу. Концепція CoAP значно відрізняється від MQTT і орієнтується на взаємодію точки-точки (клієнт-сервер). Клієнт звертається до серверу і відправляє найпростіші команди типу PUT, POST, GET, DELETE, сенс яких зрозумілий з назви та аналогічний HTTP. З цієї точки зору можна сказати «із спрощеннями», що COAP – це такий спрощений HTTP, в якому мало ресурсів. В результаті варто відмітити легку та просту інтеграцію CoAP с HTTP. За допомогою звичайного браузеру користувач може інтегрувати систему управління пристроями Інтернету речей в звичайний веб-додаток та не помічати цього, а в деяких випадках навіть можна і не здогадуватись цього. З цієї точки зору протокол відповідає вимогам RESTfull. [7]

CoAP покладається на протокол User Datagram (UDP) для встановлення безпечного зв’язку між кінцевими точками. Використовуючи широкомовну передачу та багатоадресну передачу, UDP здатний передавати дані багатьом хостам, зберігаючи швидкість зв’язку та низьке використання пропускної здатності, що робить його відмінною підсистемою для бездротових мереж, зазвичай використовуваних у середовищі M2M з обмеженими ресурсами. [6]

CoAP має якість обслуговування, яка використовується для контролю надсилаються повідомлень і позначає їх як «підтверджені» або «непідтверджені» відповідно, що вказує, чи повинен одержувач повернути « підтвердження» чи ні. Інші цікаві особливості CoAP полягають у тому, що він підтримує механізм узгодження вмісту та механізм виявлення ресурсів. Крім передачі даних IoT, CoAP використовує безпеку транспортного рівня Datagram Transport Layer Security (DTLS) для безпечного обміну повідомленнями на транспортному рівні. CoAP повністю відповідає потребам надзвичайно легкого протоколу, щоб задовольнити потреби пристроїв, що працюють на батареях або з низьким енергоспоживанням[15].

1.4.3 Extensible Messaging and Presence Protocol (XMPP)

Розроблений у 1999 році спільнотою Jabber з відкритим кодом та спочатку призначений для обміну повідомленнями в реальному часі, цей протокол IoT для комунікації проміжного програмного забезпечення, орієнтованого на повідомлення, заснований на мові XML. Це дозволяє в режимі реального часу обмінюватися структурованими, але розширюваними даними між двома або більше мережевими клієнтами[15].

З моменту свого створення XMPP широко застосовується як протокол зв’язку. З часом і з появою легкої специфікації XMPP: XMPP-IoT, вона стала використовуватись в контексті Інтернету речей. Будучи стандартом, який

підтримує відкрита спільнота, сильні сторони XMPP IoT – це адресація та можливість масштабування, що робить його ідеальним для розширення IoT, орієнтованого на споживача[15].

Серед недоліків використання XMPP в IoT-зв'язку слід зазначити, що він не пропонує ні якості обслуговування, ні шифрування в кінці. Зважаючи на ці обмеження, серед інших, передбачається, що його застосування в IoT залишатиметься слабко пов'язаним з галуззю, оскільки протокол, безумовно, не стане стандартним, що використовується для обміну даними та управління пристроями з обмеженими ресурсами, як і MQTT або LwM2M[15].

1.4.4 Lightweight M2M (LwM2M)

Lightweight M2M був розроблений по суті для зменшення енергоспоживання та споживання даних, таким чином визнаючи та реагуючи на вимоги зростаючого ринку M2M для пристроїв малої потужності з обмеженими можливостями обробки та зберігання. [8]

Специфікація даного протоколу також описує безліч типових функцій управління пристроєм, такі як дії віддалених пристроїв, оновлення прошивки та програмного забезпечення (FOTA & SOTA), моніторинг і управління підключенням, включаючи управління стільниковим зв'язком і налаштування.

На відміну від будь-якого іншого протоколу IoT на ринку, архітектура LwM2M підтримує чотири логічні інтерфейси, які допомагають стандартизувати спосіб власне управління пристроєм та телеметрією[15]:

- Інтерфейс завантаження – цей інтерфейс дозволяє керувати пристроями без головного пристрою. Це означає, що можна налаштувати пристрій для надання потрібної послуги без необхідності попереднього налаштування на заводі, що значно скорочує витрати та оптимізує час виходу продукту чи послуги на ринок.

- Інтерфейс реєстрації клієнта – повідомляє сервер про «існування» клієнта та підтримуваний функціонал. Крім того, він дозволяє оновити прошивку та програмне забезпечення по повітрю[15].
- Інтерфейс управління пристроєм та підтримка сервісу – LwM2M дозволяє постачальнику отримати доступ до примірників та ресурсів об’єкта, що дозволяє йому змінювати налаштування та параметри пристрою.
- Інтерфейс інформаційних звітів – завдяки взаємодії з публікацією/підпискою користувач може отримувати з пристроїв звіти про помилки, коли сервіс не працює належним чином, а також надсилати запити про стан пристрою[15].

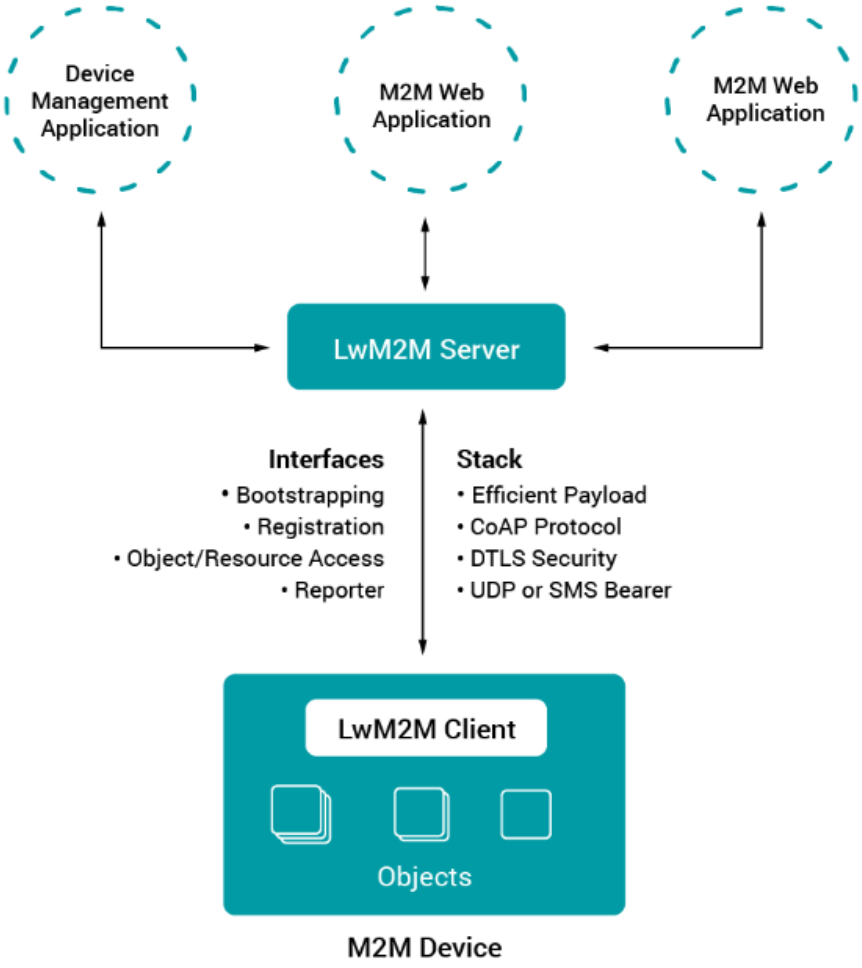

Рис. 1.8. Схематичне зображення принципу дії протоколу LwM2M

1.4.5 Data-Distribution Service (DDS)

Протокол DDS також був розроблений на основі методології публікації-підписки. Розроблений групою Object Management Group (OMG), протокол DDS в режимі реального часу для M2M зв'язку забезпечує масштабований, надійний, високоефективний та сумісний обмін даними між підключеними пристроями, незалежними від апаратного забезпечення та програмної платформи. DDS підтримує архітектуру без брокерів та багатоадресну передачу для забезпечення високої якості QoS та забезпечення сумісності[15].

Архітектура протоколу DDS базується на рівні Data Centric Publish-Subscribe (DCPS) та рівні Data-Local Reconstruction Layer (DLRL). В той час як DCPS відповідає за масштабованість та розподіл даних з урахуванням ресурсів, DLRL пропонує інтерфейс для функціональних можливостей DCPS, що дозволяє передавати дані між об'єктами, підключеними до IoT[15].

Не дивлячись на те, що DDS є не типовим рішенням для IoT, він все ще знаходить своє застосування в деяких розгортаннях Індустріального Інтернету речей, таких як: управління повітряним рухом, управління розумними мережами, автономні транспортні засоби, транспортні системи, робототехніка, виробництво електроенергії та послуги охорони здоров'я. В цілому DDS може використовуватися для управління обміном даними між легкими пристроями та взаємозв'язку великих, високоефективних сенсорних мереж. Він також може надсилати та отримувати дані з хмари[15].

Висновки

В даному розділі було розглянуто архітектуру Інтернету речей та її складові. Технології, які використовує IoT за для забезпечення ефективної роботи та виконання поставлених задач. Був проведений аналіз перспективних інфокомунікаційних технологій з якими взаємодії або може взаємодіяти

Інтернет речей. Також було розглянуто найбільш використовувані протоколи IoT.

РОЗДІЛ 2
АНАЛІТИЧНА ПЛАТФОРМА ДЛЯ ІНТЕРНЕТУ РЕЧЕЙ

2.1 Концепція аналітичної IoT платформи

Цифрова трансформація дедалі більше зосереджується на підключених пристроях, штучному інтелекті та інших рішеннях, щоб зробити робочі місця автоматизованими та ефективними. Розумні пристрої — від заводського обладнання до офісної техніки — можуть допомогти в автоматизації завдань, а також розкрити інформацію про компанію, будівлю, співробітників і клієнтів. Підключені активи допомагають відстежувати деталі, пов’язані з результатами, продуктивністю та взаємодією з програмними платформами, співробітниками, клієнтами та іншими додатками, підключеними до мережі. Аналітичні рішення Інтернету речей дають можливість бізнес-користувачам сортувати та розуміти ці висновки. Вони можуть дізнатися більше про свої бізнес-операції з точки зору повсякденних «речей», які використовуються в компанії, незалежно від їх вартості, розміру чи функції.

Багато платформ цієї категорії пропонують інструменти для перегляду та перетворення необроблених даних у формати для спільного використання. У деяких випадках це тягне за собою інтеграцію з програмним забезпеченням для візуалізації даних, платформами бізнес-аналітики або іншими інструментами, що використовуються для аналізу даних. Окрім збору та підготовки цінних даних із розумних пристроїв, деякі платформи надають інструменти для моніторингу та звітності в реальному часі, допомагаючи користувачам приймати миттєві рішення на основі миттєвих подій. Потокова передача потужних статистичних даних у реальному часі допомагає особам, які приймають рішення, коригувати та покращувати процеси, коли задіяні ці пристрої, не чекаючи довгих звітів.

Завдяки сучасній технології периферійних обчислень дані, зібрані на цих платформах, можна обробляти та зберігати на межі мереж компанії, а не централізованих сховищ даних. Це допомагає доставити потрібні дані з більш високою швидкістю, не витрачаючи пропускну здатність критичних систем. Щоб створити сценарій граничних обчислень для пристроїв IoT і даних, які вони збирають, компанія повинна налаштувати граничні пристрої (наприклад, маршрутизатори, інтегровані пристрої доступу), які контролюють потік даних. Оскільки компанія створює свій Інтернет речей, аналітичні платформи допомагають розкрити весь потенціал цих пристроїв без шкоди для продуктивності їхніх активів або IT-інфраструктури. Деякі платформи Інтернету речей включають аналітичні рішення або певні функції звітності, але спеціальні аналітичні платформи, як-от у цій категорії, пропонують більш глибоке уявлення про пристрої, мережі та інші пов’язані функції в організації.

2.2 Аспекти використання та загальна структура аналітичних платформ

IoT аналітика — це програми, які допомагають зрозуміти величезний обсяг даних, що генеруються підключеними пристроями IoT. Іншими словами, Інтернет речей — це просто екосистема, фізичні речі, які підключені до Інтернету, і ми бачимо результати через аналітичну платформу IoT. Ви також можете дізнатися більше про виявлення аномалій у реальному часі для когнітивного інтелекту. Наприклад, ці фізичні речі включають смартфони таблетки, датчики тощо.

Численні червоні прапорці можуть допомогти підприємствам визначити, що їм не вистачає належної аналітичної системи. Тому розглянемо детальніше про основні проблеми, з якими стикаються організації без високофункціональної аналітичної платформи.

- Слабке управління даними з різних джерел

  Численні компанії можуть мати кілька потокових зовнішніх і внутрішніх джерел даних, таких як системи CRM, веб-додатки, бази даних, системи керування робочими процесами тощо. Зростання кількості розрізнених джерел даних у компанії без єдиного місця для управління може призводять до поганої аналітики даних, невидимості даних та недостатнього керування даними. Більше того, таке погане управління даними призводить до виснажливого структурування даних.

- Звіти компанії часто застарілі та неактуальні

  Відсутність динамічної звітності часто може призвести до нездатності проаналізувати результати діяльності компанії на більш глибоких рівнях і з'ясувати, які області відстають, а які працюють вище очікувань. Важливість регулярних і точних звітів велика для всіх керівників, оскільки слабка система звітності може вплинути на важливі бізнес-рішення і, як наслідок, призвести до невдоволення клієнтів і збільшення відтоку клієнтів.

- Невідповідність даних через погане керування сховищем

  Невідповідність даних, втрата, непотрібне дублювання тощо можуть бути результатом збільшення обсягу даних всередині організації та слабкого управління зберіганням даних. Компанії, які не мають таких рішень, можуть зіткнутися з проблемою зберігання структурованих і неструктурованих даних різних форматів і з кількох джерел.

- Відсутність єдиного місця для візуалізації даних

  Відсутність багатогранної інформаційної панелі для моніторингу продуктивності є ще однією причиною неефективної аналітики. Компанії з постійними потоками даних з кількох джерел часто можуть знадобитися в реальному часі знайти інформацію про KPI компанії (критичні показники

ефективності), точну статистику від окремого відділу тощо. Відсутність централізованого місця для візуального представлення даних може призвести до у неточному розумінні життєво важливих показників, нездатності відповідати новим тенденціям та вчасно реагувати на сповіщення.

Таким чином, підприємства можуть виявити неефективну аналітику даних за певними показниками. На щастя, це не кінець світу, і всі можливі проблеми можна вирішити за допомогою певної функції повноцінної аналітичної платформи, яку ми розглянемо нижче.

Розглядаючи архітектуру та принципи роботи IoT-систем, ми вже згадували, що найбільш інтелектуальна частина роботи з аналізу даних виконується у хмарі за допомогою спеціальних засобів Big Data, об'єднаних у загальну платформу. А тепер розглянемо функції IoT-платформ та технології, на яких засновані ці хмарні рішення.

В IoT-платформах використовуються такі проколи:

- MQTT – Протокол MQTT використовує архітектуру публікації/підписки. Центральною точкою зв'язку цього протоколу є брокер MQTT. Кожен клієнт включає назву теми під час публікації даних брокеру. Теми відповідають за маршрутизацію інформації для брокера. Кожен клієнт, який хоче отримувати повідомлення, підписується на певну тему, і брокер доставляє клієнту всі повідомлення з відповідною темою.
- COAP – це веб-протокол, розроблений для підключення легких пристроїв до Інтернету речей. Як і протоколи HTTP, COAP також використовується модель запит-відповідь. Він також дозволяє здійснювати виклики API GET, PUT, POST, DELETE через URL.

- AMQP - є відкритим стандартом для передачі повідомлень між додатками та організаціями. Він з'єднує систему, забезпечує бізнес-процеси необхідною інформацією.
- HTTP – це стандартний протокол для веб-сервісів, який досі використовується в аналітичних рішеннях IoT. Найпопулярніший архітектурний стиль під назвою RESTFul широко використовується в мобільних і веб-додатках, і його необхідно враховувати в рішеннях IoT.
- DDS – є стандартом для аналітики Інтернету речей у реальному часі, масштабованого та високопродуктивного міжмашинного зв’язку. DDS можна розгорнути як на пристроях з низьким розміром, так і в хмарі.

Рис. 2.1. Структурна схема принципу роботи аналітичної IoT-платформи

Збір і прийом даних - це буде перший крок у обробці даних із пристрою IoT. Збір може відбуватись даних із датчиків, пристроїв або за допомогою іншої платформи. Потокова обробка даних - це безперервний запит даних, коли вони надходять із джерела введення. Обробка потоків робить можливим аналіз у реальному часі, об’єднуючи програми з аналізом. Обробка потоку в реальному часі відноситься до обробки даних потоком даних, зібраним з пристрою IoT в режимі реального часу. Аналітика в режимі реального часу використовує дані, коли вони доступні. Більшість аналізу в реальному часі базується на концепції реагування на дані в момент їх надходження. Час відповіді завжди є основною та ключовою віссю для аналізу. Швидкість потоку даних також може змінюватися залежно від часу. Отже, наш потік даних повинен мати можливість

обробляти дані будь-якого розміру та з будь-якою швидкістю. Платформа повинна бути достатньо розумною, щоб автоматично збільшуватися і зменшуватися відповідно до навантаження на конвеєр даних. Обробка потоків допомагає отримувати статистику набагато швидше, ніж інші рішення. Наступним кроком є зберігання вже опрацьованої та проаналізованої інформації, після цього процесу іде найважливіша частина – це представлення результатів. Звіт може бути сформований у вигляді спеціальної інформаційної панелі для візуалізації даних, графіків тощо. [9]

2.3 Основні функції аналітичної платформи

Чим ретельніше підійти до вибору набору функцій для аналітичної платформи, тим легше потенційні користувачі досягнуть своїх цілей. Отже, першим кроком на шляху до створення аналітичного програмного забезпечення є поглиблений аналіз цільової аудиторії, яка буде використовувати надане програмне забезпечення. Потрібно враховувати параметрами[10]:

- тип і розмір організацій, які будуть використовувати ваше програмне забезпечення;
- потреби користувачів щодо набору інструментів аналітики;
- приблизний обсяг даних всередині організацій;
- тип користувачів (адміністратори, менеджери, які матимуть повний доступ до всіх інструментів аналітики, або користувачі, які періодично переглядатимуть аналітику тощо);
- необхідність інструментів для співпраці тощо.

Проаналізувавши, було виділено ключові функції, необхідні для створення аналітичної платформи. Майте на увазі, що набір інструментів веб-аналітики вашої платформи може бути розширений і налаштований відповідно до потреб вашої потенційної цільової аудиторії.

1. Розширена звітність

Це одна з основних функцій, яка дозволяє користувачам упорядковувати свої дані в інформаційні зведення і допомагає підприємствам відстежувати, як працюють різні сфери їхньої компанії. Створюючи аналітичну платформу, переконайтеся, що ви створили достатню кількість інструментів для звітності, таких як таблиці, діаграми, штрих-коди, інфографіку тощо, які допоможуть вашим клієнтам керувати звітами. Крім того, переконайтеся, що розроблені інструменти покривають весь спектр потреб ваших користувачів щодо звітності, дозволяючи їм виконувати планування доставки звітів, форматування, перетворення та налаштування.

2. Надійна візуалізація даних

Сама суть аналітичної платформи полягає в тому, щоб дозволити її користувачам представляти дані візуально спрощеним та ефективним способом. Подумайте про надання клієнтам широкого спектру інструментів графічного представлення даних, таких як інтерактивні інформаційні панелі, індикатори шаблонів, діаграми тощо. Візуалізація даних допомагає компаніям перетворювати нудні числові дані в яскраве представлення їхньої ефективності, виявляти приховані закономірності та знаходити кореляцію між різними наборами даних.

3. Фільтрація та сортування даних

Іншою корисною функцією є фільтрація та сортування даних, яка може допомогти користувачам у зручному та прискореному способі керування даними. Коли ви створюєте програму для веб-аналітики, не забудьте включити розширені фільтри, щоб користувачі могли звузити свій пошук і зосередитися на аналізі конкретних наборів даних компанії. Функція сортування дозволить

користувачам упорядкувати свої набори даних за певними значеннями у змістовному порядку для кращого розуміння та подальшого аналізу.

4. Трансформація даних

Інструменти для перетворення даних життєво необхідні для процвітання аналітичної платформи. Вони допомагають витягти дані з місця, де вони зберігаються, перетворити їх у потрібний формат і передати їх у цільове місце, де вони будуть остаточно проаналізовані. Оскільки організація може мати кілька джерел даних (таблиці, файли тощо), аналітична платформа повинна обробляти складні процедури трансформації, щоб користувачі могли отримувати цінну бізнес-інсайт із правильно оброблених і структурованих даних.

5. Інтеграція джерел даних

Розробляючи аналітичний додаток, переконайтеся, що ваші користувачі мають зручний інструмент для отримання інформації з кількох сховищ даних (CRM, ERP, електронні таблиці тощо). Це стане можливим завдяки розробці єдиного сховища даних, де неструктуровані та необроблені дані будуть безпечно зберігатися з різних джерел, несумісних один з одним. Така функція забезпечить прискорене та спрощене вилучення та подальшу обробку даних.

6. Контрольований обмін

Оскільки інформація, накопичена різними відділами та командами, зберігається в різних джерелах даних, користувачі повинні мати ефективний набір інструментів для спільного керування даними та їх візуалізації. Розробляючи аналітичну програму, переконайтеся, що ваш інтерфейс містить ефективні інструменти для спільного використання даних і звітів, коментування, закріплення, додавання анотацій, сповіщень та інші необхідні інструменти для спільного аналізу даних.

7. Веб та мобільний доступ

В епоху оцифрування необхідно забезпечити ваших користувачів веб та мобільний доступ до наданих послуг. Співробітники повинні мати можливість робити вибір на основі даних зручним і доступним способом, не змушуючи бути прикріпленим виключно до своїх столів. Якщо ви не плануєте мати мобільну версію своєї аналітичної платформи, подумайте про те, щоб ваш веб-додаток був зручним для мобільних пристроїв, щоб отримати більше лояльних клієнтів.

Як мінімум, аналітичні функції повинні виконувати:

- Описову аналітику того, що відбувається в будь-який момент часу.
- Прогнозну аналітику для запобігання небажаних і дорогих простоїв.
- Рекомендаційну аналітику, яка допоможе відповісти на запитання про рентабельність інвестицій, нові бізнес-моделі та способи максимізації результату та ефективності.

За даними 451 Research, критичним аспектом будь-якої платформи IoT загалом є її здатність керувати обсягами згенерованих даних і надавати користувачам можливість інтегрувати результати, які можна діяти.[11]

2.4 Основні переваги аналітичних платформ

Платформи аналітики охоплюють різні інструменти для інтеграції, збору, обробки та представлення даних, допомагаючи підприємствам приймати більш точні бізнес-рішення. Цей тип програмного забезпечення дозволяє компаніям інтегрувати свої дані з кількох джерел даних, охоплювати весь життєвий цикл даних, зручно їх візуалізувати тощо. Отже, давайте розглянемо основні переваги створення платформи для аналізу даних.

1. Ефективне управління даними

Аналітична платформа є одним із найзручніших варіантів BI для ефективного управління даними. Це гарантує, що дані можна легко отримати та отримати з усіх сховищ компанії, а потім структурувати для швидкого аналізу.

2. Масштабованість

Масштабованість Створюючи платформу для аналізу даних, ви забезпечите належну обробку зростаючих обсягів даних без необхідності жертвувати продуктивністю системи.

3. Дієві рішення

Платформа допомагає відстежувати всі взаємодії клієнтів із наданими послугами чи продуктами: як вони прокручують сайт, причини, чому вони приймають певні рішення про покупку, момент, коли вони залишають сайт тощо. Це дозволяє власникам бізнесу отримувати більш точні та цінні ідеї на основі зібраних наборів даних. Отримані бізнес-інсайти можуть допомогти створити потужніші маркетингові кампанії, покращити досвід клієнтів, персоналізувати його та залучити нових користувачів.

4. Краще представлення даних

Надійні аналітичні платформи надають користувачам численні інструменти візуалізації даних, такі як багатогранні інформаційні панелі, які з’єднують усі збережені дані з різних джерел, шаблони, інтерактивні діаграми, інтерфейси перетягування для легкого створення документів із графічними даними тощо. Гнучкі дані візуалізація допомагає аналізувати ефективність компанії та витягувати сенс із даних.

5. Підвищена лояльність клієнтів

Кращий аналіз взаємодії клієнтів із наданим продуктом означає більш глибоке розуміння їхніх проблем, потреб і бажань. Інструменти, які можна отримати шляхом створення аналітичного програмного забезпечення, дозволяють бізнесу глибше зануритися в свідомість цільової аудиторії,

адаптувати та покращити надані послуги відповідно до її потреб, а також перетворити клієнтів на лояльних.

Як бачите, переваги аналітичної платформи можуть значно підвищити бізнес-можливості компанії та надати вичерпний огляд її ефективності. Однак успіх аналітичної платформи значною мірою залежить від її функціональності.

2.5 AWS IoT Analytics

AWS IoT Analytics автоматизує дії, необхідні для аналізу даних із пристроїв IoT. AWS IoT Analytics фільтрує, перетворює та збагачує дані Інтернету речей, перш ніж зберігати їх у сховищі часових рядів для аналізу. Ви можете налаштувати службу для збору лише потрібних даних із ваших пристроїв, застосувати математичні перетворення для обробки даних і збагатити дані метаданими для специфіки пристрою, такими як тип пристрою та його розташування, перш ніж зберігати їх. Потім ви можете аналізувати свої дані, виконуючи запити за допомогою вбудованого механізму запитів SQL, або виконувати складнішу аналітику та висновок машинного навчання. AWS IoT Analytics забезпечує розширене дослідження даних завдяки інтеграції з Jupyter Notebook. AWS IoT Analytics також забезпечує візуалізацію даних завдяки інтеграції з Amazon QuickSight.

Традиційні інструменти аналітики та бізнес-аналітики призначені для обробки структурованих даних. Необроблені дані IoT часто надходять з пристроїв, які записують менш структуровані дані (наприклад, температура, рух або звук). В результаті дані з цих пристроїв можуть мати значні прогалини, пошкоджені повідомлення та помилкові показання, які необхідно очистити, перш ніж проводити аналіз. Крім того, дані IoT часто мають значення лише в контексті інших даних із зовнішніх джерел. AWS IoT Analytics дозволяє вирішити ці проблеми та збирати великі обсяги даних пристрою, обробляти

повідомлення та зберігати їх. Потім ви можете запитати дані та проаналізувати їх. AWS IoT Analytics включає готові моделі для поширених випадків використання Інтернету речей, щоб ви могли відповісти на запитання, наприклад, які пристрої ось-ось вийдуть з ладу або які клієнти ризикують відмовитися від своїх носимих пристроїв.

Рис. 2.2. Модель використання AWS IoT Analytics

Ключові функції[12]:

1) Збір даних - збір даних з пристроїв в різних форматах та різній частоті.
   - Інтегрована з AWS IoT Core — AWS IoT Analytics повністю інтегрована з AWS IoT Core, тому вона може отримувати повідомлення від підключених пристроїв, коли вони надходять.
   - Використовуйте пакетний API для додавання даних з будь-якого джерела — AWS IoT Analytics може отримувати дані з будь-якого джерела через HTTP. Це означає, що будь-який пристрій або служба, підключені до Інтернету, можуть надсилати дані в AWS IoT Analytics. Щоб отримати додаткові відомості.
   - Збирайте лише ті дані, які потрібно зберігати та аналізувати. Ви можете використовувати консоль AWS IoT Analytics, щоб налаштувати AWS IoT Analytics для отримання повідомлень від пристроїв через фільтри тем MQTT у різних форматах і частотах. AWS IoT Analytics перевіряє, що дані знаходяться в межах

визначених вами параметрів, і створює канали. Потім служба направляє канали до відповідних конвеєрів для обробки, перетворення та збагачення повідомлень.

2) Обробка даних – збагачення даних зовнішніми джерелами.
   - Очищення та фільтрація — AWS IoT Analytics дозволяє визначати функції AWS Lambda, які запускаються, коли AWS IoT Analytics виявляє відсутні дані, тож ви можете запускати код, щоб оцінити та заповнити прогалини. Ви також можете визначити максимальні та мінімальні фільтри та відсоткові пороги, щоб звести до нуля викиди у ваших даних.
   - Трансформація — AWS IoT Analytics може перетворювати повідомлення за допомогою визначеної вами математичної або умовної логіки, щоб ви могли виконувати звичайні обчислення, як-от перетворення Цельсія в Фаренгейт.
   - Розширення — AWS IoT Analytics може збагатити дані зовнішніми джерелами даних, як-от прогноз погоди, а потім направляти дані до сховища даних AWS IoT Analytics.
3) Збереження – збереження даних в сховищах даних часових рядів для аналізу.
   - Сховище даних часових рядів — AWS IoT Analytics зберігає дані пристрою в оптимізованому сховищі часових рядів для швидшого пошуку та аналізу. Ви також можете керувати дозволами доступу, впроваджувати політику збереження даних та експортувати свої дані до зовнішніх точок доступу.
   - Зберігайте оброблені та необроблені дані — AWS IoT Analytics зберігає оброблені дані, а також автоматично зберігає необроблені дані, щоб ви могли обробити їх пізніше.

4) Аналіз – використання SQL-запити або попередньо створених моделей, щоб виконувати машинне навчання та робити прогнози.
   - Виконуйте спеціальні запити SQL — AWS IoT Analytics надає механізм SQL-запитів, щоб ви могли виконувати спеціальні запити та швидко отримувати результати. Служба дає змогу використовувати стандартні запити SQL для вилучення даних із сховища даних, щоб відповісти на такі запитання, як середня відстань, пройдену парком підключених транспортних засобів, або скільки дверей у розумній будівлі зачинено після 19:00. Ці запити можна використовувати повторно, навіть якщо підключені пристрої, розмір парку та аналітичні вимоги змінюються.
   - Аналіз часових рядів — AWS IoT Analytics підтримує аналіз часових рядів, щоб ви могли аналізувати продуктивність пристроїв з часом і розуміти, як і де вони використовуються, постійно відстежувати дані пристрою, щоб передбачити проблеми з техобслуговуванням, а також контролювати датчики, щоб прогнозувати та реагувати на екологічні умови.
   - Розміщені блокноти для складної аналітики та машинного навчання — AWS IoT Analytics включає підтримку розміщених ноутбуків у Jupyter Notebook для статистичного аналізу та машинного навчання. Сервіс включає в себе набір шаблонів блокнотів, які містять моделі машинного навчання та візуалізації, розроблені AWS. Ви можете використовувати шаблони, щоб розпочати роботу з IoT-випадками, пов’язаними з профілюванням збоїв у пристрої, прогнозуванням подій, таких як низьке використання, що може сигналізувати про відмову клієнта від продукту, або сегментування пристроїв за рівнями використання клієнта (наприклад, активні користувачі, користувачі у вихідні дні). або стан пристрою. Після того як ви

створите блокнот, ви можете помістити його в контейнер і виконати за розкладом, який ви вкажете.

- Прогнозування — статистичну класифікацію можна здійснити за допомогою методу, який називається логістичною регресією. Ви також можете використовувати довгострокову пам’ять (LSTM), яка є потужним методом нейронної мережі для прогнозування результату або стану процесу, який змінюється з часом. Попередньо вбудовані шаблони ноутбуків також підтримують алгоритм кластеризації K-середніх для сегментації пристроїв, який об’єднує ваші пристрої в когорти подібних пристроїв. Ці шаблони зазвичай використовуються для профілювання працездатності пристрою та стану пристрою, наприклад, установок HVAC на шоколадній фабриці або зносу лопатей вітрової турбіни. Знову ж таки, ці шаблони блокнотів можна містити та виконувати за розкладом.

5) Звіт – робота з результатами аналітика та їх візуалізація.
   - Інтеграція Amazon QuickSight — AWS IoT Analytics забезпечує з’єднання з Amazon QuickSight, щоб ви могли візуалізувати свої набори даних на інформаційній панелі QuickSight.
   - Інтеграція консолі. Ви також можете візуалізувати результати або спеціальний аналіз у вбудованому блокноті Jupyter на консолі AWS IoT Analytics.

Компоненти та концепції AWS IoT Analytics:

1) Канал

Канал збирає дані з MQTT і архівує необроблені повідомлення перед публікацією даних у конвеєр. Ви також можете надсилати повідомлення на канал безпосередньо за допомогою API BatchPutMessage. Необроблені

повідомлення зберігаються в пакеті Amazon Simple Storage Service (Amazon S3), яким керуєте ви або AWS IoT Analytics.

2) Потік

Потік отримує дані з каналу і дає змогу обробити повідомлення перед тим, як зберегти їх у сховищі даних. Етапи обробки, які називаються діями, виконують перетворення ваших повідомлень, такі як видалення, перейменування або додавання атрибутів повідомлення, фільтрація повідомлень на основі значень атрибутів, виклик ваших лямбда-функцій у повідомленнях для розширеної обробки або виконання математичних перетворень для нормалізації даних пристрою.

3) Сховище даних

Конвеєри зберігають свої оброблені повідомлення в сховищі даних. Сховище даних не є базою даних, але це масштабоване сховище ваших повідомлень, яке можна запитувати. Ви можете мати кілька сховищ даних для повідомлень, які надходять з різних пристроїв або місць, або фільтрувати за атрибутами повідомлень залежно від конфігурації конвеєра та вимог. Як і в разі необроблених повідомлень каналу, оброблені повідомлення сховища даних зберігаються в сегменті Amazon S3, яким керуєте ви або AWS IoT Analytics.

4) Набір даних

Ви отримуєте дані зі сховища даних, створюючи набір даних. AWS IoT Analytics дозволяє створити набір даних SQL або набір даних контейнера.

Отримавши набір даних, ви можете досліджувати свої дані та отримувати уявлення про свої дані за допомогою інтеграції за допомогою Amazon QuickSight. Ви також можете виконувати більш розширені

аналітичні функції завдяки інтеграції з Jupyter Notebook. Jupyter Notebook надає потужні інструменти для науки про дані, які можуть виконувати машинне навчання та ряд статистичних аналізів. Для отримання додаткової інформації див. Шаблони блокнотів.

Ви можете надсилати вміст набору даних у корзину Amazon S3, забезпечуючи інтеграцію з наявними озерами даних або доступ із внутрішніх додатків та інструментів візуалізації. Ви також можете надсилати вміст набору даних як вхід до служби AWS IoT Events, яка дає змогу відстежувати пристрої чи процеси на предмет збоїв чи змін у роботі, а також ініціювати додаткові дії, коли такі події відбуваються.

5) Набір даних SQL

Набір даних SQL схожий на матеріалізоване представлення з бази даних SQL. Ви можете створити набір даних SQL, застосувавши дію SQL. Набори даних SQL можна генерувати автоматично за регулярним розкладом, вказавши тригер.

6) Набір даних контейнера

Набір даних контейнера дає змогу автоматично запускати інструменти аналізу та генерувати результати. Додаткову інформацію див. у розділі Автоматизація робочого процесу. Він об’єднує набір даних SQL як вхідні дані, контейнер Docker з вашими інструментами аналізу та необхідними файлами бібліотеки, вхідними та вихідними змінними та додатковим тригером розкладу. Вхідні та вихідні змінні повідомляють виконуваному образу, де отримати дані та зберігати результати. Тригер може запустити ваш аналіз, коли набір даних SQL закінчує створення свого вмісту або відповідно до виразу розкладу часу. Набір даних контейнера автоматично запускається, генерує та зберігає результати інструментів аналізу.

7) Тригер

Ви можете автоматично створити набір даних, вказавши тригер. Тригером може бути часовий інтервал (наприклад, створювати цей набір даних кожні дві години) або коли було створено вміст іншого набору даних (наприклад, створити цей набір даних, коли myOtherDataset закінчить створення свого вмісту). Або ви можете створити вміст набору даних вручну за допомогою API CreateDatasetContent.

8) Контейнер Docker

Ви можете створити власний контейнер Docker, щоб упакувати інструменти аналізу або використовувати параметри, які надає SageMaker. Додаткову інформацію див. у статті Контейнер Docker. Ви можете створити власний контейнер Docker, щоб упакувати інструменти аналізу або використовувати параметри, надані SageMaker. Ви можете зберігати контейнер у вказаному вами реєстрі Amazon ECR, щоб він був доступним для встановлення на бажаній платформі. Контейнери Docker можуть запускати ваш власний аналітичний код, підготовлений за допомогою Matlab, Octave, Wise.io, SPSS, R, Fortran, Python, Scala, Java, C++ тощо. Додаткову інформацію див. у розділі Контейнерування блокнота.

9) Дельта вікна

Дельта-вікна — це серія визначених користувачем, не перекриваються та суміжних інтервалів часу. Вікна Delta дозволяють створювати вміст набору даних з новими даними, які надійшли до сховища даних після останнього аналізу, та виконувати їх аналіз. Ви створюєте дельта-вікно, встановлюючи deltaTime у частині фільтрів queryAction набору даних. Для отримання додаткової інформації див. API CreateDataset. Зазвичай ви хочете автоматично створювати вміст набору даних, також налаштувавши

тригер часового інтервалу (triggers:schedule:expression). Це дозволяє фільтрувати повідомлення, які надійшли протягом певного часового вікна, тому дані, що містяться в повідомленнях з попередніх часових вікон, не зараховуються двічі. Для отримання додаткової інформації див. Приклад 6 – створення набору даних SQL із вікном Delta (CLI).

2.6 Microsoft Azure IoT

Дане рішення включає в себе конструювання, інтелектуальний аналіз, виробництво чи інші галузеві рішення, пов'язані з великими обсягами даних із багатьох вхідних даних на основі Інтернету речей.[13]

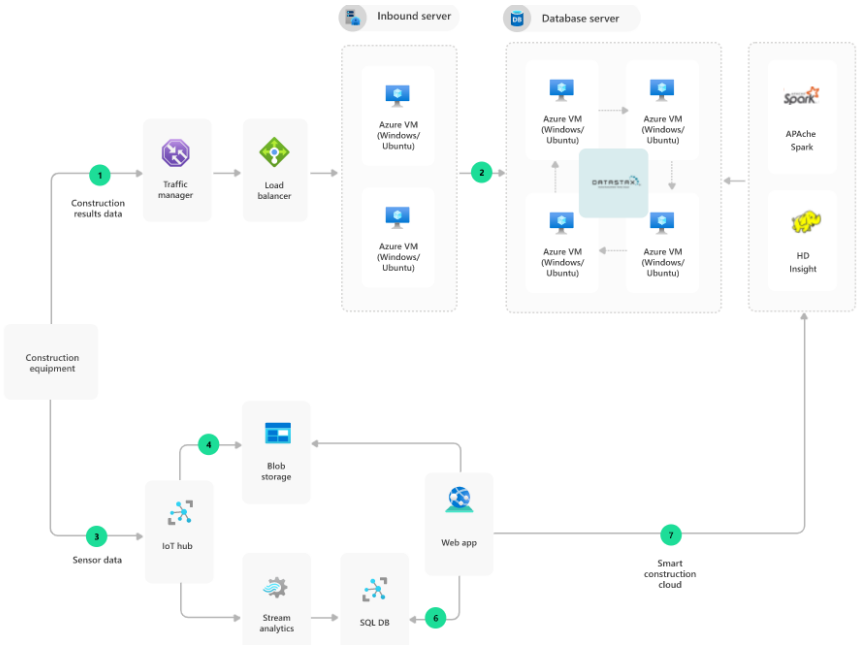

Рис. 2.3. Архітектура Microsoft Azure IoT

Потік даних проходить через рішення в такий спосіб.

1) Устаткування конструювання збирає дані датчиків і надсилає дані результатів конструкції з регулярними інтервалами для завантаження з балансуванням веб-служб, розміщених у кластері віртуальних машин Azure.

2) Веб-служби користувача приймають дані результатів конструювання і зберігають їх в кластері Apache Cassandra, запущеному на віртуальних машинах Azure.
3) Інший набір даних збирається з датчиків Інтернету речей на різні конструкції обладнання та відправляються до Центру Інтернету речей.
4) Зібрані необроблені дані надсилаються безпосередньо з Центру Інтернету речей до сховища BLOB-об'єктів Azure і водночас стають доступними для перегляду та аналізу.
5) Дані, зібрані за допомогою Інтернету речей, обробляються практично в реальному часі шляхом завдання Azure Stream Analytics та зберігаються в базі даних Azure SQL.
6) Веб-програма Smart Construction Cloud доступна аналітикам та користувачам для перегляду та аналізу даних датчиків та зображень.
7) Пакетні завдання ініціюються користувачами веб-застосунків на запит. Завдання виконується в Apache Spark HDINsight і аналізує нові збережені дані в кластері Cassandra.

Дане архітектурне рішення складається з таких компонентів:

- Центр Інтернету речей виступає як центральний концентратор повідомлень для безпечного двонаправленого зв'язку з ідентифікатором кожного пристрою між хмарною платформою та будівельним обладнанням та іншими елементами сайту. Центр Інтернету речей швидко може збирати дані для кожного пристрою для отримання конвеєр аналітики даних.
- Azure Stream Analytics — це модуль обробки подій, який перевіряє великі потоки даних із пристроїв та інших джерел даних. Ця служба також підтримує вилучення інформації з потоків даних для визначення шаблонів

та зв'язків. У цьому випадку Stream Analytics приймає та аналізує дані з пристроїв Інтернету речей та зберігає результати у базі даних SQL Azure.

- База даних SQL Azure містить результати проаналізованих даних з пристроїв Інтернету речей та показники, які за допомогою веб-програми на основі Azure можуть переглянути аналітики та користувачі.
- Сховище BLOB-об'єктів зберігає дані образи, зібрані з пристроїв Центру Інтернету речей. Дані зображення можна переглянути за допомогою веб-програми.
- Диспетчер трафіку Azure управляє розподілом користувача трафіку між кінцевими точками служби в різних регіонах Azure.
- Підсистема балансування навантаження розподіляє дані від пристроїв будівельного обладнання через веб-служби на основі віртуальної машини для забезпечення високої доступності.
- Віртуальні машини Azure розгортають веб-служби, які отримують та приймають дані результатів конструювання у базі даних Apache Cassandra.
- Apache Cassandra — розподілена база даних NoSQL, яку використовують для зберігання даних для конструювання для подальшої обробки службою Apache Spark.
- Веб-програми розміщують веб-програму кінцевого користувача, яка може використовуватися для запиту та перегляду вихідних даних та зображень. Користувачі також можуть ініціювати пакетні завдання в Apache Spark через програму.
- Apache Spark на HDInsight підтримує обробку пам'яті для підвищення продуктивності програм великих аналітик даних. У цьому випадку, щоб запустити складні алгоритми над даними, що зберігаються в Apache Cassandra, використовується Spark.

Також можуть використовуватись альтернативні:

- Cosmos DB є альтернативною технологією бази даних NoSQL. Cosmos DB забезпечує підтримку кількох джерел у глобальному масштабі з кількома чітко визначеними рівнями узгодженості відповідно до вимог різних клієнтів. Вона також підтримує API Cassandra.
- Azure Databricks – це високопродуктивна платформа на основі Apache Spark, оптимізована для Azure. Вона інтегрована з Azure, щоб забезпечити налаштування одним клацанням миші, упорядкувати робочі процеси та інтерактивний робочий простір для спільної роботи.
- Data Lake Storage є альтернативою для сховища BLOB-об'єктів. У цьому випадку Data Lake Storage недоступна у цільовому регіоні.
- Веб-програми також можуть використовуватися для розміщення веб-служб для отримання даних результатів конструювання.
- Різні варіанти технологій доступні в режимі реального часу для отримання повідомлень, зберігання даних, обробки потоків даних, зберігання аналітичних даних, а також аналітики та звітності.

2.7 Google IoT Platform

Чим більше рішень IoT впроваджується, тим більше даних потребує гострої аналітики і навіть штучного інтелекту для швидкого отримання відповідної інформації. Така інформація дозволяє компаніям у реальному часі визначити відхилення та технічні поломки. Google Cloud IoT намагається змінити існуючі ситуації, використовуючи машинне навчання на кінцевих пристроях і вводячи такі сервіси, як Cloud IoT Core, Cloud IoT Edge, Edge TPU.[14]

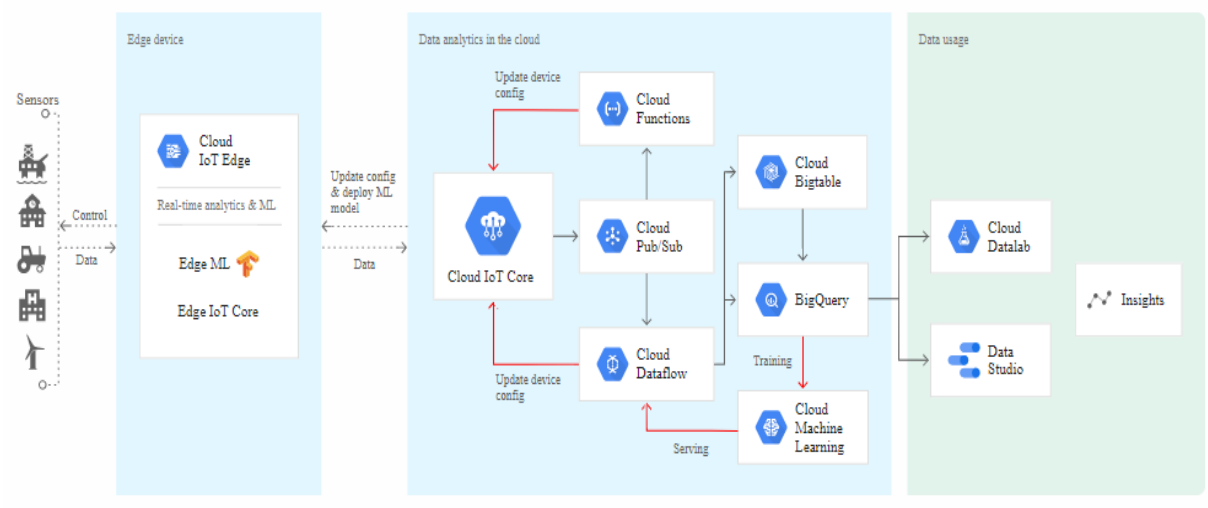

Рис. 2.4. Архітектура Google IoT Platform

Core Cloud IoT

Cloud IoT Core — це повністю керований сервіс, який дозволяє швидко та безпечно підключатися, налаштовувати й отримувати дані з багатьох пристроїв. Використовуючи Cloud Pub/Sub, Core може об'єднувати дані з децентралізованих пристроїв в єдину глобальну систему. У поєднанні з іншими сервісами Google Cloud IoT Core пропонує комплексне рішення для збору, аналізу та візуалізації даних Інтернету речей у режимі реального часу. Це, у свою чергу, дозволяє створювати багатофункціональні моделі, які оптимізують або прогнозують цінні дані для вашого бізнесу.

Cloud IoT Core підтримує звичайні протоколи MQTT і HTTP, що дозволяє клієнтам використовувати багато існуючих пристроїв для створення своїх систем IoT. Ця послуга працює в інфраструктурі Google, яка автоматично масштабується в режимі реального часу, забезпечуючи при необхідності підключення мільйона пристроїв.

Ядро включає в себе:

· Диспетчер пристроїв

Налаштуйте та керуйте окремими пристроями за допомогою консолі або програмного забезпечення. Використовуйте механізми ідентифікації та аутентифікації пристрою під час підключення. Ви також можете налаштувати логіку кожного пристрою та дистанційно керувати ним за допомогою хмари.

· Протокольний міст

Кінцеві точки підключення для балансування навантаження компонентів, вбудована підтримка MQTT і HTTP, збереження всіх телеметрії в Pub/Sub — найкращі функції, які він надає.

Clout IoT Core збирає дані, опубліковані в Cloud Pub/Sub, для подальшої аналітики. Використовуйте Google BigQuery для виконання конкретного аналізу або Cloud Machine Learning Engine для застосування машинного навчання. Після цього ви можете візуалізувати результати за допомогою різноманітних звітів та інформаційних панелей у Google Data Studio.

Крім того, щоб спростити та прискорити підключення різних пристроїв до Core, ви можете використовувати додаткову службу надання послуг Google IoT. Це забезпечує адаптацію як клієнтів, так і виробників. Ця служба використовує надійний апаратний рівень безпеки (він же крипточіп або Secure Element (SE)) для безпечного забезпечення пристроїв. Це дозволяє надавати мільйони пристроїв правильному ядру без втручання людини.

Cloud IoT Edge

Корисний інструмент для розширення обробки даних Google Cloud та використання машинного навчання на периферійних пристроях, таких як автоматичні конвеєрні лінії, турбомашини тощо. Завдяки цьому пристрої можуть працювати в режимі реального часу, використовуючи дані своїх датчиків, і прогнозувати результати локально. Edge може працювати в операційних системах на базі Linux. Він складається з двох компонентів під час

виконання (Connect і ML) і спеціального апаратного прискорювача (TPU) на чіпі ASIC.

- Edge Connect — для безпечного підключення граничних пристроїв до хмари, оновлення програмного забезпечення та мікропрограм, керування передачею даних Core.
- Edge ML — для визначення на пристрої моделі машинного навчання TensorFlow Lite. Переваги: підвищення потужності обробки, низька затримка, універсальність тощо.
- Edge TPU — AI-чіп для роботи з моделями TensorFlow Lite на краю. Високоефективний, оптимізований і має різноманітне використання. Це може бути величезним плюсом при використанні з Cloud TPU.

Переваги:

1. Локальна обробка зображень, відео, жестів, звуків і рухів. Ефективніше, ніж надсилання необроблених даних у хмару.
2. Усунення необхідності надсилати великі обсяги — потенційно конфіденційних — даних.
3. Аутентифікація граничних пристроїв за допомогою веб-токена JSON. Більш ефективна, ніж взаємна аутентифікація стека TLS.

Ще кілька переваг Edge TPU

Для моделей машинного навчання TensorFlow Lite новий чіп Edge TPU є обов’язковим. Він створений спеціально для:

- Швидшої підготовки ML у хмарі.
- Швидшої аналітики ML на периферійних пристроях.
- Дозволяє датчикам приймати локальні розумні рішення в реальному часі.

- Запуск точного штучного інтелекту з високою ефективністю.

У зв’язку зі стрімким зростанням потреби у взаємопов’язаних пристроях, які потребують конфіденційності, низького часу перебування та високої ємності, запуск моделей штучного інтелекту на периферійних пристроях стає звичайним явищем. Edge TPU ідеально підходить для цього, оскільки забезпечує високу ефективність при низькому споживанні фізичних ресурсів та енергії.

2.8 Варіанти застосування аналітичних платформ

Охорона здоров'я

Сектор охорони здоров'я є одним із найшвидших секторів, які впроваджують Інтернет речей. Існує багато датчиків, які входять до сценарію, і сектор охорони здоров’я приймає його дуже швидко. Ці пристрої дозволяють лікарю відстежувати своїх пацієнтів, чи застрягли вони на лікуванні чи ні.

Розумні перевезення

Тримайте транспортні засоби в дорозі, передбачаючи потребу в технічному обслуговуванні та оптимізуючи логістику, використовуючи дані та сповіщення в режимі реального часу, щоб оптимізувати маршрути доставки, контролювати продуктивність та швидко реагувати на затримки чи проблеми, коли вони трапляються. Тому ми можемо зменшити затори на дорогах за допомогою аналітики в реальному часі.

Прогнозне обслуговування обладнання використовується для управління енергією, прогнозування відмови обладнання або виявлення інших проблем. Отже, IoT-аналітика в роздрібній торгівлі допомагає забезпечити точне управління запасами і, що найважливіше, покращує досвід покупок у споживача.

Розумні будівлі

Безсумнівно, IoT допомагає підключатися, зменшувати витрати на управління та комунальні послуги, вивчаючи зібрані вами дані. Наприклад, IoT-

аналітика для розумних будівель допомагає персоналізувати та автоматизувати опалення та охолодження вашої будівлі, використання приміщення, що також допомагає створити більш комфортне та продуктивне робоче середовище.

Розумне сільське господарство

Фермери можуть використовувати значущі дані з даних Інтернету речей для визначення вологості ґрунту та поживних речовин, контролю використання води для росту рослин, щоб отримати кращі результати.

Для промисловості та виробництва:

Прогнозне обслуговування: для виробників і подібних, отримання даних із датчика та створення моделі навколо нього, щоб краще передбачити, коли обладнання може потребувати ремонту, може бути надзвичайно корисним. Виробник ліфтів ThyseenKrupp дотримується цієї методології і виявив, що вона не тільки запобігає простою, але й допомагає технікам швидше знайти правильну причину проблеми.

Оптимізація:

Процес: за допомогою розумних інструментів і компонентів підприємства можуть збирати дані про використання, щоб виявити сильні та слабкі сторони та відповідно внести необхідні правки.

Якість: тестування продуктів у капіталовкладеннях важкої промисловості є дорогим. Випробування вимагають відправки виробничих зразків до лабораторій, що є повільним і ручним процесом. Натомість компанії можуть використовувати дані датчиків, щоб передбачити проблеми з якістю та визначити правильний набір вхідних даних для оптимізації якості.

Безпека промислової інфраструктури: візьмемо, наприклад, склад, який залишається під наглядом протягом ночі. Датчики IoT на детекторах руху можуть «дізнатися», скільки активності вважається «подією», і попередити

людей-операторів, якщо щось станеться, що перевищує цей поріг. З часом, оскільки збирається все більше даних, ефективність виявлення аномалій на основі датчиків Інтернету речей буде тільки покращуватися. Це пояснюється тим, що машинне навчання розвивається краще з даними без необхідності втручання людини для встановлення складних правил щодо того, що вважати подією.

Для маркетингу та продажів:

Соціальна аналітика: поєднуючи дані датчиків, соціальних мереж і відео, менеджери подій можуть покращити досвід учасників на основі тонких і швидких змін таких речей, як рухи обличчя та мова тіла. Датчики IoT дають змогу працювати за допомогою форми аналітики, відомої як «аналіз настроїв», яка підтримується камерами як багатими джерелами даних, у поєднанні з біометричними датчиками ключових учасників подібних заходів, наприклад, тренерів у разі прямих спортивних змагань.

Для споживчих товарів:

Потокова аналітика: для безперервної обробки даних необхідний безперервний збір даних. Цей тип аналізу стане все більш поширеним, коли мова заходить про такі заходи, як самокерування автомобілів, які повинні бути в змозі миттєво реагувати, коли щось змінюється.

Використання споживчих товарів: сьогодні багато продуктів підключені та надають виробникам дані про те, як вони використовуються. Таке розуміння того, що відбувається з продуктами після того, як вони прибувають до кінцевого пункту призначення, може допомогти підприємствам створювати продукти, які є кориснішими, ефективнішими та, зрештою, краще продаються. Це також може допомогти з маркетингом і продажами в поєднанні з даними, зібраними про демографічні дані та інформацію про покупців і аудиторію.

Висновки

В даному розділі було розглянуто принцип роботи аналітичних платформ для Інтернету речей. Визначено ключові функції, які повинні виконуватись аналітичними платформами:

- розширена звітність,
- надійна візуалізація даних,
- фільтрація та сортування,
- трансформація,
- інтеграція джерел даних,
- контрольований обмін,
- мобільний та веб доступ.

В процесі дослідження були встановлені переваги використання аналітичних платформ:

- ефективне управління даними,
- масштабованість,
- дієві рішення,
- краще представлення даних,
- підвищена лояльність клієнтів.

Переваги аналітичної платформи можуть значно підвищити бізнес-можливості компаній та надати вичерпний огляд ефективності. Однак успіх аналітичної платформи значною мірою залежить від її функціональності.

Також було проаналізовано три аналітичні IoT платформи від компаній світового рівня, а саме AWS IoT Analytics, Microsoft Azure IoT, Google IoT Platform.

РОЗДІЛ 3
ПОБУДОВА БЕЗСЕРВЕРНОГО ПОТОКУ ДАНИХ

В даному розділі буде проведено моделювання роботи аналітичної платформи для Інтернету речей на базі сервісу «Google Cloud Platform», а саме ми побудуємо потік даних про погоду, який починається з пристрою Інтернету речей, використовує чергу повідомлень для отримання та доставки даних, використовує безсерверну функцію для переміщення даних у сховище даних, а потім відображає інформацію. Під час виконання будуть використані такі хмарні служби як Google Pub/Sub, Google Cloud Function, Google BigQuery.

BigQuery — це безсерверне, масштабоване, недороге сховище даних підприємства, яке стане ідеальним варіантом для зберігання даних, які передаються з пристроїв IoT, а також дозволить аналітичній панелі запитувати інформацію. Створюємо таблицю, яка міститиме всі погодні дані IoT.

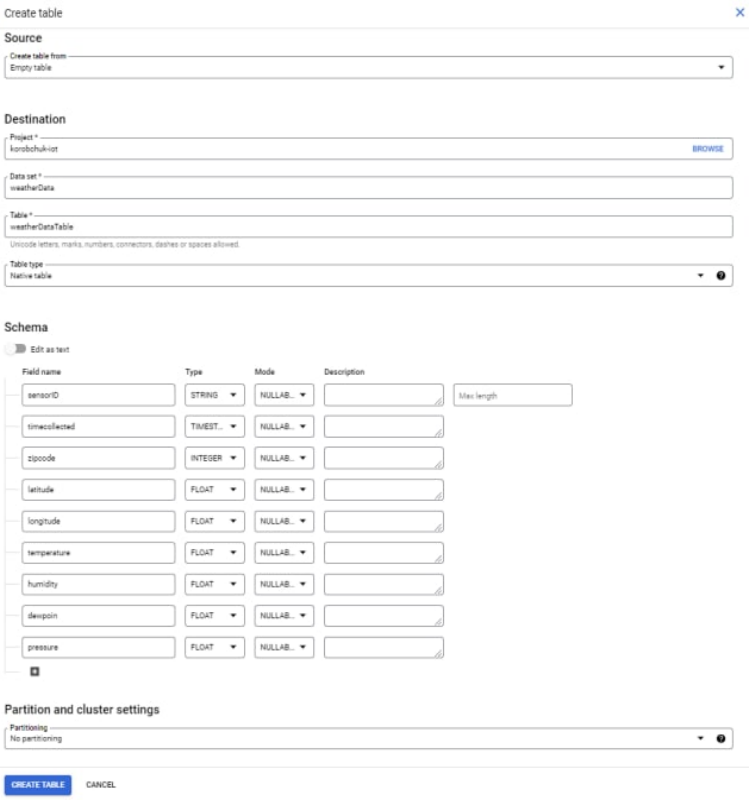

Рис. 3.1. Створення таблиці даних про погоду

Cloud Pub/Sub — це проста, надійна, масштабована основа для потокової аналітики та обчислювальних систем, керованих подіями. Як результат, він ідеально підходить для обробки вхідних повідомлень Інтернету речей, а потім дозволить низхідним системам обробляти їх. Для цього створили тему в Cloud Pub/Sub, щоб було як публікувати повідомлення IoT, так і дозволяти іншим процесам отримати доступ до цих повідомлень.

Хмарні обчислення зробили можливими повністю безсерверні моделі обчислень, де логіка може бути створена на вимогу у відповідь на події, які відбуваються звідусіль. В нашому випадку хмарна функція запускатиметься щоразу, коли публікується повідомлення на тему погоди, читатиме повідомлення, а потім зберігатиме його в BigQuery.

Для цього на Cloud Console обираємо «Cloud Functions». У полі Name вводимо function-weatherPubSubToBQ. Для Trigger обираємо тему Cloud Pub/Sub і з випадаючого списку обираємо дані про погоду. Для вихідного коду виберіть вбудований редактор.

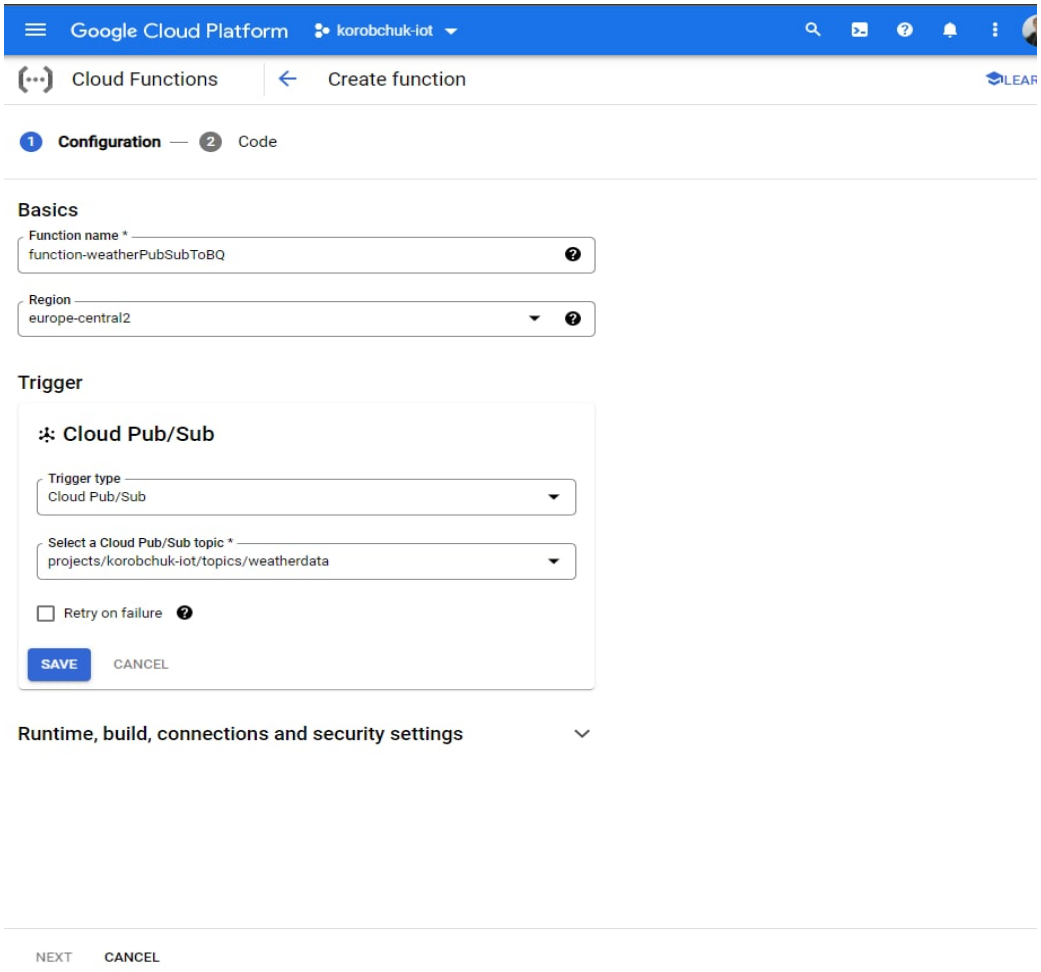

Рис. 3.2. Процес створення Cloud Functions

Далі переходимо до написання програмного коду самої функції. Дана функція буде отримувати дані у форматі JSON, далі проводить розпарсинг даних та занесення даних в таблицю, яку ми створили на початку роботи.

На вкладці index.js вставляємо заздалегідь написаний програмний код, який буде виконувати основну логіку платформи, поверх того, що вказаний за замовченням.

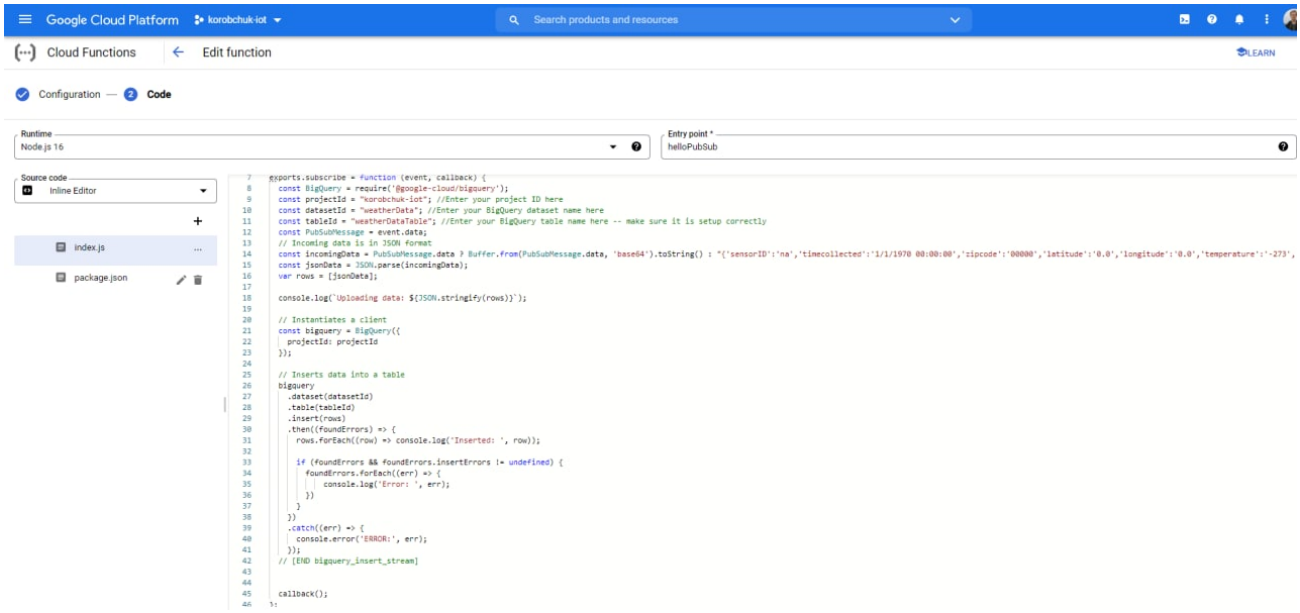

Рис. 3.3. Програмний код функції в Cloud Functions

На вкладці package.json вказуємо конфігурацію необхідних інструментів.

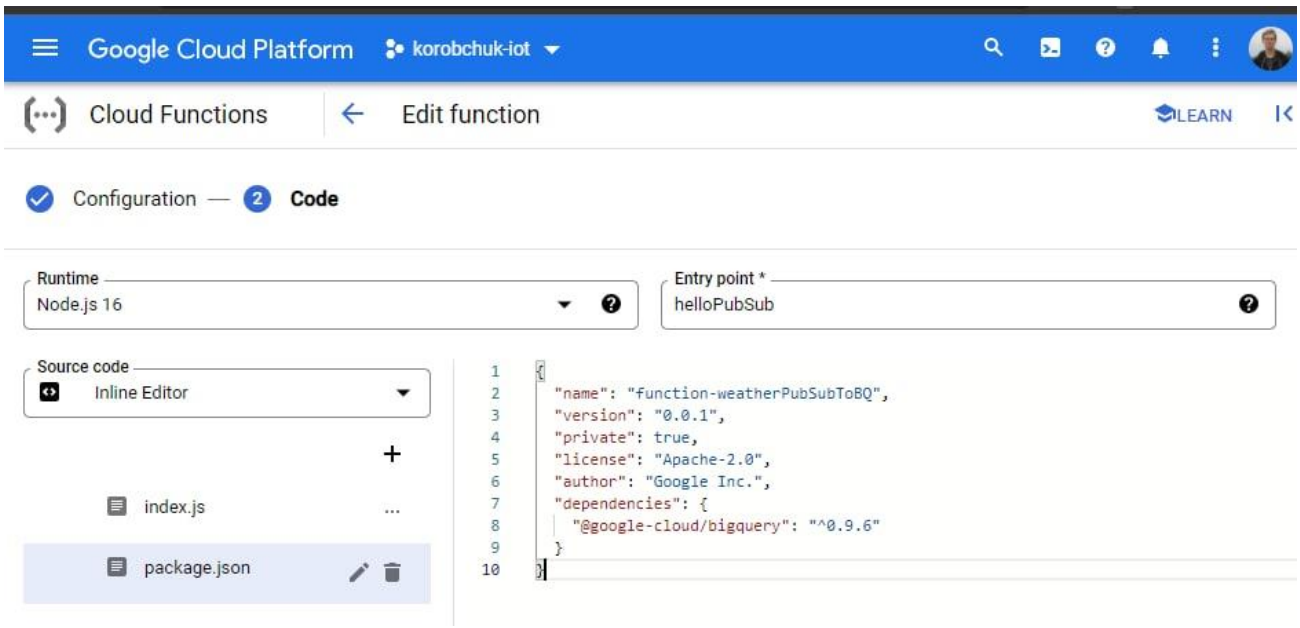

Рис. 3.4. Встановлення пакетів конфігурації

Зберігаємо та запускаємо процес розгортання. Якщо функція була написана правильно, то впродовж 2 хв ми побачимо позитивний статус.

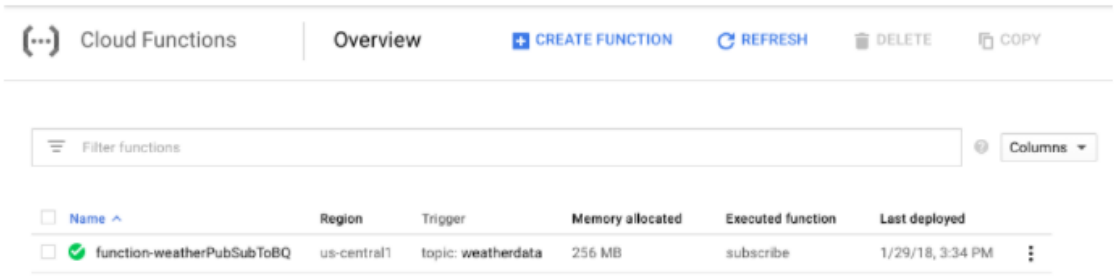

Рис. 3.5. Статус функції в Cloud Functions

Наступний кроком імітуємо потокову передачу даних, використовуючи загальнодоступний набір даних, який зберігається в Google Cloud Storage, і подаємо його в існуючу тему Pub/Sub. Буде використовуватися Google Dataflow разом із наданим Google шаблоном для читання з Cloud Storage та публікації в Pub/Sub. У рамках процесу Dataflow знадобиться тимчасове місце зберігання, тому створюємо для цієї мети місце для зберігання.

У Cloud Console знаходимо обираємо Storage, а потім Browser. При створені корзини вказуємо бажане ім’я, а решту параметрів, в нашому випадку, залишаємо без змін. Тепер необхідно заповнити деталі роботи, як показано нижче.

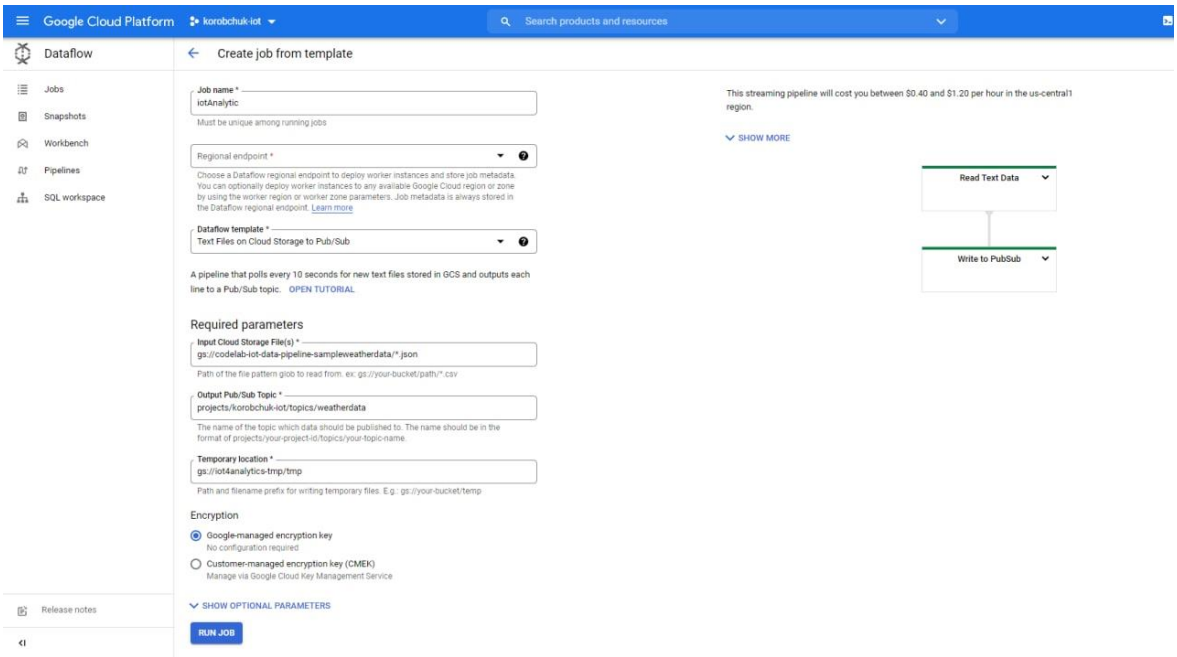

Рис. 3.6. Параметри деталей роботи

Вказані параметри:

- Назву завдання dataflow-gcs-to-pubsub
- Регіон має автоматично обиратись відповідно до місця розміщення вашого проекту, і його не потрібно змінювати.
- Шаблон хмарного потоку даних Text files to Cloud Storage Pub/Sub
- Для вхідних файлів хмарного сховища вводимо gs://codelab-iot-data-pipeline-sampleweatherdata/*.json (це загальнодоступний набір даних)
- Для вихідної публікації/підтеми точний шлях залежатиме від назви проекту і виглядатиме приблизно як "projects/korobchuk-iot /topics/weatherdata"
- Необхідно встановити для тимчасового розташування назву щойно створеного сегмента Google Cloud Storage разом із префіксом назви файлу «tmp» - "gs://iot4analytics-tmp/tmp".

Після встановлення всіх параметрів запускаємо. І маємо знову отримати позитивний статус, а саме «Running».

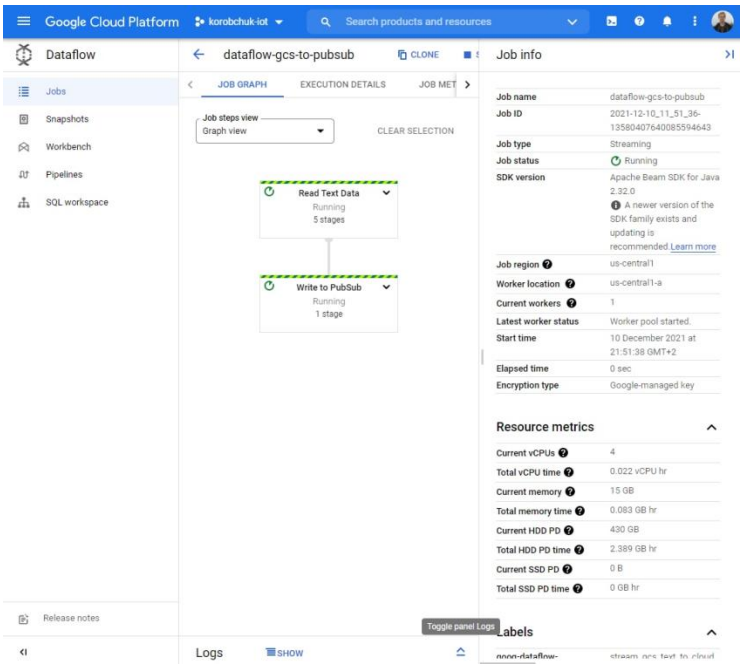

Рис. 3.7. Статус роботи Dataflow

Переконаємося, що все працює вірно та дані надходять в таблицю BigQuery.

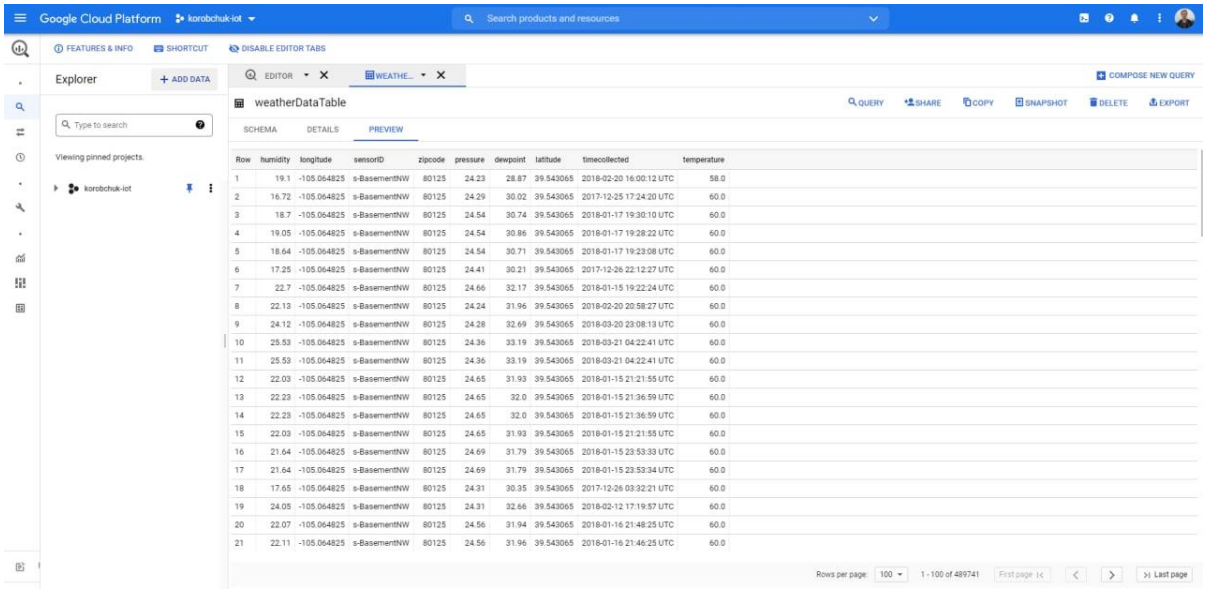

Рис. 3.8. Заповнена таблиця BigQuery з опрацьованими даними

Як ми бачимо з рисунка, дані в таблиці розсортовані та можуть бути використанні в подальшому за потребою. В нашому випадку, ми створимо панель аналітики отриманих даних за допомогою «Data Studio» та відобразимо зміну показників температури, тиску, вологості та точки роси в залежності від дати з отриманих параметрів графічно.

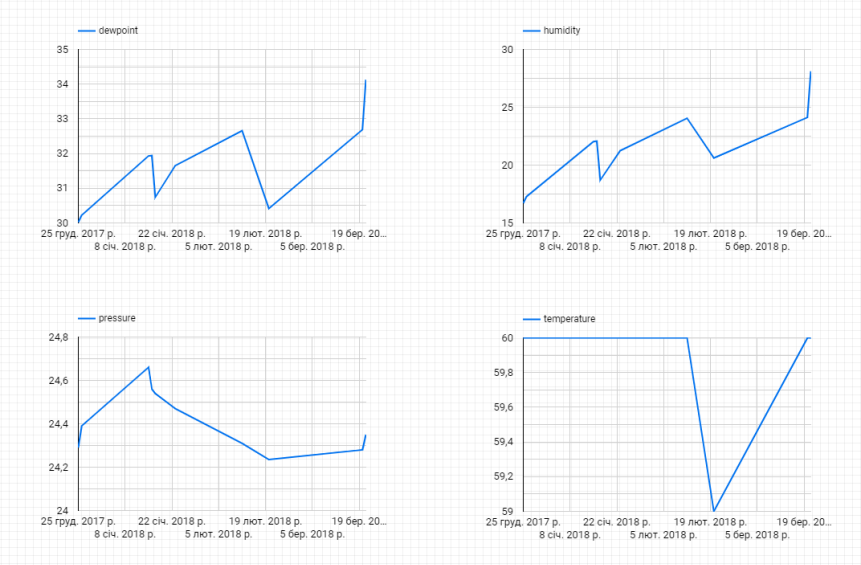

Рис. 3.9. Графічне зображення отриманих результатів

Висновки

В даному розділі було проведено моделювання архітектурного шаблону аналітичної платформи для Інтернету речей за допомогою сервісу «Google Cloud Platform». Було самостійно налаштоване середовище для роботи, а саме було розроблено систему, яка за допомогою таких хмарних рішень як Google Pub/Sub, Google Cloud Function, Google BigQuery приймала дані з пристроїв IoT, обробляла дані та записувала в таблицю. За даними з таблиці було побудовано панель аналітики для графічного відображення результатів роботи.

РОЗДІЛ 4
РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

В даному розділі викладено аналіз перспектив впровадження розумної парковки з використанням аналітичних платформ для Інтернету речей в міській інфраструктурі.

Опис ідеї продукту

Проект, запропонований як варіант із вирішення проблем з заторами на вулицях міст «мільйонників» та розвантаження центральних офісних частин міста від хаотичного запарковування за рахунок впровадження використання аналітичної платформи Інтернету речей в міську інфраструктуру, а саме в сферу паркування.

Таблиця 4.1

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розумна парковка з використання аналітичної платформи для Інтернету речей	1. Гнучка система тарифікації за оплату послуг. 2. Впровадження нових методів моніторингу та керування паркуванням.	Комунальні та приватні підприємства зможуть збільшити попит на малопопулярні паркомісця, а також надавати актуальну інформації з приводу вільних місць на популярних паркінгах в години пік.

При порівнянні з конкурентами в першу чергу увага надається аналітиці даних щодо вільних паркомісць в реальному часі та за результатами аналітики динамічній зміні оплати за користування даною послугою. Порівняння з конкурентами, а також визначення переваг і недоліків наведено у наступні таблиці.

Таблиця 4.2

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Київ цифровий			
1	Використання безкоштовного програмного забезпечення	так	так		так	
2	Непередбачуване зростання вартості	ні	ні			так
3	Індивідуальні можливості кастомізації системи	так	ні			так
4	Максимальне покриття функціоналом доступних послуг	так	ні			
5	Єдина підтримка апаратної та програмної частини	так	так		так	

Конкурент володіє лише частковим покриттям функціоналу. В запропонованому проекті буде проводитись аналіз використання паркувальних місць в реальному часі та відповідно до навантаження збільшувати або зменшувати ціну за оплату використання паркувального місця, що в свою чергу розширює можливості для замовників та є однією з переваг.

Висновки

В даному розділі було запропоновано стартап проект в сфері паркування, який оснований на використанні аналітичних платформ для Інтернету речей. Було проведено порівняння з вже існуючим сервісом «Київ цифровий». Перевагами даного стартапу є розширення покриття послуг, а саме аналітика використання паркомісць в реальному часі. Ця послуга дає можливість динамічно змінювати ціну в залежності від навантаження на паркувальний майданчик.

ЗАГАЛЬНИЙ ВИСНОВОК ПО РОБОТІ

У ході виконання даної магістерської дисертації був розглянутий метод побудови аналітичних платформ для Інтернету речей. Поставлені задачі були виконані.

У першому розділі було розглянуто стандартну архітектуру Інтернету речей, взаємодія IoT з перспективними інфокомунікаційними технологіями та використовувані протоколи.

У другому розділі було досліджено метод побудови аналітичних платформ для Інтернету речей. Визначено ключові функції, які повинні виконувати платформи. Проведено аналіз запропонованих аналітичних платформ від компанії Amazon, Google і Microsoft. Також було наведено сфери та варіанти застосування аналітичних платформ.

У третьому розділі було проведено моделювання важливого архітектурного шаблону, який може обробляти великі обсяги зберігаючи доступність, для побудови безсерверного потоку даних. Для цього було самостійно налаштоване середовище для роботи.

У четвертому розділі був описаний стартап проект розумної парковки з використанням аналітичної платформи для Інтернету речей. Проведено порівняння з існуючими рішеннями, опис переваг у порівнянні з аналогами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. That ‘Internet of Things’ Thing [Електронний ресурс] – Режим доступу до ресурсу: https://www.rfidjournal.com/that-internet-of-things-thing

2. Інтернет речей (IoT): мета застосування та правові проблеми [Електронний ресурс] – Режим доступу до ресурсу: http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?I21DBN=LINK&P21DBN=UJRN&Z21ID=&S21REF=10&S21CNR=20&S21STN=1&S21FMT=ASP_meta&C21COM=S&2_S21P03=FILA=&2_S21STR=Infpr_2018_2_5

3. Росляков А., Ваняшин С., Гребешков А., Самсонов М. «Интернет вещей» – Самара: ПГУТИ, ООО «Издательство Ас Гард», 2014

4. What is IoT architecture? [Електронний ресурс] – Режим доступу до ресурсу: https://www.avsystem.com/blog/what-is-iot-architecture/

5. internet of things (IoT) [Електронний ресурс] – Режим доступу до ресурсу: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

6. IoT Standards and Protocols Guide — Protocols of the Internet of Things [Електронний ресурс] – Режим доступу до ресурсу: https://www.avsystem.com/blog/iot-protocols-and-standards/

7. COAP — ОБЛЕГЧЁННЫЙ HTTP ДЛЯ ИНТЕРНЕТА ВЕЩЕЙ [Електронний ресурс] – Режим доступу до ресурсу: https://r-iot.org/2018/05/14/coap-%d0%b4%d0%bb%d1%8f-%d1%82%d0%b5%d1%85-%d1%83-%d0%ba%d0%be%d0%b3%d0%be-%d0%bc%d0%b0%d0%bb%d0%be-%d1%80%d0%b5%d1%81%d1%83%d1%80%d1%81%d0%be%d0%b2/

8. LwM2M — Lightweight M2M Standard — Protocol and its Benefits [Електронний ресурс] – Режим доступу до ресурсу: https://www.avsystem.com/blog/lightweight-m2m-lwm2m-overview/

9. Understanding IoT Analytics Platform for Real-Time Data Ingestion [Електронний ресурс] – Режим доступу до ресурсу: https://www.xenonstack.com/blog/iot-analytics-platform/

10. How to Build an Analytics Software: Comprehensive Guide 2021 [Електронний ресурс] – Режим доступу до ресурсу: https://www.aimprosoft.com/blog/building-a-data-analytics-software/

11. Top IoT Data Analytics Platforms [Електронний ресурс] – Режим доступу до ресурсу: https://www.iotworldtoday.com/2019/07/10/top-iot-data-analytics-platforms/

12. What is AWS IoT Analytics? [Електронний ресурс] – Режим доступу до ресурсу: https://docs.aws.amazon.com/iotanalytics/latest/userguide/welcome.html

13. Интернет вещей и аналитика данных [Електронний ресурс] – Режим доступу до ресурсу: https://docs.microsoft.com/ru-ru/azure/architecture/example-scenario/data/big-data-with-iot

14. Google IoT Platform: awesome tools for any project [Електронний ресурс] – Режим доступу до ресурсу: https://indeema.com/blog/google-cloud

15. Коробчук О.М. Дипломна робота “Аналіз архітектурних концепцій Інтернету речей для розумних міст ”, 2020 – 66 с.