

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Інститут телекомунікаційних систем  
(повна назва інституту/факультету)

Кафедра телекомунікацій  
(повна назва кафедри)

«На правах рукопису»  
УДК \_\_\_\_\_

До захисту допущено  
Завідувач кафедри

\_\_\_\_\_ Сергій КРАВЧУК  
(підпис) (Ім'я, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2020\_р.

**Магістерська дисертація**  
**на здобуття освітнього ступеня «магістр»**

Спеціальність 172 Телекомунікації та радіотехніка,  
(код і назва)

За освітньо-професійною програмою Інженерія та програмування інфокомунікацій.

на тему: Аналіз архітектурних концепцій Інтернету Речей для розумних будівель.

Виконала: студентка \_2\_ курсу, групи ТЗ – 91 мп  
(шифр групи)

Носкова Дар'я Сергіївна  
(прізвище, ім'я, по батькові) \_\_\_\_\_  
(підпис)

Науковий керівник проф., д.т.н. Ільченко М.Ю.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) \_\_\_\_\_  
(підпис)

Консультант 1-5 доц., к.т.н. Міночкін Д.А.  
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) \_\_\_\_\_  
(підпис)

Рецензент професор кафедри ІТМ, д.т.н., с.н.с. Скулиш М.А.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) \_\_\_\_\_  
(підпис)

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

Студентка \_\_\_\_\_  
(підпис)

Київ – 2020 р.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем

( повна назва )

Кафедра телекомунікацій

( повна назва )

Спеціальність 172 Телекомунікації та радіотехніка

(код і назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою Інженерія та програмування інфокомунікацій.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Явіся В.С.

(підпис)

(ініціали, прізвище)

«20» січня 2020 р.

### ЗАВДАННЯ

на магістерську дисертацію студентки

Носковій Дар'ї Сергіївни

(прізвище, ім'я, по батькові)

1. Тема дисертації: Аналіз архітектурних концепцій Інтернету Речей для розумних будівель

науковий керівник дисертації Ільченко Михайло Юхимович, професор, доктор технічних наук,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «03» «листопада» 2020 р. № 3208-с

2. Строк подання студентом дисертації 11.12.2020

3. Об'єкт дослідження: архітектурна концепція Інтернету Речей для розумних будівель

4. Предмет дослідження: архітектура розумної будівлі

5. Перелік завдань, які потрібно розробити:

- Дослідити архітектуру Інтернету Речей;
- Дослідити архітектуру системи «Розумний будинок»;
- Дослідити архітектуру системи Samsung SmartThings;
- Дослідити архітектуру системи Apple HomeKit;

- Дослідити архітектуру системи Xiaomi Smart Home;
- Проаналізовано системи Samsung SmartThings, Apple HomeKit та Xiaomi Smart Home;
- Модифікувати існуючу архітектуру розумної будівлі;
- Розробити стартап проект.

6. Орієнтовний перелік ілюстративного матеріалу: презентація

7. Орієнтовний перелік публікацій:

Noskova D., Minochkin D. (2020), «THE ARCHITECTURAL CONCEPT BONWIN "SMART ROOM"» Чотирнадцята міжнародна науково-технічна конференція " ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ", с. 340-342

8. Консультанти розділів дисертації

| Розділ                                                 | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------------------------------------------------------|-------------------------------------------|----------------|------------------|
|                                                        |                                           | завдання видав | завдання прийняв |
| 1. Архітектура Інтернет речей                          | Міночкін Д.А. доц., к.т.н.                | 02.10.2020     | 11.10.2020       |
| 2. Архітектура системи «Розумний будинок»              |                                           | 02.10.2020     | 18.10.2020       |
| 3. Аналіз архітектур систем «Розумний будинок»         |                                           | 02.10.2020     | 25.10.2020       |
| 4. Модифікація архітектурної концепції розумна будівля |                                           | 02.10.2020     | 14.11.2020       |
| 5. Стартап проект                                      |                                           | 02.10.2020     | 07.12.2020       |

9. Дата видачі завдання 20 січня 2020 р

#### Календарний план

| № з/п | Назва етапів виконання магістерської дисертації | Строк виконання етапів магістерської дисертації | Примітка |
|-------|-------------------------------------------------|-------------------------------------------------|----------|
| 1.    | Пошук матеріалів по темі                        | 07.09.2020 – 25.09.2020                         | Виконано |
| 2.    | Структура магістерської дисертації              | 25.09.2020 – 02.10.2020                         | Виконано |
| 3.    | Написання 1го розділу                           | 02.10.2020 – 11.10.2020                         | Виконано |
| 4.    | Написання 2го розділу                           | 11.10.2020 – 18.10.2020                         | Виконано |
| 5.    | Написання 3го розділу                           | 18.10.2020 – 25.10.2020                         | Виконано |
| 6.    | Написання 4го розділу                           | 26.10.2020 – 14.11.2020                         | Виконано |
| 7.    | Написання стартап-проекту                       | 16.11.2020 – 27.11.2020                         | Виконано |
| 8.    | Перевірка та доповнення                         | 30.11.2020 – 07.12.2020                         | Виконано |

Студент

\_\_\_\_\_  
(підпис)

Д.С. Носкова

(ініціали, прізвище)

Науковий керівник дисертації

\_\_\_\_\_  
(підпис)

М.Ю. Ільченко

(ініціали, прізвище)

## РЕФЕРАТ

Текстова частина дипломної роботи: 81 с., 27 рис., 14 табл., 38 джерел.

Основна мета магістерської дисертації – є модифікування та дослідження архітектурної концепції Інтернету Речей для розумних будинків.

В роботі розглядаються еталонні моделі IoT від MCE-T та від Всесвітнього форуму IoT, рівневу архітектуру Інтернету речей. Також розглядається історія розвитку концепції Розумний будинок, архітектура Smart Building та його компоненти, та архітектура віддаленого управління Розумним будинком. Проаналізовано три системи розумного будинку: Samsung SmartThings, Apple HomeKit, Xiaomi Smart Home.

В роботі представлена модифікована архітектурна концепція розумної будівлі: з яких елементів складається, як працює. Також описано стартап проект.

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, IOT, WI-FI, BLUETOOTH, ZIGBEE, Z-WAVE, SMART HOME, РОЗУМНИЙ БУДИНОК, SMART BUILDING, XIAOMI, APPLE HOMEKIT, SAMSUNG SMARTTHINGS, АРХІТЕКТУРА

## ABSTRACT

The main purpose of the master's dissertation is to modify and research the architectural concept of the Internet of Things for smart buildings.

The work considers the reference IoT models from MCE-T and from the IoT World Forum, the level architecture of the Internet of Things. The history of the Smart Home concept, the Smart Building architecture and its components, and the Smart Home remote control architecture are also discussed. Three smart home systems were analyzed: Samsung SmartThings, Apple HomeKit, Xiaomi Smart Home.

The work presents a modified architectural concept of a smart building: which elements it consists of, how it works. The startup project is also described.

## Зміст

|                                                                |    |
|----------------------------------------------------------------|----|
| ПЕРЕЛІК СКОРОЧЕНЬ.....                                         | 7  |
| ВСТУП.....                                                     | 10 |
| 1. АРХІТЕКТУРА ІНТЕРНЕТ РЕЧЕЙ .....                            | 12 |
| 1.1. Еталонна модель IoT від МСЕ-Т .....                       | 12 |
| 1.2. Еталонна модель Всесвітнього форуму IoT.....              | 20 |
| 1.3. Рівнева IoT архітектура.....                              | 26 |
| 1.4. Висновки до першого розділу .....                         | 35 |
| 2. АРХІТЕКТУРА СИСТЕМИ «РОЗУМНИЙ БУДИНОК» .....                | 37 |
| 2.1. Історія розвитку концепції «Розумного будинку» .....      | 37 |
| 2.2 Архітектура Smart Building.....                            | 37 |
| 2.3. Базові елементи Smart Home System .....                   | 39 |
| 2.4. Архітектура управління «Розумним будинком» .....          | 40 |
| 2.5. Висновок до другого розділу .....                         | 45 |
| 3. АНАЛІЗ АРХІТЕКТУРИ СИСТЕМ «РОЗУМНИЙ БУДИНОК».....           | 46 |
| 3.1. Samsung SmartThings.....                                  | 46 |
| 3.2. Apple HomeKit .....                                       | 51 |
| 3.3. Xiaomi Smart Home.....                                    | 57 |
| 3.4. Висновок до третього розділу .....                        | 67 |
| 4. МОДИФІКАЦІЯ АРХІТЕКТУРНОЇ КОНЦЕПЦІЇ РОЗУМНА БУДІВЛЯ .....   | 69 |
| 4.1. Архітектура розумної будівлі.....                         | 69 |
| 4.2. Протоколи зв'язку .....                                   | 74 |
| Висновок до 4 розділу .....                                    | 78 |
| 5. СТАРТАП ПРОЕКТ .....                                        | 80 |
| 5.1. Опис ідеї проекту .....                                   | 80 |
| 5.2. Технологічний аудит ідеї проекту.....                     | 82 |
| 5.3. Аналіз ринкових можливостей запуску стартап-проекту ..... | 83 |
| 5.4. Висновок до 5 розділу.....                                | 89 |
| ЗАГАЛЬНІ ВИСНОВКИ .....                                        | 90 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....                                | 91 |

## ПЕРЕЛІК СКОРОЧЕНЬ

|       |                                                                                            |
|-------|--------------------------------------------------------------------------------------------|
| IoT   | Internet of Things (Інтернет речей)                                                        |
| VoIP  | voice over IP (голос через IP)                                                             |
| MCE-T | Міжнародний союз електрозв'язку                                                            |
| RFID  | Radio frequency identification (радіочастотна ідентифікація)                               |
| QR    | Quick Response Code (код швидкого реагування)                                              |
| DSL   | digital subscriber line (Цифрова абонентська лінія)                                        |
| БД    | База даних                                                                                 |
| OSI   | The Open Systems Interconnection model (Базова еталонна модель взаємодії відкритих систем) |
| ПЗ    | Програмне забезпечення                                                                     |
| IWF   | IoT World Forum (Всесвітній форум IoT)                                                     |
| IT    | інформаційні технології                                                                    |
| OT    | Operational Technology (Операційні технології)                                             |
| СУБД  | Система управління базами даних                                                            |
| ETL   | extract, transform, load (отримання, перетворення та збереження)                           |
| M2M   | Machine-to-Machine (Машино-машинна взаємодія)                                              |
| M2P   | Machine-to-People (Людино-машинна взаємодія)                                               |
| C2C   | Consumer-to-Consumer (Від споживача до споживача)                                          |
| LTE   | Long-Term Evolution (Довготерміновий розвиток)                                             |
| PIR   | Polyisocyanurate (Поліізоціанурат)                                                         |
| AI/ML | Artificial intelligence/Machine Learning (Штучний інтелект / Машинне навчання)             |
| BLE   | Bluetooth Low Energy (Bluetooth з низьким енергоспоживанням)                               |
| NFC   | Near field communication (Зв'язок на невеликих відстанях)                                  |
| STB   | Set Top Box (Ресивер цифрового телебачення)                                                |
| RDK-V | Reference Design Kit for video                                                             |
| NAS   | Network Attached Storage (Локальне зберігання файлів)                                      |

|        |                                                                                           |
|--------|-------------------------------------------------------------------------------------------|
| LvM2M  | Level Machine-to-Machine (Рівень машинно-машинної взаємодії)                              |
| ISO    | International Organization for Standardization (Міжнародна організація зі стандартизації) |
| DNS    | Domain Name System (Система доменних імен)                                                |
| COAP   | Constrained Application Protocol (Протокол обмежених додатків)                            |
| DTLS   | Datagram Transport Layer Security (Протокол датаграмм безпеки транспортного рівня)        |
| IP     | Internet Protocol (Інтернет протокол)                                                     |
| AAA    | Authentication, Authorization and Accounting (Аутентифікація, авторизація та облік)       |
| PaaS   | Platform as a service (Платформа як послуга)                                              |
| NoSQL  | not only SQL (не тільки SQL)                                                              |
| SQL    | structured query language (мова структурованих запитів)                                   |
| OLTP   | Online Transactional Processing (Онлайнова обробка транзакцій)                            |
| OLAP   | Online Analytic Processing (Аналітична обробка у реальному часі)                          |
| OS     | Операційна система                                                                        |
| iOS    | iPhone operation system (Операційна система iPhone)                                       |
| UI/UX  | user interface/user experience (користувальницький інтерфейс/досвід взаємодії)            |
| GPS    | Global Positioning System (Система глобального позиціонування)                            |
| DynDNS | Dynamic Domain Name System (Динамічна система доменних імен)                              |
| NO-IP  | Networking over Internet Protocol (Мережа через Інтернет-протокол)                        |
| APNS   | Apple Push Notification Service (Служба push-повідомлень Apple)                           |

|           |                                                                                                                                                                               |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SSL/TLS   | Secure Sockets Layer/Transport Layer Security (рівень захищених сокетів/захист на транспортному рівні)                                                                        |
| IFTTT     | If This then That (Автоматизація взаємодії різних соцмереж, додатків або систем розумного будинку за допомогою простих правил «Якщо трапилося ситуація А, то виконай дію Б».) |
| MFi       | Made For iPhone/iPod/iPad (Створено для iPhone/iPod/iPad)                                                                                                                     |
| РБ        | Розумний Будинок                                                                                                                                                              |
| LAN       | Local area network (Локальна мережа зв'язку)                                                                                                                                  |
| ERP       | Enterprise Resource Planning (Планування ресурсів підприємства)                                                                                                               |
| MES       | Manufacturing Execution System (Система управління виробничими процесами)                                                                                                     |
| WMS       | Warehouse Management System (Система управління складом)                                                                                                                      |
| MQTT      | Message Queuing Telemetry Transport                                                                                                                                           |
| AMQP      | Advanced Message Queuing Protocol                                                                                                                                             |
| DDS       | Data Distribution Service (Служба поширення даних)                                                                                                                            |
| IoT-F2C   | Internet of Things Fog-to-Cloud (Інтернет речей туманно-хмарна взаємодія)                                                                                                     |
| XMPP      | Extensible Messaging and Presence Protocol (Розширюваний протокол обміну повідомленнями та інформацією про присутність)                                                       |
| REST HTTP | Representational State Transfer Hyper Text Transfer Protocol (Передача репрезентативного стану протоколу передачі гіпертексту)                                                |

## ВСТУП

Інтернет речей (Internet of Things, IoT) – отримав свою назву понад 15 років тому. Він потужно увійшов в життя сучасного суспільства, демонструючи вражаючі результати в усе нових і нових сферах проникнення технологій IoT [38].

Функціональні елементи, що складають технологічну структуру IoT, мають надзвичайно малі розміри і гранично мале енергоспоживання, що дозволяє розміщувати їх всередині промислових виробів, окремих товарів, побутових предметів, одягу, взуття, паспортів, кредитних карт, розміщувати на транспортних засобах, тваринах, рослинах, людях тощо [5].

Вважається, що передові технології IoT у самому найближчому майбутньому перетворять індустрію, торгівлю, охорону здоров'я, освіту, військову справу, охорону навколишнього середовища, логістику в самих різних сферах, культуру, міське, сільське і домашнє господарство, побут та багато іншого. При цьому будуть відбуватися не тільки зовнішні зміни, а й змістовні, іноді такі, що змінять сутність тієї чи іншої діяльності [5].

Smart Home – це керування кліматом, вентиляцією, світлом, усіма електричними приборами й системою охорони за допомогою одного смартфона. Датчики, що розташовані в будинку, реагують на будь-які зовнішні зміни, такі як температура повітря, час доби, присутність сторонніх, і дозволяють власнику контролювати будинок, коли він поза його межами; слідкувати за рівнем споживання води та енергії через електронні пристрої, керувати технікою за допомогою голосової системи [4].

Існує безліч архітектурних рішень, але оптимального рішення нема, бо інтернет речей ще розвивається і з'являються все нові і нові технології. Тому IoT системи повинні бути стійкими й повинні розширюватися і розвиватися. Щоб ті рішення, які вже є, і та інфраструктура, яка вже створюється, була актуальна ще 5-10 років [33].

По-перше, вже є дуже багато модифікацій та комплектацій «Розумного будинку». Безліч виробників надають комплекти системи з вбудованими Wi-Fi та Bluetooth передавачами [6].

По-друге, є модифікації, в яких управління відбувається через інтерфейс Ethernet, через сучасні домашні локальні комп'ютерні мережі. До них додатково треба підключати комутатори та Wi-Fi роутери для бездротового управління, якщо вони не вбудовані в контролер. Що стосується ручних вимикачів, то вони також можуть підключатися до центрального контролера бездротовим способом або через електропроводку. Загалом, є різноманітне обладнання для системи розумний будинок, і система складається з таких елементів [3; 6]:

- контролер розумного будинку;
- модулі розширення та зв'язку;
- елементи комутації електричного кола;
- вимірювальні прилади, датчики та сенсори;
- елементи управління системою;
- виконавчі механізми.

При виборі обладнання треба враховувати ще, який протокол передачі даних використовується в системі «Розумний будинок» [6].

Наприклад, найпоширеніший протокол EIB\KNX, передає дані за допомогою комп'ютерних мереж, радіоканалів, силових електричних мереж [6].

Також є протокол X10, який передає дані за допомогою побутової мережу змінного струму 230В. Прилади обмінюються сигналом, поданим у момент переходу змінного струму через нульове значення. Такі сигнали - це радіочастотні імпульси в 120 кГц, тривалістю 1 мс [1; 3].

Тому моя магістерська робота спрямована на дослідження та покращення вже існуючих архітектурних концепцій Інтернету Речей для розумних будівель.

## 1. АРХІТЕКТУРА ІНТЕРНЕТ РЕЧЕЙ

### 1.1. Еталонна модель IoT від MCE-T

Еталонна модель IoT від MCE-T описана в Рекомендації Y.2060. На відміну від більшості інших еталонних моделей і архітектурних моделей, описаних в літературі, модель MCE-T деталізує фактичні фізичні компоненти екосистеми IoT. Це корисно, тому що висвічує елементи екосистеми IoT, які повинні бути з'єднані, інтегровані, керовані і надані додаткам. Детальна специфікація екосистеми описує вимоги до можливостей IoT [2].

Один з важливих аспектів, – це той факт, що IoT на ділі не є мережею фізичних речей. Це перед усім мережа пристроїв, які з'єднані фізичними речами, разом з прикладними платформами – такими як комп'ютери, планшети і смартфони, які взаємодіють з цими пристроями [2].

Нижче наведено список визначень ключових термінів з Рекомендації Y.2060 [7]:

Мережа зв'язку (Communication Network) – інфраструктурна мережа, що з'єднує пристрої та додатки, мережа на основі стека протоколів IP або Інтернет.

Річ (Thing) – предмет фізичного світу (фізичні речі) або інформаційного світу (віртуальні речі), які можуть бути ідентифіковані та інтегровані в мережі зв'язку.

Пристрій (Device) – елемент обладнання, який володіє обов'язковими можливостями зв'язку та додатковими можливостями вимірювання, спрацьовування, а також введення, зберігання та обробки даних.

Пристрій перенесення даних (Data-carrying Device) – пристрій перенесення даних підключається до фізичної речі та непрямым чином з'єднує цю фізичну річ з мережами зв'язку. Прикладами можуть служити активні бирки RFID.

Пристрій збору даних (Data-capturing Device) – під пристроєм збору даних розуміється пристрій, що зчитує або що записує, який має можливість взаємодії з фізичними речами. Взаємодія може здійснюватися непрямым чином за допомогою пристроїв перенесення даних або безпосередньо за допомогою носіїв даних, підключених до фізичних речей.

Носій даних (Data Carrier) – безбатарейний об'єкт перенесення даних, підключений до фізичної речі й має можливість надавати інформацію придатному для цього пристрою збору даних. Ця категорія включає штрих-коди і QR-коди, наклеєні на фізичні речі.

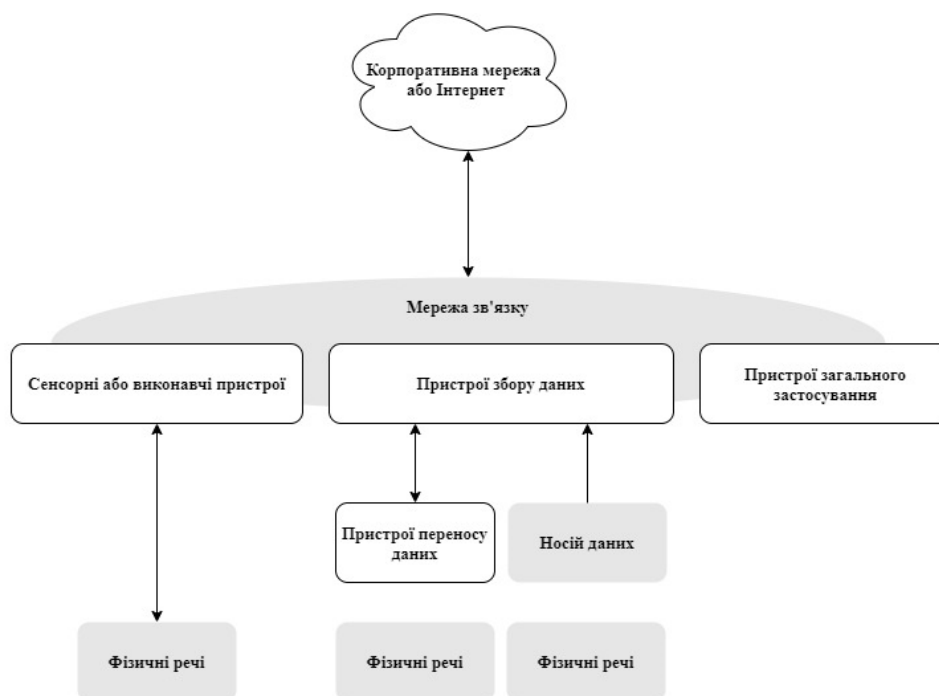
Сенсорний пристрій (Sensing Device) – пристрій, який може виявляти або вимірювати інформацію, що відноситься до навколишнього середовища, і перетворювати її в цифрові електричні сигнали.

Виконавчий пристрій (Actuating Device) – пристрій, який може перетворювати цифрові електричні сигнали, що надходять від інформаційних мереж, в дії.

Пристрій загального призначення (General Device) – пристрій загального призначення володіє вбудованими можливостями обробки і зв'язку і може обмінюватися даними з мережами зв'язку з використанням дротових або бездротових технологій. Пристрої загального призначення включають обладнання та прилади, які стосуються різних галузей застосування IoT, наприклад, верстати, побутові електроприлади і смартфони.

Шлюз (Gateway) – елемент IoT, що з'єднує пристрої з мережами зв'язку. Він виконує необхідну трансляцію між протоколами, що використовуються в мережах зв'язку і в пристроях.

На рис. 1, адаптованому з Рекомендації Y.2060, зображені типи пристроїв в моделі MCE-T. Модель розглядає IoT як мережу пристроїв, тісно пов'язаних з речами [2; 7].



*Рис.1. Типи пристроїв і їх взаємозв'язок з фізичними речами*

Сенсорні та виконавчі пристрої взаємодіють з фізичними речами в навколишньому середовищі. Пристрої збору даних зчитують дані з фізичних речей або записують дані на фізичні речі шляхом взаємодії з пристроями перенесення даних або носіями даних, підключеними або пов'язаними з фізичним об'єктом тим чи іншим чином [2].

Ця модель наводить відмінність між пристроями перенесення даних і носіями даних. Пристрій перенесення даних є пристроєм в сенсі Рекомендації Y.2060. Як мінімум, пристрій завжди має можливості зв'язку і може мати іншими електронними можливостями. Прикладом пристрою перенесення даних є RFID-бирка. Разом з тим носій даних – це елемент, приєднаний до фізичної речі з метою ідентифікації або інформування [2].

В Рекомендації Y.2060 відзначається, що технології, які використовуються для взаємодії між пристроями збору даних і пристроями перенесення даних або носіями даних, включають радіочастотне, інфрачервоне, оптичне і гальванічне збудження [2; 7].

Радіочастотні – радіочастотні ідентифікаційні RFID-бирки, або радіопозначки [2].

Інфрачервоні – інфрачервоні мітки, використовувані в Збройних Силах, лікарнях та інших середовищах, де потрібно відстежувати розташування і переміщення персоналу. Це інфрачервоні нашивки на військовій формі, і працюють від батарейок бейджі, що випромінюють ідентифікаційну інформацію. Останні можуть містити кнопку, при натисканні якої бейдж може використовуватися для проходження через портал, і бейджі, автоматично повторюють сигнал для контролю за переміщеннями персоналу. Пульти дистанційного керування, що використовуються в побуті або в інших середовищах для управління електронними пристроями, теж можна легко інтегрувати в IoT [2].

Оптичні – штрих-коди та QR-коди можуть служити прикладами ідентифікаційних носіїв даних, які зчитуються оптично [2].

Гальванічне збудження – прикладом можуть служити медичні імпланти, які використовують електропровідні властивості людського тіла. В ході комунікації між імплантом і поверхнею гальванічна пара передає сигнали з імпланту на електроди, виведені на шкіру. Ця схема використовує дуже мало енергії, що дозволяє знизити розмір і складність імплантованого пристрою [2].

Останнім типом пристроїв з рис. 1 є пристрої загального призначення. Вони володіють можливостями обробки даних і зв'язку, які можуть бути інтегровані в IoT. Хорошим прикладом є технологія «розумного будинку», яка може інтегрувати практично будь-який пристрій в будинку в мережу для централізованого або дистанційного керування [2].

На рис. 1.1 наведено огляд елементів, залучених в IoT. У лівій частині рисунку наведено різні способи зв'язку з фізичними пристроями. Передбачається, що одна або кілька мереж підтримують зв'язок між пристроями.

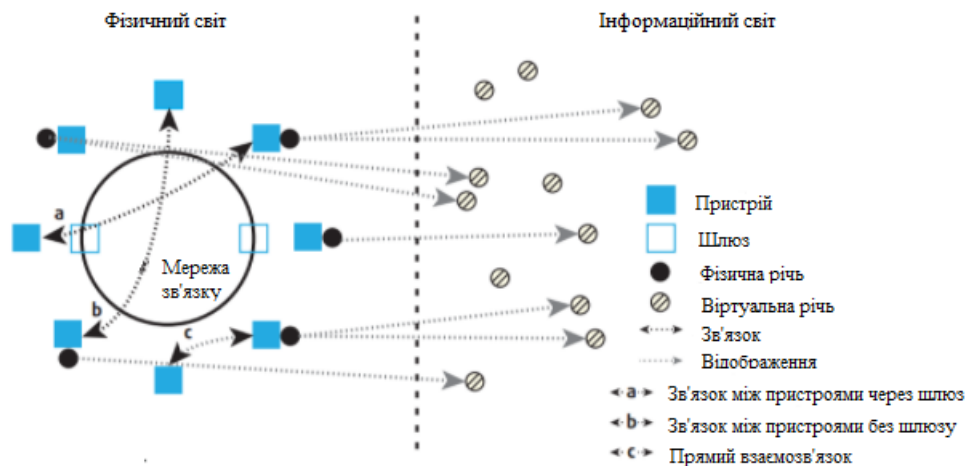


Рис. 1.1. Технічний огляд IoT (Рекомендація Y.2060)

На рис. 1.1 з'являється ще один пристрій, пов'язаний з IoT – шлюз. Шлюз працює транслятором між протоколами. Шлюзи вирішують одну з головних проблем при проектуванні IoT, а саме проблему сумісності, як між різними пристроями, так і між пристроями та Інтернетом або корпоративною мережею. «Розумні» пристрої підтримують широкий спектр бездротових і дротових технологій передачі даних і мережевих протоколів. Крім того, можливості обробки даних у таких пристроїв, як правило, обмежені [2; 34].

Рекомендація Y.2067 закріплює вимоги до шлюзів IoT, які зазвичай розпадаються на три категорії [8]:

- підтримує різні технології доступу до пристроїв, дозволяючи пристроям обмінюватися даними один з одним і з Інтернетом або корпоративною мережею, які містять додатки IoT. Такі схеми доступу можуть, включати ZigBee, Bluetooth, Wi-Fi.
- підтримує необхідні мережеві технології як для локальних, так і для глобальних мереж. Ці технології можуть містити Ethernet і Wi-Fi на території організації, а також стільниковий зв'язок, Ethernet, DSL і кабельний доступ до Інтернету і глобальним корпоративним мережам.
- підтримує взаємодію з додатками, управління мережею і функції безпеки.

Дві перших вимоги охоплюють трансляцію протоколів між різними мережевими технологіями та стеками протоколів. Третя вимога зазвичай

називається функцією IoT-агента. По суті, IoT-агент надає функціональність високого рівня від імені IoT-пристроїв, таку як організація або резюмування даних з декількох пристроїв для передачі в IoT-додатки, забезпечення протоколів і функцій безпеки й взаємодія з системами управління мережею [2].

Слід зазначити, що термін «мережа зв'язку» прямо не визначається в серії IoT-стандартів Y.206x. Мережа (або мережі) зв'язку підтримує зв'язок між пристроями та може безпосередньо підтримувати прикладні платформи. Вона може мати розміри невеликого IoT, такого як домашня мережа «розумних» пристроїв. У більш загальному сенсі мережу (або мережі) пристроїв з'єднується з корпоративними мережами або Інтернетом для зв'язку з системами додатків і серверами, на яких розташовані бази даних, пов'язані з IoT [2].

Тепер можна повернутися до лівої частини рис. 1.1, яка ілюструє можливості зв'язку пристроїв між собою.

Перша можливість – зв'язок між пристроями через шлюз. Наприклад, за допомогою шлюзу сенсорний або виконавчий пристрій з підтримкою Bluetooth може здійснювати зв'язок з пристроєм збору даних або пристроєм загального призначення, що використовують Wi-Fi [2].

Друга можливість – зв'язок через мережу зв'язку без шлюзу. Наприклад, якщо всі пристрої в мережі «розумного будинку» підтримують Bluetooth, вони можуть управлятися з комп'ютера, планшета або смартфона з підтримкою Bluetooth [2].

Третя можливість – прямий зв'язок пристроїв між собою за окремою локальної мережі, в той час як зв'язок із зовнішньою мережею (на рисунку не показана) здійснюється через шлюз LAN. Уявіть собі, що на великій території як на фермі або заводі, знаходиться велика кількість датчиків з низьким енергоспоживанням. Ці пристрої взаємодіють між собою для послідовної передачі даних на пристрій, підключений до шлюзу в мережу зв'язку [2].

У правій частині рис. 1.2 підкреслюється, що кожна фізична річ в IoT може бути представлена в інформаційному світі однією або декількома віртуальними речами, але при цьому віртуальна річ може існувати без відповідної фізичної речі.

Фізичні речі зіставлені віртуальним речам, які зберігаються в БД і інших структурах даних. Додатки обробляють віртуальні речі й працюють з ними.

На рис. 1.2 зображена еталонна модель IoT від МСЕ-T, яка складається з чотирьох рівнів та можливості управління і безпеки, що діють між рівнями. До сих пір ми говорили про рівень пристрою. У термінах функціональності зв'язку рівень пристрою містить фізичний і канальний рівні OSI. Тепер перейдемо до інших рівнів [2].

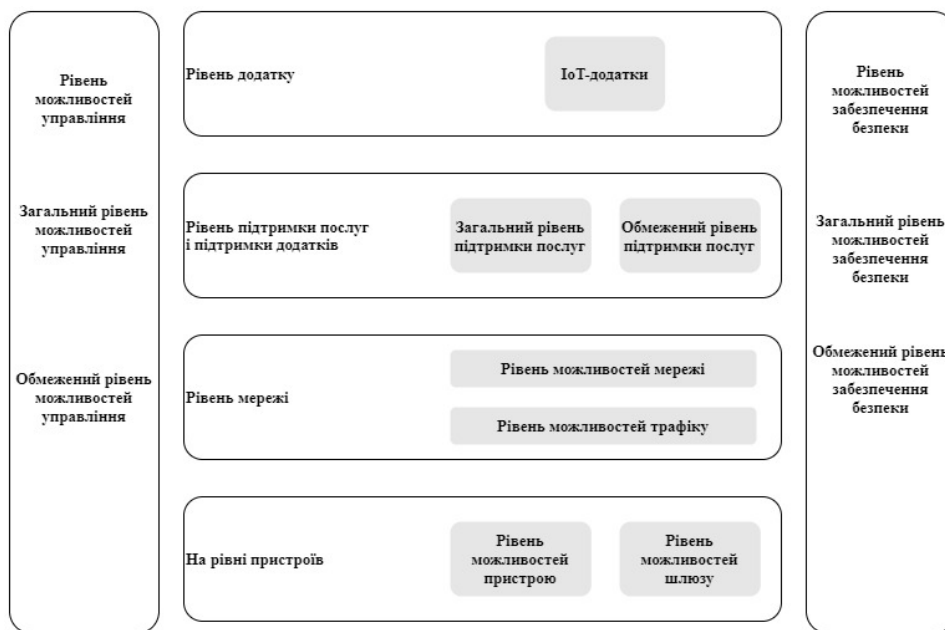


Рис. 1.2. Еталонна модель IoT по Рекомендації Y.2060

Рівень мережі виконує дві базові функції. Можливості мережі відносяться до взаємодії пристроїв і шлюзів. Транспортні можливості відносяться до транспорту інформації служб і додатків IoT, а також інформацією управління і контролю IoT. Ці можливості відповідають мережевому і транспортному рівнів OSI.

Рівень підтримки послуг і підтримки додатків надає можливості, які використовуються додатками. Багато різноманітні додатки можуть використовувати загальні можливості підтримки. До прикладів належать спільне опрацювання даних і управління БД. Спеціалізовані можливості підтримки – це конкретні можливості, які призначені для задоволення потреб конкретної підмножини додатків IoT.

Рівень додатки складається з усіх додатків, які взаємодіють з IoT-пристроями.

Рівень можливостей управління охоплює традиційні функції управління мережею, тобто управління несправностями, конфігурацією, обліком, показниками роботи та безпекою.

В Рекомендації Y.2060 прикладами загальних можливостей управління є [7]:

- управління пристроями – включає виявлення пристроїв, аутентифікацію, дистанційну активацію і деактивацію пристроїв, конфігурацію, діагностику, оновлення прошивки та ПЗ, управління робочим статусом пристрою;
- управління топологією локальної мережі – управління конфігурацією мережі;
- управління трафіком і перевантаженнями – виявлення умов перевантаженості мережі й реалізація резервування ресурсів для термінових або життєво важливих потоків трафіку.

Спеціалізовані можливості управління тісно пов'язані з вимогами додатків. Наприклад, вимогами з контролю лінії передачі електроенергії в «розумній» електромережі.

Рівень можливостей забезпечення безпеки включає загальні можливості забезпечення безпеки, які не залежать від додатків. В Рекомендації Y.2060 прикладами загальних можливостей забезпечення безпеки є [7]:

- на рівні програми – авторизація, аутентифікація, захист конфіденційності та цілісності даних програми, захист недоторканності приватного життя, аудит безпеки й антивірусний захист;
- на рівні мережі – авторизація, аутентифікація, конфіденційність даних про використання та даних сигналізації, а також захист цілісності даних сигналізації;
- на рівні пристрою – аутентифікація, авторизація, перевірка цілісності пристрою, управління доступом, захист конфіденційності та цілісності даних.

Спеціалізовані можливості забезпечення безпеки тісно пов'язані з вимогами додатків. Наприклад, вимогами безпеки мобільних платежів.

## 1.2. Еталонна модель Всесвітнього форуму IoT

Всесвітній форум IoT (IoT World Forum, IWF) – щорічна подія, що об'єднує представників бізнесу, державних структур та вузівської науки з метою просування IoT на ринок. Комітет з архітектури Всесвітнього форуму IoT, складений з лідерів індустрії: IBM, Intel і Cisco, в жовтні 2014 опублікував еталонну модель IoT. Дана модель є загальною структурою, покликаною допомогти галузі прискорити розгортання IoT. Модель призначена для того, щоб стимулювати співпрацю та сприяти створенню повторюваних моделей впровадження [2].

IWF стурбований більш масштабним питанням розробки додатків, проміжного програмного забезпечення і функцій підтримки для корпоративного IoT. На рис. 1.3 зображена семирівнева модель IoT [2].



Рис. 1.3. Еталонна модель Всесвітнього форуму IoT

Документальний опис моделі IWF, опублікований Cisco, вказує, що розроблена модель відрізняється наступними характеристиками [2] :

- спрощує – допомагає розбити складні системи на частини так, щоб кожна з цих частин стала більш зрозумілою;
- прояснює – надає додаткові відомості для точної ідентифікації рівнів IoT і вироблення загальної термінології;
- ідентифікує – ідентифікує аспекти, в яких ті чи інші типи обробки оптимізовані в різних частинах системи;

- стандартизує – є перший крок до того, щоб постачальники могли створювати продукти IoT, здатні взаємодіяти один з одним;
- організовує – робить IoT реальним і доступним, а не просто абстрактною концепцією.

Рівень 1 – фізичні пристрої та контролери, які можуть керувати кількома пристроями. Рівень 1 моделі IWF приблизно відповідає рівню пристрою в моделі MSE-T (рис. 1.3). Як і в моделі MSE-T, елементи на цьому рівні – не фізичні речі, а пристрої, які взаємодіють з фізичними речами, такі як сенсорні і виконавчі пристрої. Серед інших можливостей ці пристрої можуть вміти здійснювати аналого-цифрове і цифро-аналогове перетворення, генерацію даних, а також підтримувати дистанційний опитування або дистанційне керування [2].

З логічної точки зору цей рівень реалізує зв'язок пристроїв між собою і між пристроями і низькорівневою обробкою на рівні 3. З фізичної точки зору цей рівень складається з мережевих пристроїв, таких як маршрутизатори, комутатори, шлюзи і брандмауери, використовуваних для створення локальних і глобальних мереж і підключення до Інтернету. Цей рівень дозволяє пристроям здійснювати зв'язок один з одним і за допомогою більш високих логічних рівнів обмінюватися даними з прикладними платформами, такими як комп'ютери, пристрої дистанційного управління і смартфони [2].

Рівень 2 моделі IWF приблизно відповідає рівню мережі в моделі MSE-T. Основна відмінність в тому, що модель IWF відносить шлюзи до рівня 2, в той час як в моделі MSE-T вони відносяться до розряду 1. Оскільки шлюз є мережевим пристроєм і пристроєм зв'язку, віднесення його до рівня 2 має більше сенсу [2].

У багатьох впроваджуваних системах IoT розподілена мережа датчиків може генерувати великі обсяги даних. Наприклад, офшорні нафтові родовища і нафтопереробні заводи можуть генерувати до терабайта даних щодня. Літак може генерувати кілька терабайт даних на годину. Замість того, щоб зберігати всі ці дані постійно (або хоча б довгий час) в централізованому сховищі, доступному для додатків IoT, часто більш доцільно виконувати якомога більшу частину обробки даних якомога ближче до датчиків [2].

Тому завданням рівня периферійних обчислень (edge computing level) є перетворення мережеских потоків даних в інформацію, придатну для зберігання і більш високорівневої обробки. Елементи обробки на цьому рівні можуть мати справу з великими обсягами даних і виконувати операції перетворення даних, в результаті яких зберігати доводиться вже набагато менший обсяг. Опублікований Cisco документ по моделі IWF містить такі приклади операцій на рівні периферійних обчислень [2]:

- аналіз – аналіз даних по критеріях того, чи підлягають вони обробці на більш високому рівні;
- форматування – переформатування даних для однакової високорівневої обробки;
- розархівування/декодування – обробка криптографічних даних з додатковим контекстом (таким як походження);
- дистиляція/скорочення – скорочення і/або резюмування даних для того, щоб мінімізувати ефект на обсяг даних і трафік в мережі і в високорівневих системах обробки;
- оцінка – визначення того, чи становлять дані порогове значення або аварійний сигнал; цей процес повинен включати перенаправлення даних додатковим одержувачам.

Елементи обробки на цьому рівні відповідають пристроям загального призначення в моделі МСЕ-Т (рис. 1). Як правило, вони розгортаються фізично на краю мережі IoT, тобто поруч з сенсорами і іншими пристроями генерації даних. Таким чином, частина базової обробки великих обсягів генеруються даних знімається з прикладних програм IoT, розташованих центрально [2].

Туманні обчисленнями (Fog Computing) – це обробка на рівні периферійних обчислень. Туманні обчислення та туманні служби будуть відмінною характеристикою IoT. Цей принцип проілюстрований на рис. 1.4 [2].

Туманні обчислення – це тренд в сучасних мережеских технологіях. У хмарних обчисленнях великий обсяг централізованих ресурсів зберігання і обробки

даних, які доступні розподіленим споживачам за допомогою хмарних мережових структур для відносно невеликого числа користувачів [2; 35].

В туманних обчисленнях велике число окремих інтелектуальних об'єктів здійснюють зв'язок з туманними мережевими структурами, які здійснюють обчислення і зберігають ресурси поруч з периферійними пристроями в IoT. Туманні обчислення вирішують проблеми, що виникли внаслідок діяльності тисяч або мільйонів «розумних» пристроїв, включаючи проблеми безпеки, конфіденційності, обмежених можливостей мережі і затримки. Термін «туманні обчислення» обраний тому, що туман стелиться по землі, в той час як хмари знаходяться високо в небі [2].

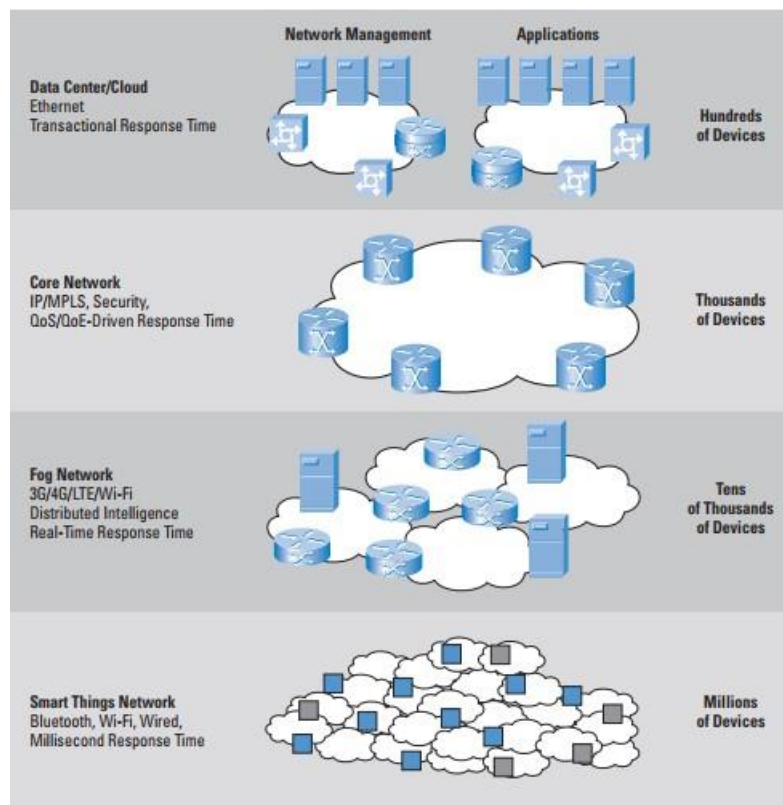


Рис. 1.4. Туманні обчислення

Порівняння хмарних і туманних обчислень наведено в таблиці 1.

Таблиця 1.

*Порівняння хмарних і туманних обчислень*

|                                          | Хмара                                         | Туман                                             |
|------------------------------------------|-----------------------------------------------|---------------------------------------------------|
| Розташування ресурсів зберігання/обробки | Центр                                         | Край                                              |
| Затримка                                 | Від низької до високої                        | Низька                                            |
| Доступ                                   | Фіксований або бездротової                    | Головним чином бездротової                        |
| Підтримка мобільності                    | Не можна застосувати                          | Так                                               |
| Контроль                                 | Централізований/ієрархічний (повний контроль) | Розподілений/ієрархічний (частковий контроль)     |
| Доступ до служб                          | Через ядро                                    | На краю/с надолонника                             |
| Доступність                              | 99,99%                                        | Висока нестабільність/високий рівень резервування |
| Число користувачів/пристроїв             | Десятки і сотні мільйонів                     | Десятки мільярдів                                 |
| Основний генератор контенту              | Люди і пристрої                               | Пристрої/сенсори                                  |
| Генерація контенту                       | У центральному розташуванні                   | Скрізь                                            |
| Споживання контенту                      | На кінцевих пристроях                         | Скрізь                                            |
| Віртуальна програмна інфраструктура      | Центральні корпоративні сервери               | Користувальницькі пристрої                        |

На рівні 4, накопичені дані, які надійшли з різних пристроїв, профільтовані та оброблені рівнем периферійних обчислень, поміщаються в сховище, де будуть доступні для більш високих рівнів. Цей рівень відрізняється і від низькорівневих (туманних), і від високорівневих (хмарних) обчислень за особливостями конструкції, вимогам і методам обробки [2].

Дані, що проходять крізь мережу, називаються «даними в русі». Швидкість і організація даних в русі визначається пристроями, що генерують дані. Генерація даних відбувається по подіям, або періодично, або по виникненні якої-небудь події в середовищі. Для збору даних та їх обробки необхідно реагувати на їх появу в реальному часі.

І навпаки, багатьом додаткам не потрібно обробляти дані зі швидкістю мережевої передачі. На практиці ні хмарна мережа, ні прикладні платформи не змогли б встигати за обсягами даних, що генеруються величезною кількістю IoT-пристроїв. Замість цього додатки мають справу з «даними в спокої», тобто даними в тому чи іншому легкодоступному сховище. Додатки можуть звертатися до даних у міру необхідності або поза режимом реального часу. Таким чином, високі рівні функціонують за принципом транзакцій, в той час як три нижніх рівні працюють по подіях.

Нижче перераховані операції, що виконуються на рівні накопичення даних:

- перетворення «даних в русі» в «дані в спокої»;
- перетворення формату з мережевих пакетів в реляційні таблиці БД;
- перехід від обчислень щодо подій до обчислень за запитом;
- значне зниження обсягу даних за рахунок фільтрації і вибіркового зберігання.

Ще один погляд на рівень накопичення даних полягає в тому, що він являє собою кордон між інформаційними технологіями (ІТ), під якими розуміється цілий спектр технологій обробки інформації, включаючи - ПЗ, обладнання, технології зв'язку і супутні служби, та операційними технологіями (Operational Technology, OT), що представляють собою обладнання і ПЗ, які виявляють або викликають зміни шляхом прямого моніторингу або контролю фізичних пристроїв, процесів і подій на підприємстві [2; 31].

Рівень накопичення даних вбирає велику кількість даних і поміщає їх в сховище, практично не пристосовуючи до потреб конкретних програм або груп додатків. З рівня периферійних обчислень в сховище може надходити безліч різних видів даних в різних форматах і від різномірних оброблювачів. Рівень абстракції

даних може агрегувати і формувати такі дані способами, які роблять доступ додатків більш керованим і ефективним.

Рівень додатки містить додатки будь-якого типу, що використовують дані IoT на вході або керуючі IoT-пристроями. Як правило, додатки взаємодіють з рівнем 5 і з даними в спокої, тому їм не обов'язково функціонувати на швидкостях мережі. Слід передбачити спрощений режим роботи, який дозволить додаткам минути проміжні рівні і безпосередньо взаємодіяти з рівнем 3 або навіть рівнем 2. Модель IWF не визначає додатки по всій строгості, вважаючи цей аспект виходять за рамки дискусії про модель IWT [2].

Рівень взаємодії і процесу з'явився в результаті того, що IoT буде корисний лише тоді, коли з ним зможуть взаємодіяти люди. Цей рівень включає кілька додатків й обмін даними та керуючою інформацією по Інтернету або корпоративної мережі [2].

Модель IWF визначає необхідний функціонал і проблеми, які потрібно вирішити до того, як галузь зможе реалізувати цінність IoT. Дана модель корисна як для постачальників, що розробляють функціональні елементи всередині моделі, так і для замовників, які допомагають їм виробити свої вимоги і оцінювати пропозиції постачальників [2].

### 1.3. Рівнева IoT архітектура

На рис. 1.5. представлена рівнева архітектура IoT рішень [9].

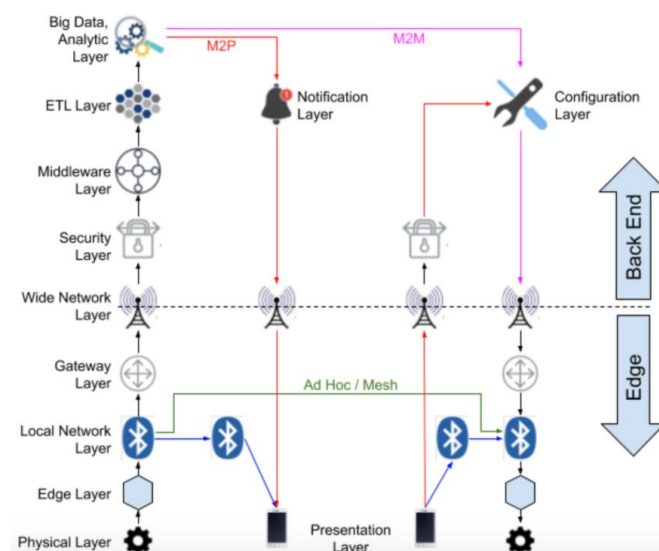


Рис. 1.5. Рівнева архітектура IoT рішень

### 1.3.1. Physical Layer – фізичний рівень

На даному рівні виконуються два типи операцій – збір інформації та здійснення механічної роботи.

Фізичні елементи мають певні загальні вимоги [9]:

- низьку ціну ;
- живлення від батарейки;
- часто розташування в важкодоступних і віддалених місцях з мінімальними витратами на установку і обслуговування;
- первинна обробка зображення з прийняттям рішення на основі штучного інтелекту.

Є дві проблеми для вимог фізичного рівня [9]:

- низький рівень споживання енергії;
- застосування відеокамер.

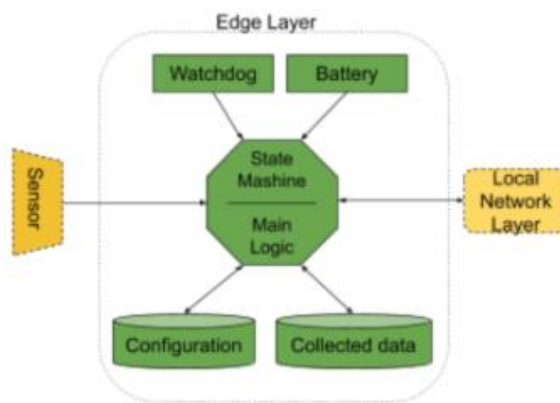
### 1.3.2. Edge Layer – рівень периферійного обчислення

Цей рівень зазвичай підключається до одного датчику або виконавчого механізму. Він забезпечує мінімальну функціональність для перетворення аналогової інформації в цифрову та навпаки. Багато виробників, що випускають ці пристрої, не мають єдиного стандарту для моделі даних, конфігурації та експлуатації [9].

Для зниження енергоспоживання периферійні пристрої мають чотири режими роботи:

- сон;
- вимірювання та збору інформації з датчиків;
- зв'язку, передачі і отримання інформації;
- установка та підключення.

Нижче представлена блок-схема периферійного пристрою.



*Рис. 1.6. Блок-схема периферійного пристрою*

Периферійний пристрій об'єднує три рівні: фізичний, периферійного обчислення і комунікаційний. Основна функціональність рівня периферійного обчислення – локальна ETL (Extract, Transformation and Load) – отримання, перетворення та збереження інформації з датчиків. Даний рівень є відповідальним за фільтрацію перешкод, попередній аналіз і локальне збереження [9].

### 1.3.3. Local Network Layer – рівень периферійної комунікації

Передача даних є найбільш енергоємною частиною периферійного пристрою, тому що більшість периферійних пристроїв не підключені до електромережі й провідним засобам зв'язку. Крім того, периферійні пристрої можуть бути розташовані досить далеко від шлюзу. З іншого боку, кількість переданої інформації зазвичай досить мало. Наступні протоколи використовуються на рівні периферійної комунікації [9]:

- ZigBee/Z-wave;
- BLE;
- Lora;
- Proprietary low band.

Для збільшення відстані і надійності зв'язку на цьому рівні широко використовуються Ad Hoc і Mesh.

Для цілей конфігурації також може використовуватися протокол NFC. У процесі першої установки або технічного обслуговування сервісний інженер з мобільним додатком може підключатися до периферійних пристроїв через рівень

периферійної комунікації. Іноді QR-код, надрукований на периферійному пристрої, також використовується для аутентифікації [9].

#### 1.3.4. Gateway Layer – рівень шлюзу

В IoT-рішенні існує кілька причин наявності рівня шлюзу:

- якщо Backend буде отримувати необроблену інформацію, це збільшить його потужність і витрати будуть дуже великі;
- робота Backend не може гарантувати реакцію в реальному часі для великої кількості периферійних пристроїв;
- через обмеження безпеки деяка інформація не може бути відправлена в Backend і не може постійно контролюватися людиною. До такої інформації відносяться дані камер вуличного спостереження, медична інформація, тощо.

Шлюз повинен забезпечувати основний функціонал:

- здійснювати другий рівень ETL від своїх периферійних пристроїв;
- фіксувати критичну ситуацію і видавати локальну реакцію, навіть без зв'язку з Backend. Це можна порівняти з сигналами серцебиття людини або диханні легенів без участі головного мозку;
- комунікувати з Backend. Відправляє на сервер оброблену інформацію з периферійних пристроїв і отримує дані конфігурації для периферійних пристроїв;
- зберігати інформацію про статус периферійних пристроїв, і дані ними зібрані.

У деяких випадках функціональність AI/ML має бути присутня на рівні шлюзу. Шлюзовий пристрій живиться від електромережі або має велику вбудовану батарею, але в деяких рішеннях також потрібно низьке енергоспоживання [9].

У найближчому майбутньому слід очікувати подальшу інтеграцію всіх трьох компонент: шлюз + STB + безпека в один пристрій.

У ньому також ймовірно будуть присутні сервіси NAS (локального зберігання файлів) і AI/ML для машинного відео/аудіо розпізнавання. Пристрої аудіо розпізнавання, такі як Alexa, базуються на Cloud інфраструктурі, але

найімовірніший первинний розпізнавання буде переноситися на периферійний рівень [9].

#### 1.3.5. Wide Network Layer - рівень зовнішнього зв'язку

Логічний рівень зовнішнього зв'язку має стандартизований протокол для рішень IoT, який називається LwM2M. Протокол LwM2M розроблений для доступу до кожного периферійного пристрою, але оскільки багато постачальників периферійних пристроїв не підтримують інтерфейси LwM2M, шлюзовий пристрій може вирішити цю проблему і створити обгортку для зв'язку з периферійними пристроями [9].

Рівень зовнішнього зв'язку містить комунікаційні. Він включає служби балансування, щоб визначити своє місцезнаходження, засновані на DNS сервісі, транспортний протокол COAP, шифрування DTLS і багато інших компонентів [9].

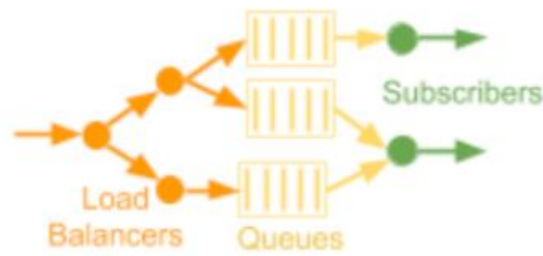
Протокол LwM2M використовує протокол шифрування DTLS. Протокол DTLS – це протокол з ключами безпеки і сесією встановлення з'єднання – handshake. Він працює за схемою точка-точка. Для розшифровки пакетів DTLS використовується той же екземпляр Backend, який був під час сесії з'єднання. Це створює проблему для балансувальника навантаження (Load Balancer), який є частиною рівня безпеки. Load Balancer необхідний для автоматичного масштабування при високому завантаженні системи. Щоб уникнути цього обмеження, служба DNS використовується в якості Load Balancer. Кожні N DNS-запит отримує новий IP-адреса примірника рівня безпеки [9; 36].

#### 1.3.6. Security Layer - рівень безпеки

Цей рівень забезпечує функції AAA (Authentication, Authorization and Accounting – аутентифікація, авторизація та облік) і шифрування та дешифрування разом з іншими послугами, пов'язаними з Інтернетом. Усі Cloud мають свої власні реалізації безпеки, але функціонально вони усі побудовані на принципі ролей та дозволів. Цей рівень виконує також функції термінатора шифрованого з'єднання DTLS [9].

#### 1.3.7. Middleware Layer - рівень внутрішньо-серверного зв'язку

Даний рівень забезпечує внутрішній Cloud функціонал балансування навантаження, черги повідомлень і передачі потокової інформації. Компоненти даного рівня повинні бути дубльовані і автоматично масштабуватися. Рівень реалізується на основі мікросервісів або PaaS від Cloud провайдерів. Така вимога випливає з парадигми стрибків і провалів обсягу даних. Автоматичне масштабування знижує вартість Backend реалізації. Фактична реалізація сервісу може бути різною, але загальний принцип залишається одним – забезпечити асинхронну передачу повідомлень з буферизацією і перерозподілом навантаження. Таким чином різні компоненти Backend можуть виконувати свою роботу незалежно і горизонтально масштабуватися в залежності від навантаження [9].



*Рис. 1.7. Схематична блок-діаграма патернів внутрішньо-серверного зв'язку*

На рис. 1.7 представлена схематична блок-діаграма патернів внутрішньо-серверного зв'язку. Load Balancer призначений для розподілу навантаження між різними сервісами [9].

Queue – черги забезпечують проміжну буферизацію для реалізації асинхронної роботи послідовних сервісів.

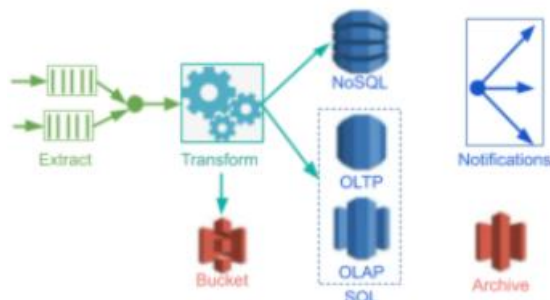
Subscribers – одержувачі, підписуються на відповідні їхньою логікою черзі, щоб отримувати послідовно повідомлення після обробки попередніх повідомлень.

### 1.3.8. ETL layer - рівень збору, обробки та зберігання даних

Внутрішній рівень ETL є третьою операцією ETL. Backend ETL накопичує дані з усіх периферійних пристроїв і шлюзів та відповідає за такі операції:

- збір інформації;
- приведення інформації до стандартного вигляду;
- збереження інформації для подальшого використання;

- управління життєвим циклом інформації включаючи архівування та знищення;
- повідомлення інших сервісів про надходження нових даних.



*Рис. 1.8. Загальна схема реалізації ETL layer*

Загальна схема реалізації цього рівня представлена на рис. 1.8. Операція збору даних (Extract) включає в себе читання інформації з релевантних черг. Операція трансформації може виконуватися спеціалізованими сервісами Cloud [9].

Основне завдання цієї операції – зробити приведення даних до зручного для зберігання, сортування і пошуку виду. Для цього часто дані об'єднуються з різних повідомлень і навіть черг [9].

Операції зберігання (Load) призначені для збереження, сортування і подальшого пошуку інформації. Залежно від типу інформації і варіантів її використання, застосовуються різні інструменти. Якщо дані не мають суворої схеми (колонки таблиці), то вони зберігаються в NoSQL базах. Однак, якщо дані можуть бути систематизовані фіксованою схемою, то використовуються SQL типи баз даних. Останні, в свою чергу мають 2 типу - OLTP (Online Transactional Processing) і OLAP (Online Analytic Processing) [9].

Перший тип більш підходить для самого процесу ETL – записи в базу нових значень, в той час як другий зручніше для пошуку та аналізу даних. Тому часто після завантаження в OLTP базу, у фоновому режимі, дані копіюються в OLAP. Трапляються ситуації, коли дані не зручно або неможливо зберігати в базах даних. Ці дані записуються в Bucket, а метадані записів зберігаються в базах даних. Для скорочення витрат на зберігання, застарілі дані архівуються або видаляються. І

останнім компонентом цього рівня є внутрішня нотифікація про наявність нових збережених даних для подання клієнтам і для сервісів аналізу [9].

#### 1.3.9. Big Data and Analytic Layer – рівень аналітики

Залежить від конкретного додатка IoT. Великі дані і аналітичний рівень отримуватимуть ситуативну інформацію з усього набору периферійних пристроїв. Ця частина менш стандартизована, тому що вона сильно відрізняється від однієї програми до іншої в силу різних завдань рішень. Алгоритми AI/ML також широко використовуються на цьому рівні.

Окремою категорією є передбачення майбутніх подій, таких як необхідні частини на складі, споживання майбутніх ресурсів, погода та ін [9].

#### 1.3.10. Notification layer – рівень повідомлення

На цьому рівні може існувати кілька компонент, але всі вони мають алгоритм повідомлення за передплатою. Клієнтська програма підписується на необхідні події і, коли це відбувається, отримує інформаційний сигнал – повідомлення.

В основному це програми електронної пошти і мобільні клієнти, також телефонні дзвінки, які використовуються для екстреного оповіщення. Мобільний додаток змушений переходити в сплячий режим для енергоспоживання, але ОС iOS і Android мають механізм повідомлень, що вказує на прибуття нових даних [9].

#### 1.3.11. Presentation Layer – рівень представлення

Додаток IoT може мати два потоки: M2M та M2P. Рівень представлення, пов'язаний з потоком M2M, де Backend обробляє інформацію і надає її клієнту або інженеру служби підтримки.

Рівень представлення також відповідає за обслуговування, конфігурацію і зміни стану системи включаючи периферійні пристрої та шлюзи. У ньому представлені команди на керуючі виконавчими механізмами периферійних пристроїв [9].

#### 1.3.12. Configuration Layer – рівень конфігурацій

Цей рівень відноситься до обох потоків – M2M і M2P і працює як сховище для трьох типів статусів периферійних пристроїв:

- актуальний стан периферійного пристрою;

- новий стан периферійного пристрою;
- проміжний статус периферійного пристрою – вказує на процес оновлення від старих станів до нових. Часто цей статус відсутній.

Периферійний пристрій і шлюз мають тільки короткий час підключення до Backend. Будь-яка зміна статусу від клієнта або системи, зберігаються в цьому рівні, і протягом часу зв'язку відправляється на шлюз або периферійний пристрій [9].

Якщо шлюз присутній в схемі передачі інформації, то більша частина інформації від периферійних пристроїв відправляється на серверну частину у вигляді пакетів даних, зібраних з кількох периферійних пристроїв [9].

Якщо підсумувати вищесказане, то можна спостерігати наступні тенденції розвитку в IoT рішеннях [9]:

- датчики поділяються на 2 групи:
  - прості, дешеві, з максимально низьким споживанням енергії. Низькою швидкістю і високою дальністю передачі інформації. Це фактично разові пристрої, які не підлягають обслуговуванню;
  - засновані на відеокамері. Пристрій інтегровано з периферійним комп'ютером. Має вбудовані механізми розпізнавання образів і прийняття базових рішень;
- ETL процеси відбуваються на кількох рівнях – периферійних пристроях, шлюзів і в backend. Поки немає єдиного підходу, що повинно робитися на кожному рівні, але ідеологія така – все що можна обробити має бути оброблено на якомога нижчому рівні;
- основним, універсальним засобом передачі інформації є бездротовий інтернет. Але фактичні протоколи відрізняються на різних рівнях. Логічний рівень зв'язку – протокол LvM2M;
- Backend більшою мірою Cloud. Більшість рішень на сьогоднішній день використовують AWS;
- в якості пристрою відображення фактичної інформації використовується мобільний додаток, а аналітична інформації представляється WEB

додатками. Екстрене оповіщення через мобільний дзвінок з передвстановленим голосовим повідомленням. Електронною поштою виходять в основному звіти.

#### 1.4. Висновки до першого розділу

З вище викладеного можна зробити висновок, що розробка та опис архітектури IoT необхідний як для виробників, так і для розробників. Також потрібно розуміти основні визначення функціональних елементів IoT.

Мікропроцесор – це обчислювальний пристрій, призначений для вироблення керуючих впливів для пристроїв IoT або інших мікропроцесорів на основі обробки даних, одержуваних від сенсорів і відповідно до завантаженої в нього програми [5].

Сенсори – це прилади, що вимірюють вагу, температуру, висоту, швидкість, частоту пульсу, висоту, артеріальний тиск, місце розташування, вологість, освітлення та багато іншого [5].

Пристрій – це прилад, механізм, конструкція, установка, яка використовується для виконання певних функцій або дій з фізичними або живими об'єктами. Пристрої можуть здійснювати певні фізичні дії з різними предметами (піднімати, переносити, кидати, свердлити, стругати, гріти, охолоджувати тощо), вмикати (вимикати, перемикаєти) електроенергію, потоки газів, рідини тощо, переміщувати предмети в просторі, виробляти вироби тощо [5].

Мережа Інтернет – це мережа передачі даних, що інформаційно з'єднує функціональні елементи IoT, кожен з яких має в ній свій унікальний ідентифікатор. Завдяки наявності мережі передачі даних стає можливим створення і функціонування системи і комплексів IoT [5].

Система – це сукупність пристроїв, мікропроцесорів і сенсорів, з'єднаних за допомогою мережі Інтернет, яка призначена для реалізації певних функцій або сукупності функцій задля досягнення конкретної мети без втручання людини або за її участю [5].

Комплекс – це сукупність систем, що функціонують під управлінням комп'ютерної програми, яка містить штучний інтелект, здатний виконувати повністю або частково окремі творчі функції людини або їх певну сукупність. У

комплексах стає можливим самостійне визначення поточних і перспективних цілей його функціонування, а також визначення та зміни цілей функціонування систем, які входять до його складу [5].

Для IoT основною відмінністю є підключення всіх елементів до мережі Інтернет так, щоб забезпечити їх спільне функціонування в умовах їх територіального розподілення [5].

## 2. АРХІТЕКТУРА СИСТЕМИ «РОЗУМНИЙ БУДИНОК»

### 2.1. Історія розвитку концепції «Розумного будинку»

Початком розвитку концепції розумного будинку можна вважати середину ХХ століття. Саме тоді почалася комп'ютерна епоха, що лежить в основі системи. Тоді не було мініатюрних компонентів, які виконують завдання, всі модулі були громіздкими, підключалися до розеток і працювали за допомогою перфокарт. Але виконувати задумане вони могли.

У 60х роках минулого століття Джеймс Сазерленд зміг запрограмувати комп'ютер на включення приладів за визначеним розкладом, зв'язатися з датчиками стеження і сигналізації.

У 1978 році шотландська фірма Pico Electronics створила стандарт передачі інформації, який є універсальним для домашньої автоматизації. Компанією була реалізована шина, за допомогою якої можна оснастити побутові прилади для автоматичного керування. Підключався все через розетки, додатково в комплектації були модулі зв'язку, пульти дистанційного керування, програми.

Після створення єдиного стандарту почалася епоха експериментів. Термінологія «Розумний будинок» офіційно з'явилася в 1984 році від Американської асоціації житлово-будівельних компаній. Далі до розвитку системи приєдналася асоціація електронної промисловості, яка допомогла реалізувати новий стандарт - CEBus.

Масово люди познайомилися з «розумним будинком» завдяки фільму від компанії Disney, в якому був показаний комп'ютеризований будинок. З нульових років масово стали з'являтися фірми-виробники. Справжній прорив зробила компанія Apple, випустивши перший iPhone. З того часу можливості кишенькових комп'ютерів спонукали інженерів провести нові винаходи. До 2012 року в Америці було встановлено понад 1,5 мільйона систем домашнього управління.

### 2.2 Архітектура Smart Building

Новозбудовані будинки часто будуються з використанням інфраструктури «Розумного будинку». З іншого боку, в старіших будинках можна використовувати

інтелектуальні технології. Хоча багато систем розумного будинку все ще працюють на X10 або Insteon, популярність Bluetooth і Wi-Fi зросла [11].

Zigbee і Z-Wave – два найбільш поширених протоколу зв'язку для домашньої автоматизації, які використовуються сьогодні. Обидва використовують технологію mesh-мережі, малопотужні радіосигнали малого радіусу дії для підключення систем «Розумного будинку». Хоча обидва націлені на одні і ті ж додатки для «Розумного будинку», Z-Wave має діапазон від 30 метрів, а Zigbee до 10 метрів, причому Zigbee часто сприймається як більш складний з двох. Чіпи Zigbee доступні від декількох компаній, а чіпи Z-Wave – тільки від Sigma Designs.

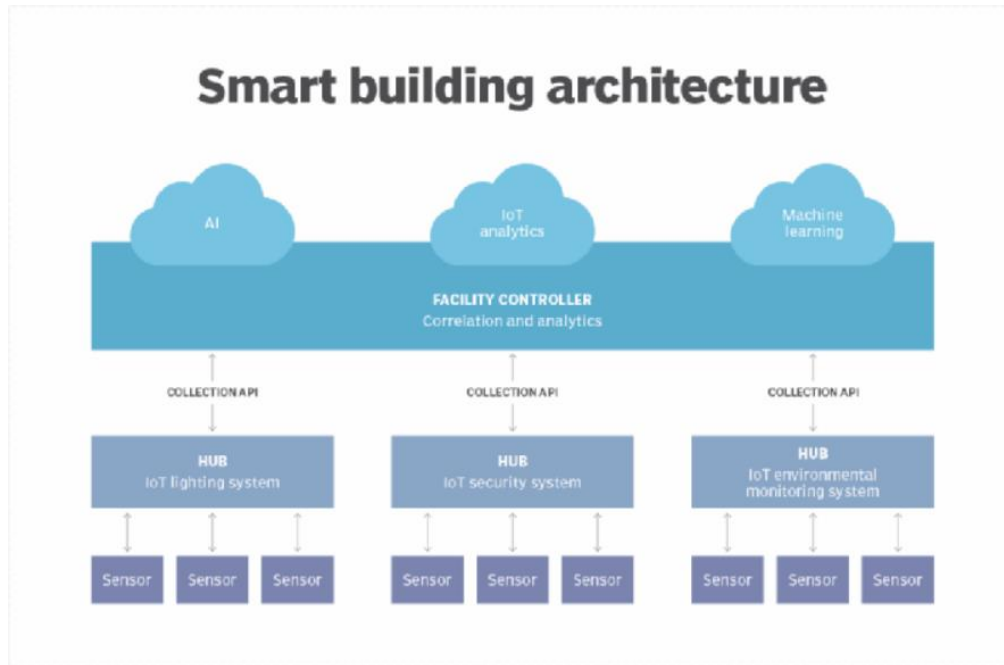
Розумний будинок – це не набір розрізнених інтелектуальних пристроїв і побутової техніки, а тих, які працюють разом, щоб створити віддалено керовану мережу. Всі пристрої управляються головним контролером домашньої автоматизації – концентратором розумного будинку.

*Концентратор розумного будинку* – це апаратний пристрій, який діє як центральна точка системи «Розумного будинку» і здатний сприймати, обробляти і обмінюватися даними по бездротовій мережі. Він об'єднує всі розрізнені додатки в один додаток для «Розумного будинку», яким споживачі можуть дистанційно керувати. Приклади концентраторів «Розумного будинку» є Amazon Echo, Google Home, Insteon Hub Pro, Samsung SmartThings і Wink Hub.

Деякі системи розумного будинку можна створити з нуля. Наприклад, за допомогою Raspberry Pi або іншої макетної плати. Інші можна придбати у вигляді пов'язаного комплекту для «Розумного будинку», також відомого як платформа для «Розумного будинку», яка містить елементи, необхідні для запуску проекту домашньої автоматизації.

У простих сценаріях розумного будинку події можуть бути синхронізовані або ініційовані. Заплановані події засновані на годиннику. Наприклад, опускання жалюзі о 18:00. А ініційовані події залежать від дій в автоматизованій системі. Наприклад, коли смартфон власника наближається до дверей, розумний замок розблокується, а розумне світло ввімкнеться.

Машинне навчання і штучний інтелект (ШІ) стають все більш популярними в системах «Розумного будинку», дозволяючи додаткам домашньої автоматизації адаптуватися до їх середовища. Наприклад, системи з голосовим управлінням, такі як Amazon Echo або Google Home, містять віртуальних помічників, які вивчають і персоналізують «розумний будинок» відповідно до своїх уподобань і особливостей жителів.



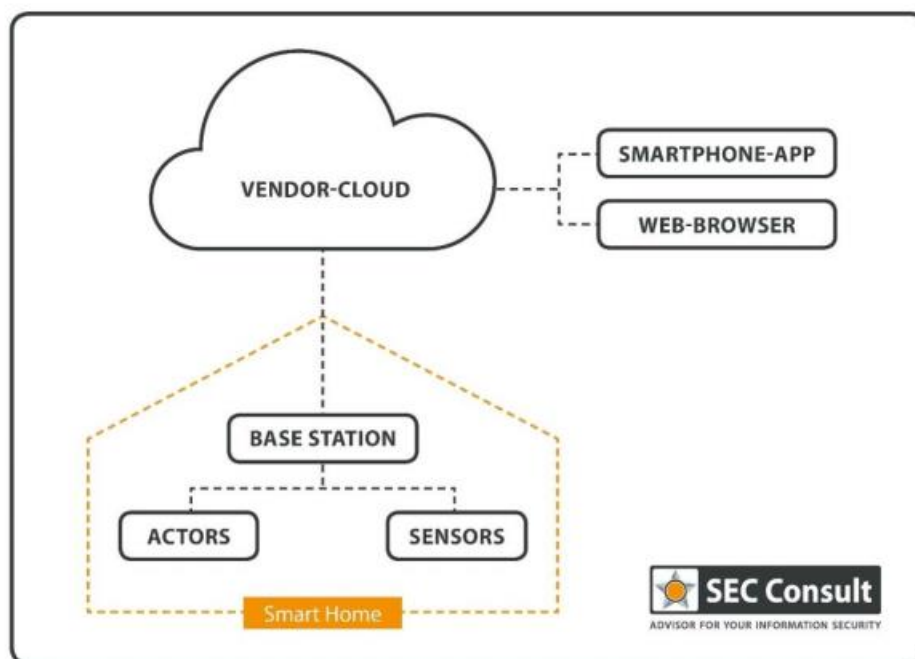
*Рис. 2. Архітектура Smart Building*

### 2.3. Базові елементи Smart Home System

Хоча існує безліч різних компонентів розумного будинку, їх можна звести до декількох базових необхідних елементів, які визначають архітектуру типової системи розумного будинку, як показано на рис 2.1. Перш за все, це датчики, які вимірюють стан навколишнього середовища і жителів, і є актори, які зберігають або змінюють цей статус [12].

Всі ці датчики і суб'єкти підключені до центральної базової станції, яка збирає і контролює всю інформацію. Крім того, база підключена до якогось клієнта віддаленого управління, зазвичай до веб-браузеру або додатком для смартфона. Це з'єднання може бути встановлено безпосередньо, наприклад, у тій же мережі або, як в більшості випадків сучасних систем розумного будинку, через хмарний

інтерфейс, що надається постачальником для забезпечення віддаленого доступу до системи [30].



*Рис. 2.1. Базові елементи Smart Home System*

#### 2.4. Архітектура управління «Розумним будинком»

Приділену управління розумним будинком це не щось нове і в великих комерційних системах успішно реалізовано. Але для більшості людей установка таких систем є невинновдано дорогою і все популярнішим стає розробка своєї системи або адаптація відкритих систем автоматизації під свої потреби. При такому підході для здійснення віддаленого управління розумним будинком доводиться стикатися з низкою проблем: необхідно мати можливість звернутися до системи ззовні, для цього потрібно мати статичний IP адреса. В основному інтернет-провайдери надають динамічний, а статичний не завжди можливо отримати або використовувати такі сервіси як DynDNS або NO-IP [13].

В якості альтернативи пропонується використовувати хмарні сервіси, які в Останнім часом все доступніше і при невеликих запитах є можливість використовувати їх ресурси безкоштовно. Також даний спосіб дає деякі переваги в порівнянні з використанням статичного адреси або динамічного DNS [32].

Для початку потрібно визначитися з архітектурою побудови системи. Існує два види архітектури системи розумний будинок:

- децентралізована
- централізована.

*Децентралізована архітектура* ґрунтується на тому, що всі вузли в мережі рівноправні. У цьому розумному будинку кожна виділена підсистема наділена рівними правами по відношенню до інших підсистем, а вся логіка роботи розумного будинку підпорядкована відносно простим сценаріями, відповідно до яких керуючі сигнали передаються безпосередньо від однієї підсистеми до іншої при настанні певних умов.

*Централізована архітектура* передбачає наявність центрального контролера, який управляє підключеними до нього модулями.

Також існує *суміщений підхід*, при якому центральний контролер управляє декількома підсистемами, які при виникненні неполадок з сервером і далі продовжують виконувати якийсь мінімально закладений в них функціонал.

Для «Розумного будинку», потрібен центральний сервер управління тому що з його допомогою простіше здійснити клієнт-серверну взаємодію між мобільним додатком та розумним будинком, а також зв'язок з хмарним сервером. У системи віддаленого управління були виділені наступні основні компоненти:

- мобільний додаток;
- хмарний сервер;
- домашній сервер;
- мікроконтролер для управління датчиками і сенсорами.

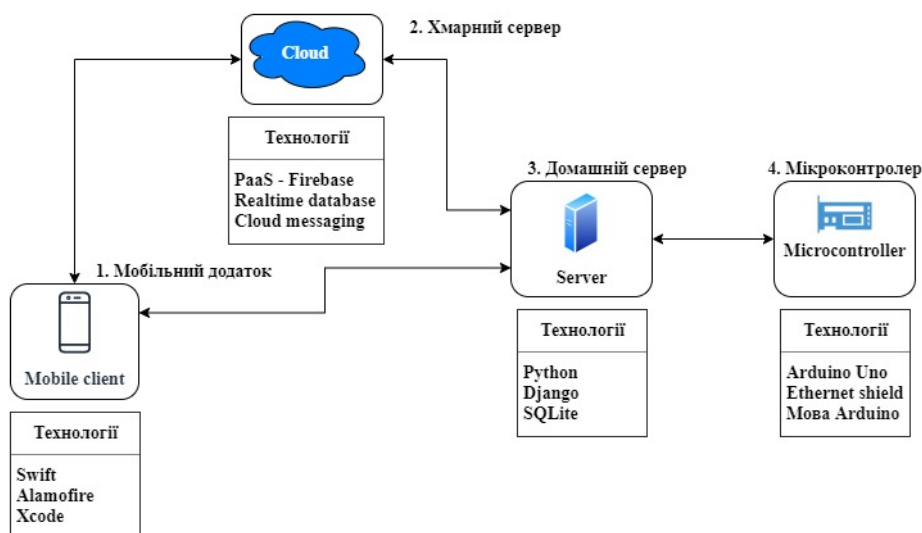


Рис. 2.2. Архітектура системи віддаленого управління

Розглянемо ці компоненти докладніше.

#### 2.4.1. Мобільний додаток

Мобільний додаток необхідний для віддаленого управління розумним будинком. Звісно можна використовувати і звичайний браузер, але додаток більш зручніший і має інші переваги, як push - повідомлення.

Push – це технологія поширення інформації від сервера клієнту, а Push повідомлення – це невеликі спливаючі вікна на екрані вашого пристрою.

За допомогою мобільного додатка здійснюється передача команд на хмарний сервер, а також отримання звітної інформації про стан датчиків [37].

Додаток під платформу iOS розробляється в середовищі Xcode з використанням мови програмування Swift. Для отримання даних і відправки команд використовується фреймворк Alamofire. Для локального зберігання даних в додатку використовується фреймворк CoreData. Додаток має мати два режими підключення до сервера: *безпосередньо з локальної мережі і через хмарний сервер*.

Після запуску додаток перевіряє наявність сервера в локальній мережі і підключається до нього, а якщо його немає, то відбувається підключення до хмарного сервера. Для отримання даних і відправки команд на сервер використовується REST API. Дані передаються в форматі JSON [13].

#### 2.4.2. Хмарний сервер

Хмарна модель надання сервісів швидко і активно розвивається в останні роки і більшість функцій по обробці, управлінню та зберігання інформації вже зараз доступні в формі хмарних сервісів і число їх продовжує бурхливо рости.

Серед найкращих і найбільш популярних PaaS можна виділити Firebase від компанії Google. Firebase є базою даних, яка змінюється в реальному часі і зберігає дані в JSON. Будь-які зміни в базі даних відразу синхронізуються між усіма клієнтами, або девайсами, які використовують одну і ту ж базу даних. Іншими словами, оновлення в Firebase відбуваються миттєво.

Хмарний сервер приймає команди з мобільного пристрою і посилає їх на домашній сервер. Якщо з домашнього сервера приходить інформація про зміну стану будь-якого пристрою в будинку, то хмарний сервер посилає push-повідомлення на мобільний пристрій [13].

1. Додаток реєструється для отримання push-повідомлень.
2. iOS запитує у сервера Apple Push Notification Service (APNS) токен девайса.
3. Додаток отримує токен девайса. Можна вважати, що токен – це адреса для відправки push-повідомлень.
4. Додаток відправляє токен девайса на наш хмарний сервер.
5. Коли відбудеться яка-небудь подія (зміниться стан датчика і т. п.), сервер надсилає запит на відправку push-повідомлення.
6. Хмарний сервер відправляє push-повідомлення в APNS.
7. APNS відправить push-повідомлення на девайс користувача.

Коли користувач отримає push-повідомлення, з'явиться повідомлення, та буде відтворений звуковий сигнал.

#### 2.4.3. Домашній сервер

Домашній сервер призначений для отримання команд від хмарного сервера і їх подальшої передачі на контролери, а також передачі в зворотному порядку даних з датчиків, що надходять на контролер. Через домашній сервер також забезпечується управління системою по локальній мережі з мобільних пристроїв по Wi-Fi. У ролі домашнього сервера можна використовувати звичайний персональний комп'ютер [13].

Розробляється із застосуванням мови Python і бібліотеки Django. Для зберігання даних про стані підключених пристроїв використовується база даних SQLite. Розберемо алгоритм роботи сервера на прикладі запиту з мобільного додатку на отримання даних з датчиків в певній кімнаті [13]:

1. Отримання запиту.
2. Визначення потрібного уявлення за шаблоном.
3. Подання запитує у моделі дані кімнати.
4. За отриманою адресою надсилається запит контролеру на отримання даних з датчиків.
5. Отримання відповіді від контролера і складання відповіді для додатка.
6. Відправка відповіді з додатком у форматі JSON.

#### 2.4.4. Мікроконтролер

Контролер Arduino зарекомендував себе як зручна платформа швидкої розробки електронних пристроїв для новачків і професіоналів. Платформа користується величезною популярністю в усьому світі завдяки зручності і простоті мови програмування, а також відкритій архітектурі і програмного коду. Пристрій програмується через USB без використання програматорів [13].

Мікроконтролер на платі програмується за допомогою мови Arduino (заснований на мові Wiring) і середовища розробки Arduino (заснована на середовищі Processing) Контролер керує підключеними датчиками, приймає команди з домашнього сервера і відправляє йому інформацію про зміну станів датчиків [13].

Алгоритм роботи контролера:

1. Сервер надсилає запит на контролер.
2. Контролер приймає запит.
3. Обробка запиту, отримання параметрів із запиту.
4. Отримує дані з датчиків або відправляє дані на датчик.
5. Формує відповідь сервера.
6. Посилає відповідь.

## 2.5. Висновок до другого розділу

Системи розумного будинку і Інтернет речей – дуже цікаві технології, які пов'язують наш віртуальний світ з нашим реальним життям. Вони вводять безліч нових протоколів зв'язку і об'єднують їх з усіма традиційними веб-технологіями [12].

Мобільні пристрої завжди під рукою, а тому ідеально підходять на роль інтерфейсів пристроїв розумного будинку. У той же час інтеграція розумного будинку в хмару покликана забезпечити віддалене управління всіма системами будинку.

Розумний будинок є дуже перспективною галуззю, розвиток якої дозволяє поліпшити комфорт, безпеку, заощадити час і витрати на комунальні послуги [13].

### 3. АНАЛІЗ АРХІТЕКТУРИ СИСТЕМ «РОЗУМНИЙ БУДИНОК»

Кількість пристроїв, що полегшують побут сучасної людини, зростає день у день. Розрізнені комунікації і гаджети вимагають все більше уваги і, як наслідок, часу для налаштування.

Ідея домашньої автоматизації – це об'єднання всіх девайсів під централізоване управління реалізована за допомогою системи «Розумний будинок».

Світ сучасних технологій влаштований так, що ідеальних рішень не існує. Основне завдання для того, хто хоче зробити свій будинок розумним – це визначити, які функції і можливості будуть необхідні, і як вони реалізуються у різних виробників.

#### 3.1. Samsung SmartThings

SmartThings – це екосистема додатків Інтернету речей.

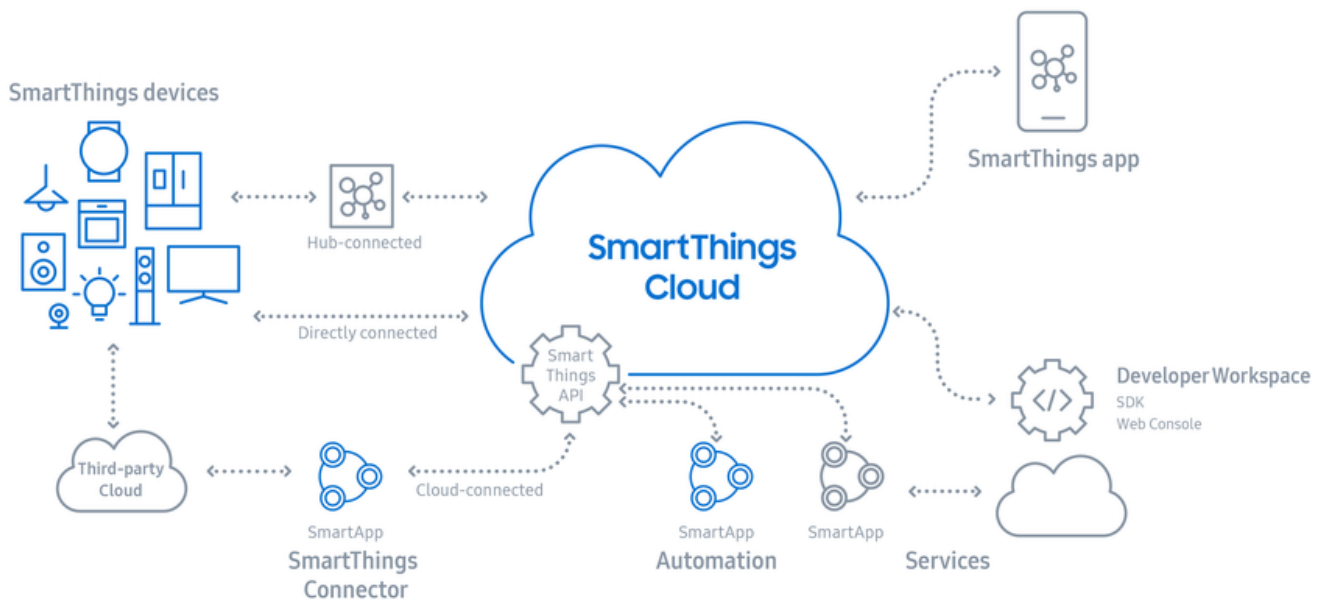


Рис. 3. Архітектура SmartThings

Проект SmartThings стартував в 2012 році в рамках кампанії на Kickstarter. Основна ідея полягала в тому, щоб реалізувати рішення для розумного домашнього середовища через концентратор, підключений до набору «речей» таких, як датчики температури, вологості та датчики диму. Коли проект був запусканий, він був доповнений додатком для смартфона, здатним для зв'язку з віддаленим концентратором [17].

2014 рік став важливою віхою для SmartThings, оскільки вона була придбана Samsung Electronics. Первісна архітектура зазнала значних змін і стала дійсно орієнтованою на хмару платформою [17].

Як показано на рис. 3, можливо підключення пристроїв до хмарної серверної частини за допомогою трьох різних шляхів, навіть без допомоги прямого підключення до концентратора [17].

Інший тип з'єднання – це з'єднання з хмарою, яке дозволяє реалізувати непряний канал зв'язку між (хмарним) стороннім пристроєм та хмарою SmartThings. Але рекомендується наявність концентратора, який діє як шлюз між пристроями і хмарою. Деякі операції можуть виконуватися локально без необхідності запитувати серверну частину [17].

У цьому випадку SmartThings спілкуються через концентратор, і вони покладаються на протоколи бездротового зв'язку ZigBee або Z-Wave. У додатку SmartThings об'єкти організовані і згруповані відповідно до приміщенням, в якому вони знаходяться. Таким чином, концепція приміщення є ключовим аспектом для клієнтів SmartThings [17].

SmartThings включає в себе концепцію автоматизації, яка дозволяє користувачеві взаємодіяти з екосистемою без будь-якого ручного втручання. Відносно автоматизації є дві можливі стратегії: перша заснована на WebHook, друга – на функціях AWS Lambda. Наприклад, можна визначити стратегію автоматизації, призначену для регулювання інтенсивності світла в конкретному приміщенні відповідно до змін погоди. Це хмарне рішення також підтримує шифрований зв'язок між усіма підключеними компонентами через протокол SSL/TLS [17].

Запропонована архітектура SmartThings, націлена на домашнє середовище або на концепцію інтелектуальної будівлі. Її особливості дозволяють застосовувати її в більш широкому контексті. Зокрема, завдяки хмарним з'єднанням можна було б підключити сенсорну мережу [17].

Більше про елементи архітектури Samsung SmartThings [14].

### 3.1.1. SmartThings Devices

SmartThings Devices – можуть бути не тільки пристрої з комплекту SmartThings, але і безліч сторонніх зі списку партнерів.

Якщо пристрою немає у списку офіційно підтримуваних, ви можете пошукати до нього Device Handler – він може бути наданий спільнотою розробників – або написати свій власний [14].

### 3.1.2. Способи підключення

Підключати пристрої до платформи ви можете трьома способами [14]:

- Hub-connected – через хаб-концентратор, актуально для пристроїв, які свого виходу в Інтернет не мають, а з'єднуються через традиційні для такого роду систем бездротові мережі, як ZigBee;
- Directly-connected – через Інтернет безпосередньо до хмари. Нещодавно SmartThings випустили відеокамеру, лампочку і розумну розетку, і вони працюють без хаба - через Wi-Fi;
- Cloud-connected – через сторонню хмару для тих пристроїв, у яких вже є своя хмарна екосистема.

### 3.1.3. Хаб

Зазвичай для підключення різноманітної апаратури в приміщенні повинен стояти спеціальний маленький сервер «Розумного будинку», що підтримує різні протоколи. Він же – хаб, концентратор, шлюз [14].

### 3.1.4. Додаток

SmartThings App – це додаток, який можна встановити на основні мобільні ОС і з його допомогою можна [14]:

- додавати нові пристрої в систему;
- контролювати вже додані;
- призначати правила автоматизації;
- отримувати повідомлення через push і SMS.

### 3.1.5. Абстракції

У SmartThings, як і в багатьох інших платформах «Розумного будинку», є такі абстракції: кімнати, сцени, сценарії [14].

Пристрої групуються в кімнати (Rooms). Ви просто приєднуєте в додатку ті чи інші гаджети до одній кімнаті, і називаєте їх, як «Кухня», «Ванна», «Вітальня» [14].

Ще є сцени (Scenes) – це такі ситуації, як «сон», «відпочинок», «робота», «перегляд фільму» - кожна із цих ситуацій характеризується своїм набором станів пристроїв [14].

### 3.1.6. Сцени та кімнати у додатку

І сценарії (Routines): правила автоматизації «якщо - то». За замовчуванням у програмі чотири сценарії: «Я повернувся», «На добраніч», «Пока», «Доброго ранку». Можна налаштувати активацію сценарію за умовою настання ранку, або в разі руху, або в разі відкриття дверей, і так далі [14].

Наприклад: коли я входжу додому – заблокувати мережевий трафік на домашній камері відеоспостереження, щоб виключити можливість витоку моїх особистих відео. Або жорстокий виховний приклад: примусово відключити планшет від мережі, коли настав 11 години вечора, щоб не дати собі залипати в Інтернеті замість здорового сну [14].

Більш того, в програмі SmartThings на телефоні є цікава можливість – запам'ятовування типових дій. Коли часто користуєшся телефоном, він з часом починає тобі підказувати, як краще налаштувати сцени [14].

Для додавання в систему нових пристроїв передбачена система плагінів. Розробник може запрограмувати плагін для свого власного пристрою – призначений для користувача інтерфейс всередині програми SmartThings і логіку роботи пристрою. По суті, це веб-додаток. З точки зору розробника, плагін може підписуватися на події або повідомлення з пристрою, посилати йому повідомлення, моніторити статус з'єднання з пристроєм [14].

### 3.1.7. Особливості роботи SmartThings

Уся робота SmartThings побудована навколо головного контролера – Hub. Саме Hub підтримує бездротову мережу, яка потрібна для підключення і роботи з усіма приладами «Розумного будинку». Hub виступає в ролі своєрідного роутера, але стандартний роутер з Wi-Fi не підійде для таких цілей. Однак для повноцінної

роботи системи необхідні обидва ці пристрої. Hub і Wi-Fi роутер приєднуються один до одного за допомогою Ethernet [16].

Hub забезпечений потужним процесором, що дозволяє підтримувати режим відеоспостереження. В умовах відключення від електричної мережі його батарея забезпечує до десяти годин безперервної роботи. Цього достатньо для підтримки системи «Розумний дім» без збоїв та в разі відключення електроенергії в будинку або квартирі. При цьому сполучення з усіма пристроями і контролером не буде втрачено і робота системи буде відбуватися за звичним сценарієм. Для таких випадків Hub використовує інтерфейс Zigbee або Z-wave, а також здатний підтримувати з'єднання по локальному IP [16].

У зв'язку з широким поширенням протоколів Zigbee та Z-wave на ринку «розумних» речей, компанія Samsung забезпечила SmartThings Hub приймачами для обох версій. Тому проблеми з підключенням зовнішніх пристроїв виключені. У разі, якщо будь-який пристрій не підтримує дані протоколи, на допомогу прийде стандарт Wi-Fi або веб-сервіси IFTTT [16].

Декілька слів про безпеку роботи системи SmartThings. У певному сенсі, сторонні прилади, підключені до цієї мережі, чисто гіпотетично можуть бути слабкою ланкою при зломі системи зловмисником. Однак розробники SmartThings запевняють, що ліцензійні пристрої, що володіють міткою «Працюю зі SmartThings», проходять якісну перевірку і відповідають всім сучасним вимогам безпеки. Тому при грамотній організації і підключенні сертифікованих смарт-пристроїв до «Розумного дому», проблеми з безпекою повністю виключені [16].

### 3.1.8. Плюси і мінуси системи Samsung SmartThings

Плюси:

- гнучкий мобільний додаток;
- підтримка Amazon Alexa, Google Assistant;
- сумісний із сотнями розумних пристроїв;
- потужне співтовариство розробників;
- надзвичайно гнучкий;
- централізоване управління;

- батарея резервного копіювання;
- підтримка Zigbee та Z-Wave.

Мінуси:

- обмежене підключення Bluetooth.

### 3.2. Apple HomeKit

Home Kit – це програмна платформа Apple, яка дозволяє користувачам iOS, macOS, watch OS і tv OS управляти розумними аксесуарами в будинку як через додаток Home, так і через голосовий помічник Siri [17].

Apple HomeKit ґрунтується на незалежних пристроях, пов'язаних один з одним по Wi-Fi. Багато з них є самостійними і можуть використовуватися поодиночі, навіть якщо нічого «розумного» ви більше купувати не плануєте. Але деякі пристрої працюють по протоколу Bluetooth, тому вимагають головного пристрою з функціями шлюзу. В основному це кнопки і датчики (температури, відкриття, руху і т. д.). Вони можуть підключатися до iPhone в зоні видимості, але краще в якості шлюзу використовувати пристрій удома: приставка Apple TV, колонка Home Pod або просто планшет iPad. Правда, дальність дії Bluetooth все ж вельми обмежена – у великому будинку це може стати проблемою, і доведеться купувати не один «хаб» [15].

Apple не виробляє жодного пристрою HomeKit, не рахуючи «хабів». Все інше зроблено партнерами і сертифіковане за стандартом MFi. Наявність суворої сертифікації є одночасно і плюсом, і мінусом. З одного боку, в ній приділено багато уваги безпеки і стійкості до зломів, з іншого – через це у Apple зовсім небагато партнерів по HomeKit, і ціна на пристрої в рази вище. Всього система налічує близько 450 офіційно підтримуваних пристроїв [15].

ПО Homebridge дозволяє додавати в HomeKit пристрої не тільки з багатьох інших систем РБ. Само собою, це не найідеальніший варіант з точки зору цілісності і безпеки. Зате це хороший компроміс для тих, хто не хоче миритися з обмеженим різноманітністю HomeKit, але при цьому отримати зручне нативне управління з iPhone або Apple Watch [15].

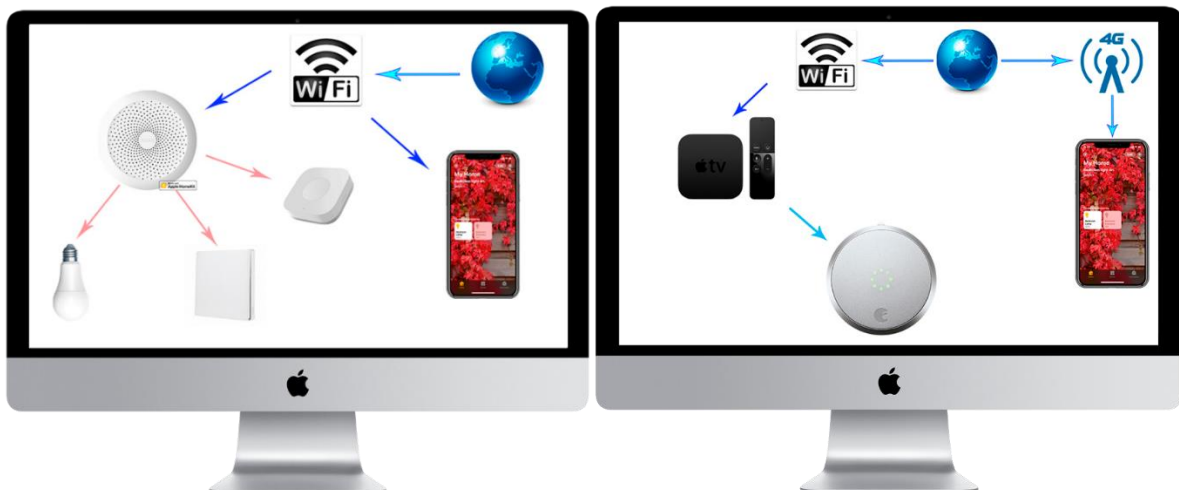


- виконує запрограмовані сценарії: включити опалення або кондиціонер в потрібний момент, забезпечує роботу охоронної сигналізації, реєструє відвідувача і передає про нього відомості користувачеві;
- повідомляє власника житла про виникнення нештатних ситуацій.

### 3.2.2. Топологія

Девайси які додаються безпосередньо в Apple HomeKit по Bluetooth або Wi-Fi. У більшості випадків, діє наступне правило [19]:

- Wi-Fi – найчастіше це аксесуари, харчування яких здійснюється зовнішнім джерелом живлення (220В);
- Bluetooth працює від батарейок;
- Девайси ZigBee, Z-Wave, 433, 868 і т. д., які додаються в HomeKit через шлюз, концентратор або hub. Харчування можливо як через мережу, так і від батарейок.



*Рис. 3.2. Топологія Apple HomeKit зі шлюзами*

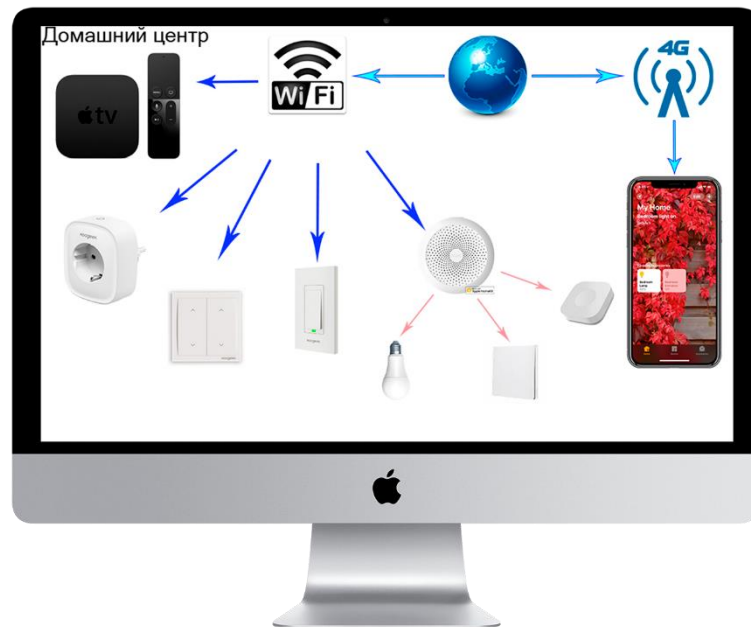
### 3.2.3. Шлюзи

Apple TV, Home Pod, iPad – це повноцінні шлюзи розумного будинку, здатні контролювати підключаються до них безпосередньо Bluetooth та Wi-Fi пристрої [19].

Якщо підключити розумний дверний замок до Hub по Bluetooth, сигнал може бути слабким, якщо вони знаходяться далеко один від одного. У такому випадку, Ви не зможете віддалено відкрити двері [19].

Пристрої, які працюють по протоколу Bluetooth не потребують роутера, а тому якщо ви приходите додому, а світло вимкнене, ви все одно зможете відкрити замок [19].

#### 3.2.4. Віддалений доступ



*Рис. 3.3. Топологія відділеного доступу Apple HomeKit*

Для віддаленого управління необхідний "Домашній центр", в ролі якого можуть виступити iPad (iOS 10 або вище), Home Pod, Apple TV. Таким чином, Домашній центр дозволяє об'єднати всі пристрої для доступу ззовні [19].

Якщо потрібен віддалений доступ або автоматизація (рис. 3.3), потрібно придбати Apple TV, iPad або Home Pod – програмно це реалізувати за поки неможливо [19].

Також хотілося б відзначити, що не всі Apple TV корисні, і можуть бути Домашніми центрами. Рекомендується використовувати Apple TV4 і вище.

Якщо в мережі використовується кілька Домашніх центрів, вони підмінюють один одного непомітно для користувача, якщо якийсь стає недоступний. А усі створені автоматизації синхронізуються між ними.

iPad і Home Pod – це такі ж шлюзи і центри розумного будинку, як і Apple TV, в мережі їх може бути велика кількість [19].

#### 3.2.5. Сценарії

Однією з корисних функцій HomeKit є сценарії. Можна згрупувати пристрої одного типу для запуску однією кнопкою. В автоматизації є можливість прив'язати до конкретної події [19].

### 3.2.6. Автоматизації

При використанні Домашнього центру в HomeKit з'являється можливість створювати Автоматизації [19]:

- Створення автоматизації по приходу/відходу з будинку. "Якщо хтось прийшов додому, то ...";
- Запуск автоматизації за часом. "Вимкнути світло о 22.30";
- Створення автоматизації станом датчика. "Якщо датчик спрацював, то ...".

Приклад: Якщо ми виходимо з дому, iPhone за допомогою геолокації розуміє це і запускає сценарій "Йду з дому". Після цього, в HomeKit відключаються аксесуари, які були додані в цей сценарій.

Кому недостатньо штатного функціоналу HomeKit, можна скористатися сторонніми додатками для створення автоматизацій. Ці програми працюють з нативним framework HomeKit. Автоматизації продовжують працювати навіть при видаленні цього стороннього додатка [19].

### 3.2.7. Голосовий помічник Siri

За допомогою Siri можна управляти всіма аксесуарами, навіть якщо вас там немає. Крім того, Голосовий асистент знає стан всіх аксесуарів. Додатково налаштовувати нічого не потрібно. Функціонал працює на iOS, macOS, watch OS. На Apple TV і Home Pod тільки англійською мовою [19].

Як бачимо, якщо звернеться до голосових асистентів і сказати їй: "Включи світло в дитячій", вона розуміє що світильник знаходиться в кімнаті "Дитяча", а пристрій називається "Світло" [19].

У додатку Будинок є можливість поділитися Вашим будинком з учасниками сімейного доступу і людьми, які прийшли до вас в гості [19].

З корисного хочеться відзначити, що є можливість обмежити використання тільки тоді, коли підключений до домашньої мережі. Для гостей дуже актуально.

З недоліків можна відзначити відсутність можливості передати управління конкретним аксесуарам [19].

### 3.2.8. Механізми безпеки і захисту в Apple HomeKit

Apple HomeKit є, мабуть, найбезпечнішою платформою для контролю за розумним будинком. Забезпечується це шляхом застосування особливого способу шифрування оперативних даних, які передаються від пристрою iOS з якого здійснюється управління, до розумних гаджетів і навпаки [20].

У кодовому ядрі HomeKit застосовується шифрування наскрізного типу. Це спосіб передачі даних, при якому доступ до всієї конфіденційної інформації та даними, що надходять мають тільки ті, хто в цьому умовному "спілкуванні" безпосередньо беруть участь [20].

Припустимо, ви контролюєте свій розумний будинок через iPhone з встановленим додатком Home – вся інформація знаходиться в цьому додатку і опції контролю за пристроями будуть доступні виключно вам як власникові. Жодна третя сторона не зможе як-небудь перехопити інформацію, підробити команди управління і отримати контроль над автоматизацією вашого будинку [20].

### 3.2.9. Плюси і мінуси HomeKit

HomeKit володіє такими перевагами:

- Простота використання. Для інтеграції приладу, сумісного з програмою управління не потрібно особливо глибоких знань в техніці або програмуванні;
- інтеграція з системою Rubetek, яка дозволяє купити девайси за доступною ціною, які хоч і не пов'язані безпосередньо з HomeKit, але за допомогою додатка це стає досить просто;
- можливість застосування для управління будь-якого мобільного девайса від Apple, зі встановленою на ньому ОС відповідної версії;
- голосовий помічник Siri. Він здатний виконувати команди користувача. Оскільки асортимент можливих команд не дуже різноманітний, то Siri розпізнає їх майже бездоганно;

- гарантія високої якості. Всі прилади, що допускаються до роботи в комплексі з HomeKit, проходять обов'язкову сертифікацію фахівців Apple.

**Серед недоліків можна виділити наступні:**

- Вузький асортимент сумісних приладів;
- Висока вартість приладів, що працюють з HomeKit;
- Недостатня інтелектуальність системи. Робота її ґрунтується на реалізації певних закладених сценаріїв, нехай часом і досить складних. Тому говорити про 100% інтелектуальності програми поки рано: це скоріше автоматизована система управління інтегрованими в неї приладами.

### 3.3. Xiaomi Smart Home

Система від Xiaomi куди більш різноманітна. Тут є Zigbee-пристрої з підключенням до шлюзу: більшість датчиків, лампочок, розеток і вимикачів. Шлюзів теж кілька. Найпопулярніший з них, Mijia Gateway (рис. 3.4), вставляється прямо в розетку і вдає із себе щось середнє між нічником і радіоточкою. Він може і сам брати участь в автоматизації: вимірювати освітленість, включати настроюється підсвічування, запускати радіостанції. Але головна його функція – об'єднувати кілька пристроїв одного протоколу в загальну мережу. Всього до нього можна під'єднати до 64 пристроїв.



*Рис. 3.4. Xiaomi Mijia Gateway*

Ще одна варіація шлюзу – Aqara Gateway, зовні схожа на перший. Усередині відмінності теж мінімальні - прибрали лише підтримку інтернет-радіо. Зате вона буває в версії з європейської виделкою (не потрібні адаптери) і, що найважливіше, підтримує інтеграцію в систему Apple HomeKit.

Третій варіант шлюзу, Aqara A/C Companion (рис 3.6), особливої популярності за межами Китаю не знайшов. Він виглядає як розетка для підключення техніки з «китайським» вилок, а замість підсвічування отримав інфрачервоні світлодіоди. Він виступає в якості універсального пульта для техніки, що не володіє Wi-Fi підключенням [21].



*Рис. 3.5 Aqara A/C Companion*

Крім Zigbee, у Xiaomi є і самостійні пристрої з Wi-Fi. В основному це складна техніка, що вимагає широкого каналу для передачі даних: роботи-пилососи, холодильники, мультиварки і так далі. Але є і прості лампочки/розетки. Якщо збирати «розумний будинок на мінімалку», без складних сценаріїв, то вони можуть стати цілком підходящим варіантом. Ніякого шлюзу купувати не доведеться, але все взаємодія буде відбуватися через інтернет, без можливості локального управління [21].

В системі Mi Home є й певна кількість Bluetooth-пристроїв. Як правило, це допоміжні речі – датчики, пульти і так далі. Є і дверні замки з Bluetooth. Всі їх можна підключати безпосередньо до смартфона, багато хто має свій окремий

додаток. Якщо є необхідність керувати ними віддалено і не підключати їх кожен раз до телефону, то можна скористатися Bluetooth-шлюзом. Крім самостійного хаба, цю функцію можуть виконувати і деякі інші пристрої екосистеми. Наприклад, нічник Mijia Bedside Lamp і зволожувач повітря Smartmi Air Humidifier [21].

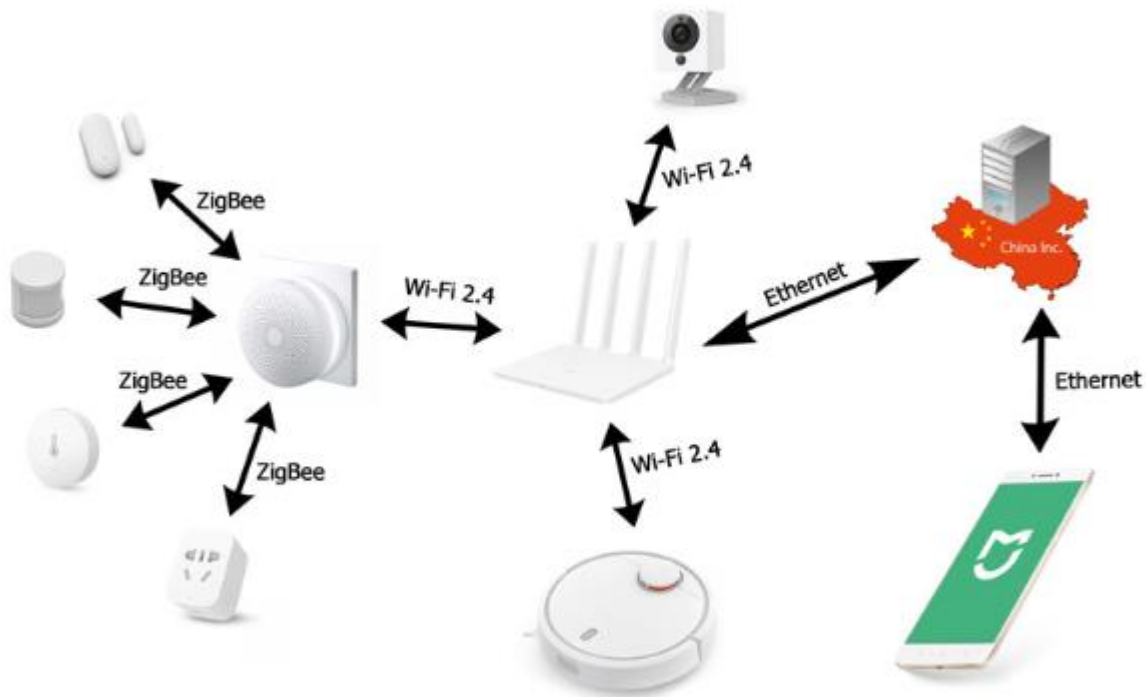
### 3.3.1. Принцип роботи «розумного будинку» Xiaomi

Для управління розумним будинком створили інтелектуальний комплект електроніки Smart Kit Home. У продаж поставляється він у вигляді готового набору до підключення. У нього входять [22]:

- Центральний хаб – інтелектуальний центр всієї системи. Має вигляд пластикової шайби висотою 37, і діаметром 80 мм. Його особливістю є штепсель китайського стандарту Type-1, з трьома ріжками. Шлюз забезпечений різнокольоровими світлодіодними індикаторами, що сигналізують користувачеві про роботу приладу.
- Дистанційна кнопка, яка працює в якості тригера ( «пусковий схеми»).
- Камера відеоспостереження.
- Електромагнітний детектор відкриття вікон і дверей.
- Датчик руху.
- Кріплення-липучки для установки елементів системи в різних місцях квартири.
- Пристосування для скидання налаштувань, виконане у вигляді скріпки.

Крім цього комплекту, дочірні компанії Xiaomi, Mijia і Aqara випускають такі різновиди хабів [22]:

- Smart Home Multifunctional Gateway від Mijia. Має круглу форму, і одночасно використовується у вигляді світильника, дверного дзвінка і оповіщувача сигналізації.
- Air Conditioning від Aqara. Виповнюється як в круглому, так і в квадратному варіанті.



*Рис. 3.6. Архітектура «Розумного будинку» Xiaomi*

В основі роботи будь-якого варіанту «розумного будинку» від Xiaomi лежить специфікація протоколів ZigBee. Вона є окремою мережею, подібно Bluetooth або Wi-Fi, виступаючи в якості надбудови до базового стандарту IEEE 802.15.4. Дана мережа забезпечує стабільний зв'язок з датчиками, що вкрай важливо для коректного функціонування системи. Інший плюс ZigBee – низька енергоємність, що дозволяє працювати комплексу «розумний дім» довгий час функціонувати в автономному режимі. Наприклад, датчик Smart Kit Home здатний працювати на одному елементі в 3 вольти близько 1,5 років. При цьому не має значення, яким з підрозділів компанії випущений хаб і розумний прилад – Mijia або Aqara. Всі вони прекрасно взаємодіють один з одним завдяки протоколу ZigBee [22].

Такі недоліки специфікації ZigBee, як невеликий радіус дії і невисока пропускна здатність для системи «розумний дім» не є критичними. Інший невеликий мінус технології полягає в неможливості з її допомогою безпосередньо управляти інтелектуальними приладами, так як інтерфейс ZigBee не підтримує сучасними мобільними пристроями. Тому для передачі команд через смартфон або планшет, буде потрібно підключення до шлюзу-посередника за допомогою Bluetooth або Wi-Fi-з'єднання. Хаб Air Conditioning також здатний управляти

розумними датчиками за допомогою інфрачервоного зв'язку за допомогою голосових команд [22].

З мобільними пристроями шлюз здатний працювати через Wi-Fi-роутер, віддалений від нього, згідно з даними техпаспорта, на відстань не більше 2-3 метрів. На ділі ж ця відстань може складати до 5 метрів. Хаби Mijia і Aqara здатні взаємодіяти з мобільними пристроями, що працюють на Android і iOS. На екрані смартфона або планшета при цьому можуть відображатися такі піктограми [22]:

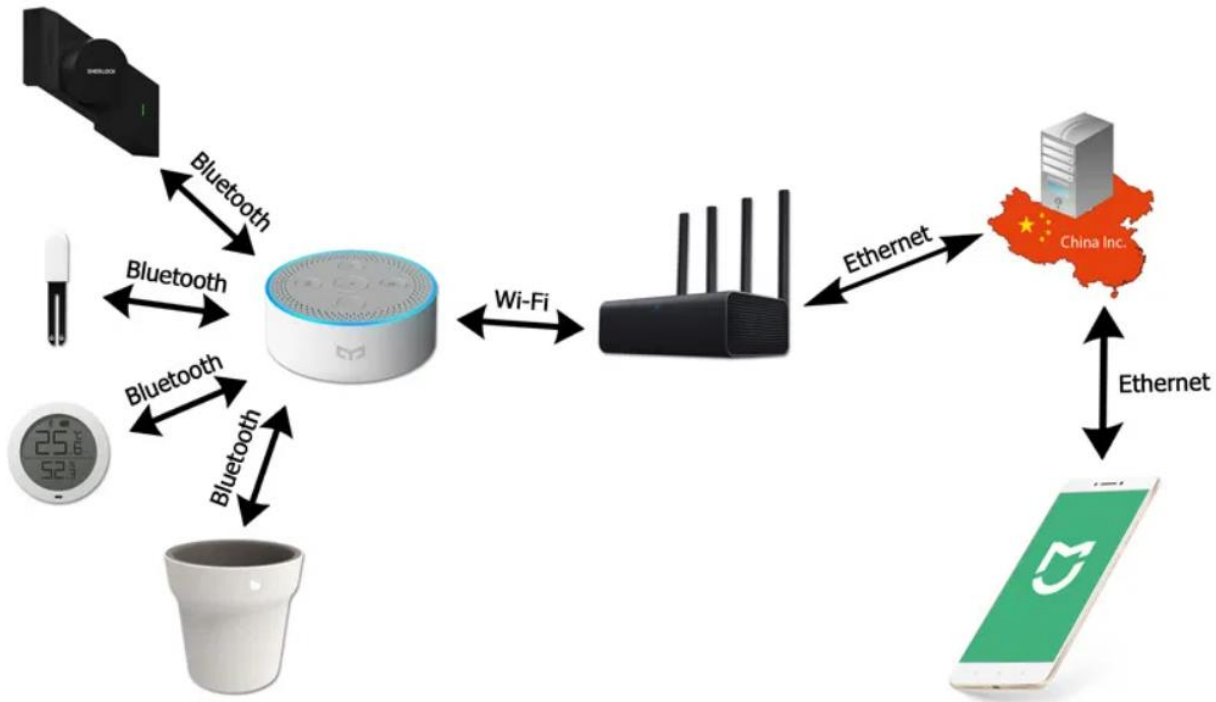
- Пристрої. Тут можливий перегляд даних про підключених до шлюзу приладах.
- Магазин. Через це додаток можлива покупка нового устаткування для «розумного будинку».
- Профіль. Тут можна вибирати аккаунт, переглядати офіційні повідомлення компанії, здійснювати моніторинг підключених приладів [31].

### 3.3.2. Bluetooth протокол

В системі «розумного будинку» Xiaomi використовуються 3 протоколи бездротового зв'язку Wi-Fi, ZigBee і Bluetooth. Через його невеликий радіус дії пристрою з Bluetooth бачаться тільки в безпосередній близькості зі смартфоном [23].

Щоб вирішити цю проблему, в 2017 році було введено поняття Bluetooth-шлюз.

Принцип роботи аналогічний ZigBee шлюзів, датчик з модулем Bluetooth підключається до одного з пристроїв, яке має одночасно 2 бездротових інтерфейсу, по Bluetooth тримає зв'язок з датчиком, а по Wi-Fi обмінюється даними з хмарою. Схематично це виглядає так [23]:



*Рис. 3.7. Схема роботи Xiaomi Smart Home за допомогою Bluetooth-шлюза*

Така схема роботи (рис. 3.7) дозволяє забезпечити тривалу роботу розумних пристроїв від батарейки і, на відміну від пристроїв з ZigBee протоколом, можливість підключення до пристрою безпосередньо [23].

### 3.3.3. ZigBee протокол в системі "розумний будинок" Хіаомі

Напевно, всі уявляють що таке Wi-Fi мережа, яка є майже в кожній квартирі. Є пристрій передавач (роутер) і приймачі (телефон, ноутбук та інші пристрої), їх пов'язує єдина мережа, яка створюється на рівні роутера. Схематично це виглядає так (рис 3.9) [24]:



*Рис. 3.8. Схема Wi-Fi мережі*

Такий тип мережі називається "Зірка", через візуальної подібності (Центральний пристрій і промені відходить від нього). Тут все здається досить просто [24].

Якщо звернути увагу на схему, то можна помітити, що всі пристрої або підключаються до розетки, або вимагають частоті зарядки. Тут у всій красі постає головний мінус Wi-Fi мережі, а саме високе енергоспоживання. І якщо для пристроїв, які живляться від розетки це не страшно, то для пристроїв і датчиків, які працюють від батарейки – це смертельно [24].

Але ж є протокол передачі даних, з низьким енергоспоживанням, який нам давно відомий і досі активно використовується. Згадайте фразу: "включи Bluetooth, я тобі перекину фото.". Саме Bluetooth, який раніше використовувався для передачі даних, а зараз зайняв нішу підключення бездротових аудіо-пристроїв і гаджетів, так само застосовується в системі "Розумний будинок" [24].

Але і тут все не так гладко, як хотілося б. Як і Wi-Fi, Bluetooth використовує топологію "Зірка". А що в цьому страшного? Відповідь проста, мала дальність дії і низька відмовостійкість. Головний блок необхідно розміщувати максимально по центру приміщення, щоб забезпечити зв'язок з усіма пристроями. І якщо для квартири це не так критично, то в приватному будинку просто не вийде забезпечити постійний зв'язок між пристроями, що знаходяться на помітному відстані один від одного. Для вирішення цієї проблеми були впроваджені Bluetooth шлюзи. І поступово розвивається система Mesh, для якої випустили спеціальні лампочки [24].

Логічно, що потрібен протокол передачі даних з низьким енергоспоживанням і надійністю з'єднання. Тому ще в 2003 році була ратифікована специфікація ZigBee. Основна відмінність від Bluetooth - це топологія "Mesh".

У ній мається на увазі наявність 3 типів пристроїв [24]:

- Координатор – головний пристрій, яке створює мережу. У розумному будинку Xiaomi цю роль виконує один з шлюзів;
- Роутер – пристрій, який має постійне харчування. Цю роль беруть на себе вимикачі і розетки;

- Кінцеві пристрої – датчики і гаджети працюють від батарейки.

На схемі (рис 3.10) розумного будинку Xiaomi протокол буде виглядати так:

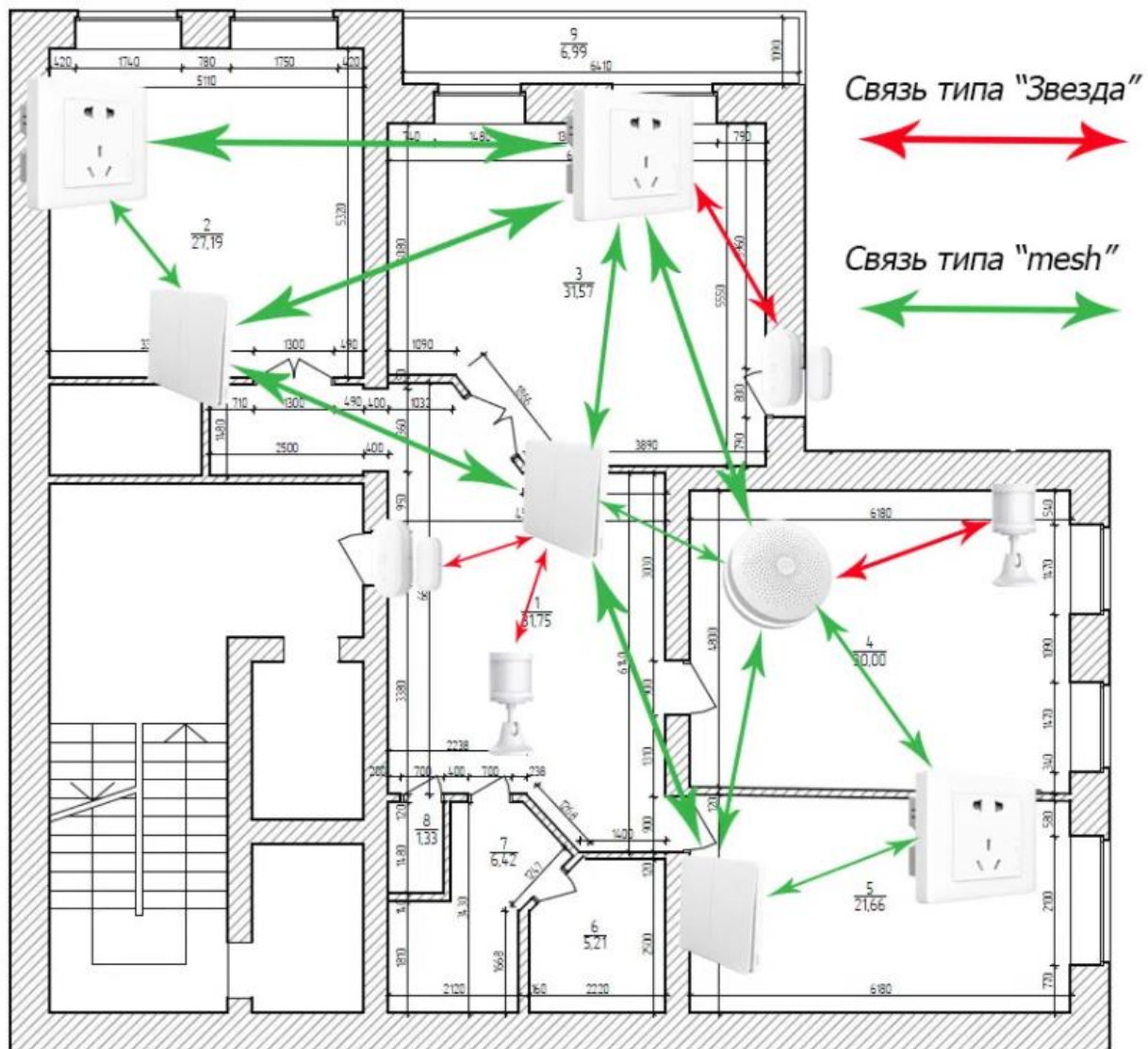


Рис. 3.9. Схема «розумного будинку» Xiaomi з протоколом ZigBee

На схемі є шлюз Gateway 2, він виконує роль координатора. До нього, в додатку Mi Home ми підключаємо всі інші пристрої. Як в Wi-Fi і Bluetooth, тільки стрілок тут помітно більше. Все це тому, що розетки і вимикачі виконують роль "Роутера", вони приймають сигнал від головного пристрою і транслюють його далі по мережі до кінцевих датчиків. Це дозволяє масштабувати мережі до рівня будинку або навіть великого виробничої будівлі. А вихід з ладу одного з пристроїв-роутер чи не позначиться на працездатності всієї мережі [24].

Хочу звернути увагу на те, що, якщо у вашій системі 2 шлюза, то це буде 2 різних, незалежних ZigBee мережі тільки в разі недосяжності. А так другий і наступний шлюз підхоплює мережа першого. Але якщо раптом один з шлюзів помре, то датчики можна перепідключити до іншого [24].

#### 3.3.4. Сценарії автоматизації

Сценарії автоматизації служать для автоматичного запуску тих чи інших дій.

Тригером для запуску сценаріїв тут може бути як пристрій (датчик, кнопка, пульт), так і різні зовнішні чинники (погода на вулиці, час світанку/заходу, час доби і день тижня). Сценарії можна налаштувати так, щоб вони запускалися тільки в конкретний проміжок часу і в певні дні тижня. Наприклад, в денний час датчик руху буде включати лампу в коридорі на всю потужність, а в нічний - на мінімум [21].

Найголовніше, що сценарії тут теж можуть взаємодіяти. Системі можна задати запуск одного сценарію тільки тоді, коли спрацював інший. Наприклад з лампою в коридорі: датчик руху буде включати її, тільки поки працює сценарій «хтось є вдома», який базується на свідченнях «розумного» замку і датчика відкриття дверей [21].

#### 3.3.5. Віддалене управління

Віддалене управління тут реалізовано відмінно – управляти своїми пристроями і змінювати сценарії можна, навіть перебуваючи на іншому кінці світу. Але з цим пов'язаний і великий мінус системи Xiaomi – навіть перебуваючи в одній мережі Wi-Fi з пристроями «розумного будинку», ви не зможете управляти ними без Інтернету. Уся взаємодія відбувається через віддалені сервери, як правило, китайські. У додатку можна вибрати і інші офіційні сервери, але кількість підтримуваних пристроїв на них вкрай мало. Тому більшості користувачів доводиться вибирати сервер саме в Китаї.

В системі передбачений тільки один тип офлайн-взаємодії. Це «локальні» сценарії (LAN scenes), які зберігаються в самому шлюзі. Сценарії в Mi Home вважаються локальними, якщо задіяні пристрої зв'язуються зі шлюзом по протоколу Zigbee. Наприклад, кімнатний вимикач і люстра при такому сполученні

будуть взаємодіяти навіть без мережі, але ось налаштувати їх уже не вийде. Єдиний варіант не залежати від віддалених серверів – підняти свій домашній сервер і скористатися «альтернативними» системами автоматизації [21].

### 3.3.6. Переваги та недоліки системи

Система «розумний будинок» від Xiaomi має цілу низку переваг, але їй також властивий і ряд технічних недоліків [22].

Плюси:

- Тривала автономність роботи завдяки низькому рівню енергоспоживання.
- Велика кількість всіляких приладів, інтегрованих з системою управління через ZigBee.
- Зручне управління за допомогою мобільних додатків з віддалених пристроїв.
- Велика кількість параметрів, що сценаріїв для різних приладів.
- Доступна вартість базової комплектації «шлюз + ряд найнеобхідніших пристроїв».

Недоліки:

- Можливі проблеми з локалізацією додатків для російського користувача.
- Відносно невелика дальність дії центрального хаба.
- Невелика кількість приладів в базовому наборі Smart Kit Home.

### 3.4. Висновок до третього розділу

Розумний будинок Apple – безпроблемна та легка в управлінні система. Його головний мінус — ціна. Гаджети від Xiaomi, об'єднані в Mi Home, можна вважати повноцінною заміною яблучному смарт-дому. В арсеналі компанії величезна кількість смарт-пристроїв, число їх постійно зростає, а ціни виглядають набагато привабливіше.

При виборі Xiaomi варто звертати увагу на протоколи. У більшості випадків не обійтися без Hub, якщо чайник зав'язаний на Bluetooth, датчики використовують ZigBee, а лампи підкоряються командам через Wi-Fi. Попри подібні складнощі, у гаджетів Xiaomi дві переваги – ціна та непоганий дизайн, що нагадує класичні продукти конкурента [23].

Розумний будинок SmartThings – це широкомасштабний автоматизований комплекс, який являє собою центральний Hub, а також цілий ряд камер і датчиків. Компанія Samsung зробила революційний прорив, створивши відкриту екосистему «розумних речей», в якій об'єднуються всі можливі пристрої, оснащені інтелектом.

Ці три системи універсальні та годяться для вирішення будь-яких завдань, від контролю освітлення до управління мікрокліматом. Голосові помічники Siri та Google Assistant не працюють без інтернету, а зв'язок із зовнішніми пристроями здійснюється тільки по Wi-Fi. Хмарні рішення — не найкращий варіант в плані безпеки, тому за правилами Apple у всіх HomeKit-сумісних пристроях використовується принцип повного шифрування (end-to-end), підхід правильний з точки зору надійності та захисту даних, але збільшує кінцеву вартість продукту.

Додатки розумного будинку Xiaomi, SmartThings та Apple відрізняються по можливості використання на різних пристроях. Так, HomeKit неможливо встановити на андроїд-смартфони, тільки на рідні яблучні гаджети; тоді як додаток Mi Home та SmartThings може функціонувати в обох системах [25].

Таблиця 3.

## Порівняльна таблиця

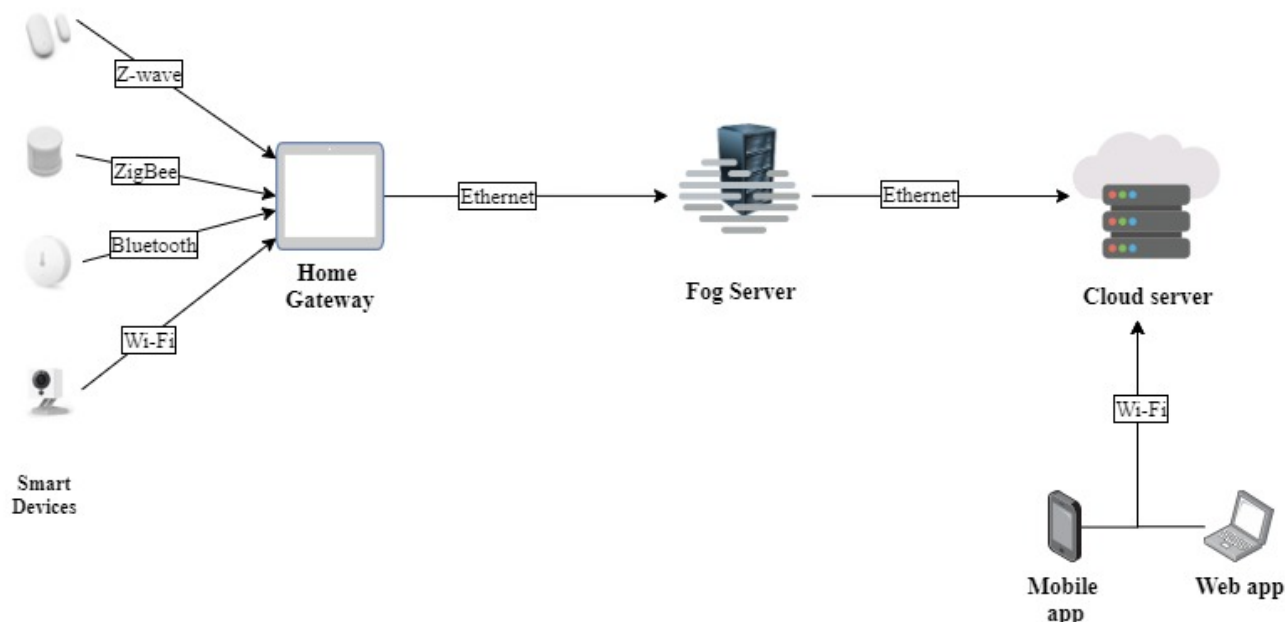
|                                              | Apple Home Kit          | Samsung SmartThings | Xiaomi Smart Home |
|----------------------------------------------|-------------------------|---------------------|-------------------|
|                                              | Підтримувані технології |                     |                   |
| Z-Wave                                       | ✓                       | ✓                   | ✗                 |
| Zigbee                                       | ✓                       | ✓                   | ✓                 |
| Wi-Fi                                        | ✓                       | ✓                   | ✓                 |
| Bluetooth Low Energy                         | ✓                       | ✗                   | ✓                 |
| 868 MHz                                      | ✓                       | ✗                   | ✗                 |
| 433 MHz – Nexa/Telldus                       | ✓                       | ✗                   | ✗                 |
| 433MHz – Somfy                               | ✓                       | ✗                   | ✗                 |
| Infrared                                     | ✗                       | ✗                   | ✓                 |
|                                              | Способи підключення     |                     |                   |
| Hub-connected<br>(Підключення через шлюз)    | ✓                       | ✓                   | ✓                 |
| Directly-connected<br>(Пряме підключення)    | ✓                       | ✓                   | ✓                 |
| Cloud-connected<br>(Підключення через хмару) | ✗                       | ✓                   | ✗                 |
|                                              | Елементи                |                     |                   |
| Smart devices<br>(Розумні пристрої)          | ✓                       | ✓                   | ✓                 |
| Hub<br>(Шлюз)                                | ✓                       | ✓                   | ✓                 |
| Router<br>(Роутер)                           | ✓                       | ✓                   | ✓                 |
| Cloud<br>(Хмара)                             | ✓                       | ✓                   | ✓                 |
|                                              |                         |                     |                   |
| Голосові помічники                           | ✓                       | ✓                   | ✓                 |
| Мобільний додаток                            | ✓                       | ✓                   | ✓                 |
| Повідомлення про конфіденційність            | ✓                       | ✗                   | ✗                 |

#### 4. МОДИФІКАЦІЯ АРХІТЕКТУРНОЇ КОНЦЕПЦІЇ РОЗУМНА БУДІВЛЯ

В даному розділі я пропоную модифіковану архітектуру розумної будівлі.

##### 4.1. Архітектура розумної будівлі

На рис 4.1, я представила загальну архітектуру та взаємодію між пристроями.



*Рис. 4. Загальна архітектура та взаємодія між пристроями*

Уся робота побудована навколо головного контролера – Home Gateway. Саме Home Gateway підтримує бездротову мережу, яка потрібна для підключення і роботи з усіма приладами «Розумного будинку». Він виступає в ролі своєрідного роутера.

Home Gateway забезпечений потужним процесором. В умовах відключення від електричної мережі його батарея забезпечує до десяти годин безперервної роботи. Цього достатньо для підтримки системи «Розумний дім» без збоїв та в разі відключення електроенергії в будинку або квартирі. При цьому сполучення з усіма пристроями і контролером не буде втрачено і робота системи буде відбуватися за звичним сценарієм. Для таких випадків Hub використовує інтерфейс Zigbee або Z-wave.

Далі дані передаються до fog server. Fog server знаходиться недалеко від розумного будинку. Даний сервер потрібен для того, щоб обробляти дані, які не потребують великих обчислювальних процесів. Fog server забезпечує уніфікований

інтерфейс для інтеграції всіх різних незалежних пристроїв та надає інтелектуальним програмам гнучкі ресурси, що дозволяє зберігати та обробляти дані в реальному часі та має малу затримку передачі даних.

Потім, дані передаються до cloud server. Cloud server відповідає за вирішення завдань, що вимагають великого обсягу обчислювальних ресурсів або є місцем для зберігання даних.

Більш детально, я представила, блоки системи та спосіб їх з'єднання для збору, зберігання та обробки даних на рис. 4.1.

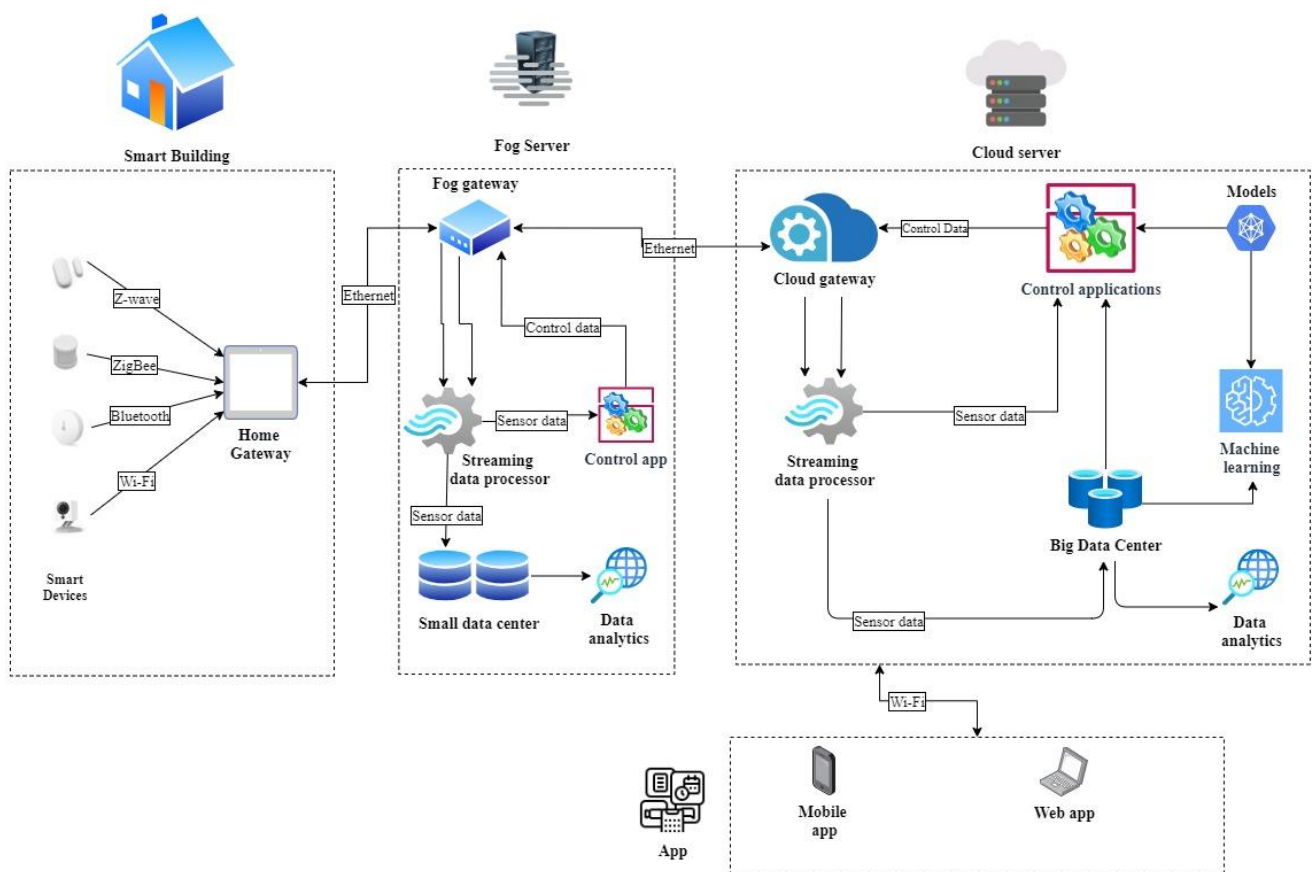


Рис. 4.1. Архітектура «Розумного будинку»

**Smart devices.** Smart device – це об'єкт, оснащений датчиками, які збирають дані, які будуть передаватися через мережу, і виконавчими механізмами, що дозволяють речам діяти. Наприклад, увімкнути або вимкнути світло, відкрити або закрити двері, збільшити або зменшити швидкість обертання двигуна та інше. Ця концепція включає холодильники, вуличні ліхтарі, будівлі, транспортні засоби, виробниче обладнання, реабілітаційне обладнання та все інше, що можна собі

уявити. Датчики не у всіх випадках фізично прикріплені до речей: сенсорам може знадобитися контролювати, наприклад, що відбувається в найближчому до речі середовищі.

**Gateway.** Дані надходять від речей до туману і навпаки через шлюзи. Шлюз забезпечує зв'язок між речами та туманною частиною рішення IoT, забезпечує попередню обробку та фільтрацію даних перед переміщенням їх у туман, щоб зменшити обсяг даних для детальної обробки та зберігання і передає команди управління, що йдуть із хмари до речей. Потім розумні речі виконують команди за допомогою своїх виконавчих механізмів.

**Fog gateway** полегшує стиснення даних та безпечну передачу даних між home gateway та хмарними серверами IoT. Він забезпечує сумісність з різними протоколами та взаємодіє з home gateway, використовуючи різні протоколи, залежно від того, який протокол підтримується шлюзами.

**Streaming data processor (Потоковий процесор даних)** - забезпечує ефективний перехід вхідних даних в small/big data center та control applications. Жодні дані не можуть бути втрачені або пошкоджені.

**Small data center** - використовується для зберігання та попередньої обробки даних. Саме тут оброблюється критично важлива для часу дані, як стан пристрою, попередження про несправність, стан тривоги та багато іншого, щоб мінімізувати затримку, покращити ефективність та запобігти великим пошкодженням

**Cloud gateway** полегшує стиснення даних та безпечну передачу даних між fog gateway та хмарними серверами IoT.

**Big data center** - використовується для зберігання вже відфільтрованих та попередньо оброблених даних, які необхідні для значущих уявлень. Він містить лише очищені, структуровані та зіставлені дані. Крім того, зберігає контекстну інформацію про пристрої (де він встановлений), та команди, які control applications надсилає пристроям.

**Data analytics.** Аналітики даних можуть використовувати дані з data center, щоб знаходити тенденції та отримувати практичну інформацію. При аналізі великі дані показують, продуктивність пристроїв, допомагають виявити неефективність

та виробити шляхи вдосконалення системи IoT тобто зробити її більш надійною та орієнтованою. Крім того, кореляції та шаблони, знайдені вручну, можуть додатково сприяти створенню алгоритмів для програм управління.

**Машинне навчання та моделі, які генерує МН.** Завдяки машинному навчанню є можливість створювати більш точні та ефективніші моделі для програм управління. Моделі регулярно оновлюються на основі історичних даних, накопичених у data center. Коли застосовність та ефективність нових моделей перевіряються та схвалюються аналітиками даних, нові моделі використовуються програмами управління.

Control applications надсилають автоматичні команди та попередження на виконавчі механізми:

- вікна в розумному будинку можуть отримати автоматичну команду відкриватись або закриватися залежно від прогнозів, взятих з метеорологічної служби;
- коли датчики показують, що ґрунт сухий, системи поливу отримують автоматичну команду поливати рослини.

Команди, що надсилаються програмами керування на виконавчі механізми, також можуть бути додатково збережені у big data center. Це може допомогти розслідувати проблемні випадки.

Наприклад, програма управління надсилає команди, але вони не виконуються виконавчими механізмами – тоді потрібно перевірити підключення, шлюзи та виконавчі механізми.

З іншого боку, зберігання команд з програм керування може сприяти безпеці, оскільки система IoT може виявити, що деякі команди занадто дивні або надходять у занадто великих кількостях, що може свідчити про порушення безпеки, а також про інші проблеми, які потребують розслідування та коригувальних заходів.

Програми керування можуть бути як на основі правил, так і на основі машинного навчання. У першому випадку контрольні програми працюють за правилами, визначеними фахівцями. У другому випадку контрольні програми

використовують моделі, які регулярно оновлюються, а історичні дані зберігаються у big data center.

Хоча додатки управління забезпечують кращу автоматизацію системи IoT, у користувачів завжди повинна бути можливість впливати на поведінку таких програм. Наприклад, у надзвичайних ситуаціях або коли виявляється, що система IoT погано налаштована на виконання певних дій.

Applications – це програмний компонент системи IoT, який дозволяє підключати користувачів до системи IoT і дає можливість контролювати та контролювати їхні розумні речі. За допомогою мобільного або веб-додатку користувачі можуть стежити за станом своїх речей, надсилати команди, встановлювати параметри автоматичної поведінки.

#### 4.1.1. Control applications

Щоб забезпечити достатнє функціонування пристроїв IoT, недостатньо встановити їх і дозволити справі йти своїм шляхом. Існує кілька процедур, необхідних для управління продуктивністю підключених пристроїв, тобто полегшити взаємодію між пристроями, забезпечити безпечну передачу даних тощо:

- ідентифікація пристрою, щоб встановити особу пристрою, щоб переконатися, що це справжній пристрій із надійним програмним забезпеченням, що передає надійні дані;
- конфігурація та управління для налаштування пристроїв відповідно до цілей системи IoT. Деякі параметри потрібно писати після встановлення пристрою, такий як унікальний ідентифікатор пристрою. Інші налаштування можуть потребувати оновлення. Наприклад, час між надсиланням повідомлень із даними;
- моніторинг та діагностика для забезпечення безперебійної та безпечної роботи кожного пристрою в мережі та зменшення ризику поломок;
- оновлення програмного забезпечення та технічне обслуговування для додавання функціональних можливостей, виправлення помилок, усунення вразливостей системи безпеки.

#### 4.1.2. Управління користувачами

Поряд із керуванням пристроями важливо забезпечити контроль над користувачами, які мають доступ до системи IoT.

Управління користувачами включає визначення користувачів, їх ролей, рівнів доступу та власності в системі. Він включає такі опції, як додавання та видалення користувачів, управління налаштуваннями користувачів, контроль доступу різних користувачів до певної інформації, а також дозвіл на виконання певних операцій у системі, контроль та запис дій користувача тощо.

#### 4.1.3. Моніторинг безпеки

Безпека є однією з головних проблем в Інтернеті речей. Пов'язані речі дають величезні обсяги даних, які потрібно надійно передавати та захищати від кіберзлочинців. Інша сторона полягає в тому, що речі, підключені до Інтернету, можуть бути точками входу для лиходіїв. Більше того, кіберзлочинці можуть отримати доступ до "мозку" всієї системи IoT та взяти під контроль.

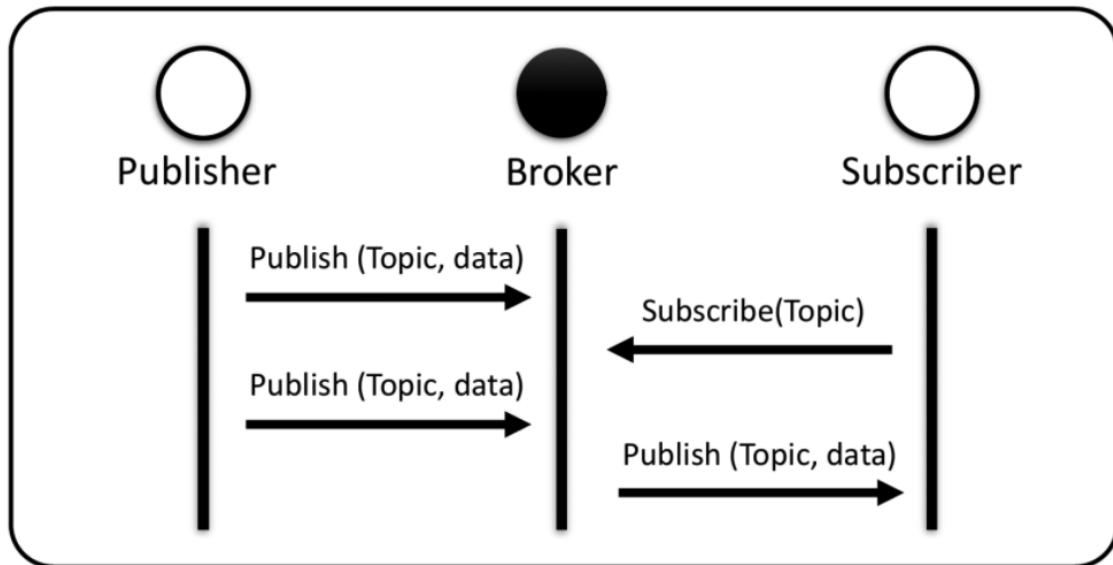
Для запобігання подібним проблемам має сенс реєструвати та аналізувати команди, що надсилаються програмами керування речам, контролювати дії користувачів та зберігати всі ці дані у хмарі. Завдяки такому підходу можна усунути порушення безпеки на ранніх стадіях і вжити заходів для зменшення їх впливу на систему IoT. Наприклад, блокувати певні команди, що надходять із програм управління.

Крім того, можна виявити зразки підозрілої поведінки, зберегти ці зразки та порівняти їх із журналами, створеними системами IoT, щоб запобігти потенційному проникненню та мінімізувати їх вплив на систему IoT [26].

#### 4.2. Протоколи зв'язку

Існує 2 моделі протоколів зв'язку, що підходять для даної системи. Це запит-відповідь і публікація-підписка (рис. 4.2). Перша модель відома ширше, особливо в архітектурі сервер-клієнт. Клієнт запитує інформацію з сервера, а той отримує запит, обробляє його і повертає відповідь. За цією моделлю працюють протоколи REST HTTP і CoAP.

Друга модель виникла через необхідність забезпечити асинхронний, розподілений, слабкий зв'язок між джерелами, генеруючими дані, і одержувачами цих даних.



*Рис. 4.2. Модель публікація-підписка*

Модель передбачає трьох учасників:

- видавець (джерело даних);
- брокер (диспетчер);
- передплатник (одержувач).

Тут клієнт, який виступає в ролі передплатника, не повинен запитувати інформацію з сервера. Замість відправки запитів він підписується на певні події в системі через брокера, відповідального за фільтрацію всіх вхідних повідомлень і їх маршрутизацію між видавцями і передплатниками. А видавець, коли відбувається подія, що стосується певної теми, публікує його брокеру, який відправляє передплатнику дані по запитаній темі.

Така модель взаємодії цікава для додатків в IoT, хмарі, тумані через її здатність забезпечувати масштабованість і спрощувати взаємозв'язок між різними пристроями, підтримувати динамічний зв'язок «багато до багатьох» і асинхронний зв'язок. Серед найбільш відомих стандартизованих протоколів обміну повідомленнями, що використовують модель «публікація-підписка», можна назвати MQTT, AMQP і DDS.

#### 4.2.1. Використання багато протокольного рішення

Також я пропоную в даній архітектурі використовувати багато протокольне рішення, тому що рішення з одним протоколом більш легкі з реалізацію. Але очевидно, що в системах IoT-F2C має сенс комбінувати різні протоколи. Сенс в тому, що на різних рівнях можуть працювати різні протоколи.

Наприклад візьме, три абстракції: рівні IoT, туману і хмарних обчислень (рис. 4.3). Пристрої на рівні IoT зазвичай вважаються обмеженими. Для цього огляду давайте розглянемо рівні IoT як найобмеженіші, хмарні найменш обмежені і обчислення туману як «десь посередині». Тоді виходить, що між IoT і абстракціями туману поточні протокольні рішення включають в себе MQTT, CoAP і XMPP. Між туманом і хмарию, з іншого боку, AMQP є одним з основних використовуваних протоколів разом з REST HTTP, який завдяки своїй гнучкості також використовується між IoT і шарами туману.

Основною проблемою тут виступає функціональна сумісність протоколів і простота перекладу повідомлень з одного протоколу в інший. В ідеалі, в майбутньому архітектура системи інтернету речей з хмарними і туманними ресурсами буде незалежною від використовуваного протоколу зв'язку і забезпечить хорошу взаємодію різних протоколів.

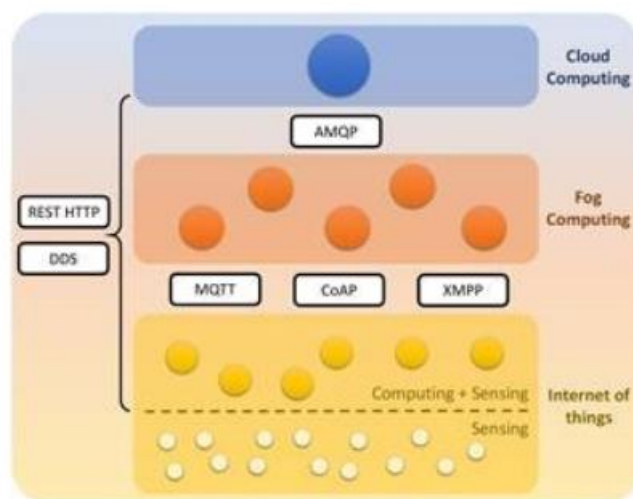


Рис. 4.3. Використання протоколів між трьома абстракціями

Оскільки на даний момент це не так, має сенс поєднувати протоколи, які не мають значних відмінностей. З цією метою одне потенційне рішення засноване на

комбінації двох протоколів, які поділяють таку ж архітектурного стилю, REST HTTP і CoAP. Інше запропоноване рішення засноване на поєднанні двох протоколів, які пропонують взаємодію по моделі «публікація-підписка», MQTT і AMQP. Використання близьких концепцій, спрощує реалізацію цих комбінацій і вимагає менших зусиль по інтеграції.

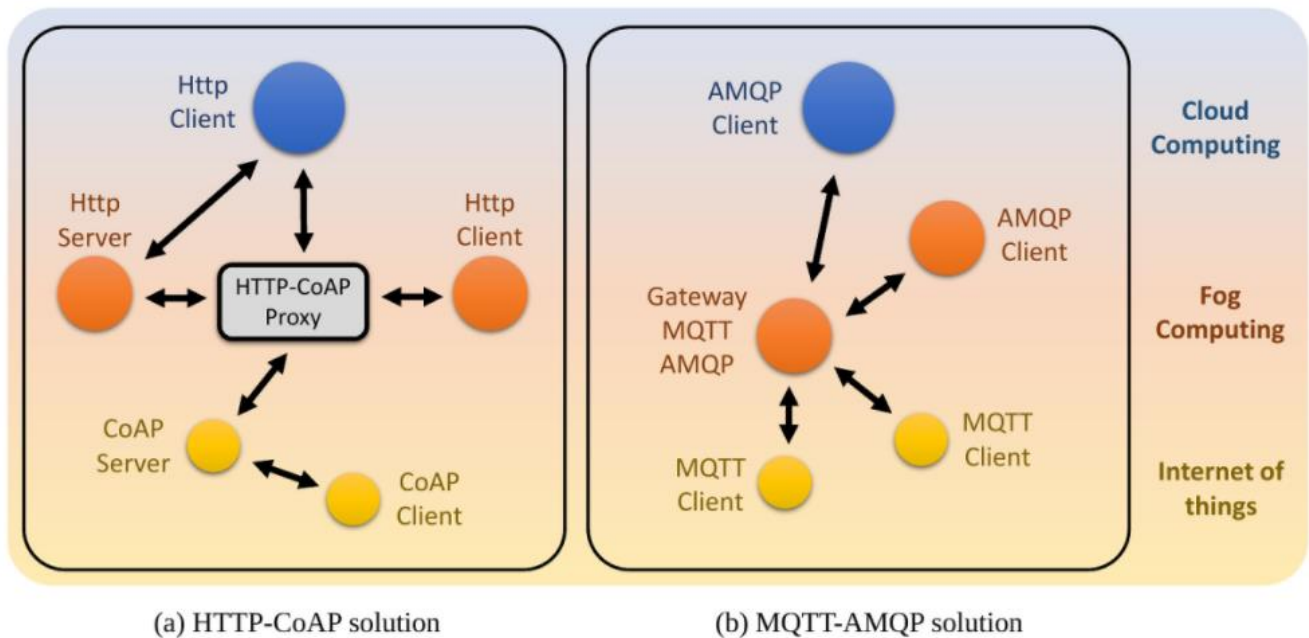


Рис. 4.4. Два рішення взаємодії протоколів зв'язку

На рис. 4.4 (а) показані дві моделі на основі запитів-відповідей, HTTP і CoAP, і їх можливе розміщення в рішенні IoT-F2C. Оскільки HTTP є одним з найбільш відомих і адаптованих протоколів в сучасних мережах, малоймовірно, що він буде повністю замінений іншими протоколами обміну повідомленнями. Серед вузлів, що представляють потужні пристрої, які знаходяться між хмарою і туманом, REST HTTP є розумним рішенням.

З іншого боку, для пристроїв з обмеженими обчислювальними ресурсами, які зв'язуються між рівнями туману і IoT, більш ефективно використовувати CoAP. Одним з великих переваг CoAP насправді є його сумісність з HTTP, так як обидва протоколи засновані на принципах REST.

На рис. 4.4 (б) показані дві моделі взаємодії «публікація-підписка» в одному сценарії, включаючи MQTT і AMQP. Хоча гіпотетично обидва протоколи можуть використовуватися для зв'язку між вузлами на кожному рівні абстракції, їх

положення повинно визначатися на основі продуктивності. MQTT був розроблений як спрощений протокол для пристроїв з обмеженими обчислювальними ресурсами, тому його можна використовувати для зв'язку між IoT і туманом. AMQP більше підходить для більш потужних пристроїв, які ідеально розташували б його між вузлами туману і хмари. Замість MQTT в IoT можна використовувати протокол XMPP, так як він вважається легковажним. Але він не так широко використовується в подібних сценаріях [27].

#### Висновок до 4 розділу

Запропонована мною в роботі архітектура IoT містить такі компоненти:

- Smart devices, оснащені датчиками для збору даних та виконавчими механізмами для виконання команд, отриманих із хмари;
- Gateway (шлюз) для фільтрації даних, попередньої обробки та переміщення їх у хмару (туман) і навпаки, - отримання команд із хмари (туману);
- Fog gateway – забезпечення переходу даних між home gateway та центральними серверами IoT;
- Small data center – зберігання та попередньої обробки даних;
- Streaming data processor (Потокові процесори даних) для розподілу даних, що надходять від датчиків, між відповідними компонентами IoT-рішення;
- Cloud gateway для забезпечення переходу даних fog gateway та центральними серверами IoT;
- Big data center – для збору цінних даних;
- Control applications (управління додатками) для надсилення команд на виконавчі механізми;
- Машинне навчання для генерації моделей, які потім використовуються програмами управління;
- Applications (додатки), що дозволяють користувачам контролювати контроль своїх підключених речей;
- Аналіз даних для ручної обробки даних.

Також вона містить компоненти управління пристроями та користувачами, щоб забезпечити стабільне та безпечне функціонування речей та контролювати проблеми доступу користувачів.

В таблиці 4 представлена порівняльна характеристика Apple HomeKit, Samsung SmartThings та Xiaomi Smart Home з модифікованою архітектурою.

Таблиця 4.

*Порівняння модифікованої архітектури*

|                                              | Модифікована<br>архітектура | Apple Home<br>Kit | Samsung<br>SmartThings | Xiaomi<br>Smart Home |
|----------------------------------------------|-----------------------------|-------------------|------------------------|----------------------|
|                                              | Елементи                    |                   |                        |                      |
| Smart devices (Розумні пристрої)             | ✓                           | ✓                 | ✓                      | ✓                    |
| Hub (Шлюз)                                   | ✓                           | ✓                 | ✓                      | ✓                    |
| Router (Роутер)                              | ✓                           | ✓                 | ✓                      | ✓                    |
| Fog (Туман)                                  | ✓                           | ✗                 | ✗                      | ✗                    |
| Cloud (Хмара)                                | ✓                           | ✓                 | ✓                      | ✓                    |
|                                              | Способи підключення         |                   |                        |                      |
| Hub-connected<br>(Підключення через шлюз)    | ✓                           | ✓                 | ✓                      | ✓                    |
| Directly-connected<br>(Пряме підключення)    | ✗                           | ✓                 | ✓                      | ✓                    |
| Cloud-connected<br>(Підключення через хмару) | ✗                           | ✗                 | ✓                      | ✗                    |
|                                              | Підтримувані технології     |                   |                        |                      |
| Z-Wave                                       | ✓                           | ✓                 | ✓                      | ✗                    |
| Zigbee                                       | ✓                           | ✓                 | ✓                      | ✓                    |
| Wi-Fi                                        | ✓                           | ✓                 | ✓                      | ✓                    |
| BLE                                          | ✓                           | ✓                 | ✗                      | ✓                    |
| 868 MHz                                      | ✗                           | ✓                 | ✗                      | ✗                    |
| 433 MHz –<br>Nexa/Telldus                    | ✗                           | ✓                 | ✗                      | ✗                    |
| 433MHz – Somfy                               | ✗                           | ✓                 | ✗                      | ✗                    |
| Infrared                                     | ✗                           | ✗                 | ✗                      | ✓                    |

## 5. СТАРТАП ПРОЕКТ

### 5.1. Опис ідеї проекту

Таблиця 5.1.

#### Опис ідеї проекту

| Зміст ідеї                                                  | Напрямки застосування      | Вигоди для користувача                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Модифікування архітектурної концепції Smart Building</b> | 1. Безпека                 | Передача сигналу тривоги в реальному часі та з малою затримкою, виступає як проксі-сервер для обмежених ресурсів пристроїв для оновлення програмного забезпечення або облікових даних безпеки цих пристроїв. Вони також можуть контролювати стан безпеки прилеглих пристроїв.      |
|                                                             | 2. Розумна будівля         | Забезпечує уніфікований інтерфейс для інтеграції всіх різних незалежних пристроїв і надає інтелектуальним домашнім програмам гнучкі ресурси, що дозволяє зберігати дані, обробляти в режимі реального часу та мати малу затримку.                                                  |
|                                                             | 3. Постійні послуги        | Можуть працювати незалежно та забезпечувати безперебійне обслуговування навіть за умов коливання мережевого підключення до хмари                                                                                                                                                   |
|                                                             | 4. Галузь Охорони здоров'я | Забезпечує обробку даних в реальному часі та відповіді на події, що є критично важливими для охорони здоров'я. Крім того, він також вирішує питання щодо мережевого підключення та трафіку, необхідних для віддаленого зберігання, обробки та отримання медичних записів із хмари. |
|                                                             | 5. Мережа                  | Дозволяє обробляти дані на основі вимог програми, наявних мережевих та обчислювальних ресурсів. Це зменшує обсяг даних, необхідних для передачі в хмару, в кінцевому рахунку економить пропускну здатність мережі.                                                                 |
|                                                             | 6. Додатки                 | Виконує всі чутливі до часу дії, близькі до кінцевих користувачів, що відповідає обмеженням затримки додатків IoT.                                                                                                                                                                 |

Таблиця 5.2.

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

| №<br>п/п | Техніко-<br>економічні<br>характерист<br>ики ідеї                  | (потенційні) товари/концепції конкурентів |                            |                   |                         | W | N | S |
|----------|--------------------------------------------------------------------|-------------------------------------------|----------------------------|-------------------|-------------------------|---|---|---|
|          |                                                                    | Мій<br>проект                             | Samsung<br>SmartThin<br>gs | Apple<br>Home Kit | Xiaomi<br>Smart<br>Home |   |   |   |
| 1.       | Пропускну<br>здатність                                             | економія                                  | -                          | -                 | -                       |   |   | + |
| 2.       | Зберігати та<br>обробляти<br>дані в<br>режимі<br>реального<br>часу | Fog server<br>Cloud server                | Cloud<br>server            | Cloud<br>server   | Cloud<br>server         |   |   | + |
| 3.       | Час<br>затримки                                                    | мінімальний                               | середній                   | середній          | середній                |   |   | + |
| 4.       | Масштабува<br>ння                                                  | +                                         | +                          | +                 | +                       |   |   | + |
| 5.       | Моніторинг<br>безпеки                                              | +                                         | -                          | -                 | -                       |   |   | + |
| 6.       | Багато<br>протокольне<br>рішення<br>зв'язку                        | +                                         | +                          | +                 | +                       |   |   | + |
| 7.       | Задоволеніс<br>ть покупця                                          | висока                                    | середня                    | індивідуа<br>льна | висока                  |   |   | + |
| 8.       | Інтеграція з<br>корпоратив<br>ними<br>системами                    | +                                         | -                          | +                 | -                       |   | + |   |

## 5.2. Технологічний аудит ідеї проекту

В даному розділі проведемо аудит технології, за допомогою якої можна реалізувати ідею проекту [28].

Для визначення технологічної здійсненності ідеї проекту проведемо аналіз таких складових (табл. 3):

- за якою технологією буде виготовлено товар згідно ідеї проекту?
- чи існують такі технології, чи їх потрібно розробити/доробити?
- чи доступні такі технології авторам проекту?

Таблиця 5.3.

Технологічна здійсненність ідеї проекту

| №<br>n/n                                                                                  | Ідея проекту                       | Технології її реалізації                                                                                                      | Наявність технологій | Доступність технологій |
|-------------------------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|----------------------|------------------------|
| 1.                                                                                        | Багато протокольне рішення зв'язку | Використання таких протоколів, як MQTT, CoAP, XMPP, AMQP, REST HTTP                                                           | Наявні               | Доступні               |
| 2.                                                                                        | Cloud server                       | Використання AWS IoT - єдиний набір хмарних сервісів, який об'єднує управління даними і докладну аналітику в зручному рішенні | Наявні               | Доступні               |
| 3.                                                                                        | Fog server                         | Використання фізичного сервера як Fog-вузла                                                                                   | Наявні               | Доступні               |
| <b>Обрана технологія реалізації ідеї проекту: Технологічна реалізація проекту можлива</b> |                                    |                                                                                                                               |                      |                        |

### 5.3. Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

5.3.1) Спочатку проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 5.4).

Таблиця 5.4.

*Попередня характеристика потенційного ринку стартап-проекту*

| <i>№<br/>п/<br/>п</i> | <i>Показники стану ринку (найменування)</i>              | <i>Характеристика</i>                        |
|-----------------------|----------------------------------------------------------|----------------------------------------------|
| 1                     | Кількість головних гравців, од                           | Безліч                                       |
| 2                     | Загальний обсяг продаж, грн/ум. од                       | -                                            |
| 3                     | Динаміка ринку (якісна оцінка)                           | Зростає                                      |
| 4                     | Наявність обмежень для входу (вказати характер обмежень) | Необхідно запатентувати архітектурну систему |
| 5                     | Специфічні вимоги до стандартизації та сертифікації      | немає                                        |
| 6                     | Середня норма рентабельності в галузі (або по ринку), %  | 45%                                          |

5.3.2) Далі визначаємо потенційні групи клієнтів, їх характеристики, та формуємо орієнтовний перелік вимог до товару для кожної групи (табл. 5.5).

Таблиця 5.5.

*Характеристика потенційних клієнтів стартап-проекту*

| <i>№<br/>п/п</i> | <i>Потреба, що<br/>формує ринок</i>                   | <i>Цільова<br/>аудиторія<br/>(цільові<br/>сегменти<br/>ринку)</i> | <i>Відмінності у<br/>поведінці різних<br/>потенційних цільових<br/>груп клієнтів</i>                             | <i>Вимоги<br/>споживачів до<br/>товару</i>                                |
|------------------|-------------------------------------------------------|-------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| 1.               | Реагування датчиків в короткі терміни                 | 1. Власники квартир/домів                                         | Використання модифікованої архітектури в різних послугах IoT:<br>Автоматизація дома/будівлі<br>Охорона<br>Мережа | Передавати та обробляти дані в режимі реального часу та з малою затримкою |
| 2.               | Зменшити обсяг даних, необхідних для передачі в хмару | 2. Власники бізнесів                                              |                                                                                                                  | Економія пропускну здатності                                              |
| 3.               | Автономність                                          | 3. Комунальні підприємства                                        |                                                                                                                  | Працювати незалежно та забезпечувати безперебійне обслуговування          |

5.3.3) Після визначення потенційних груп клієнтів проведемо аналіз ринкового середовища. Фактори в таблицях 5.6. - 5.7. подані в порядку зменшення значущості.

Таблиця 5.6.

*Фактори загроз*

| <i>№<br/>n/n</i> | <i>Фактор</i>      | <i>Зміст загрози</i>                                            | <i>Можлива реакція компанії</i>                                                         |
|------------------|--------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| 1.               | Конкуренція        | Виникнення більш ефективної системи                             | Удосконалення системи                                                                   |
| 2.               | Якість             | Низка якість обладнання, програмного продукту та обслуговування | Пошук нового обладнання/програмного продукту або удосконалення, підвищення класифікації |
| 3.               | Функціонування     | Недостача наявного функціоналу для задоволення потреб клієнтів  | Розширення функціоналу                                                                  |
| 4.               | Зростання інфляції | Падіння платоспроможності клієнтів                              | Планування, гнучке ціноутворення                                                        |

Таблиця 5.7.

*Фактори можливостей*

| <i>№<br/>n/n</i> | <i>Фактор</i>                      | <i>Зміст можливості</i>                                                                   | <i>Можлива реакція компанії</i>                                                       |
|------------------|------------------------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 1.               | Отримання необхідних інвестицій    | Додаткове фінансування, яке допоможе долучити більше інженерів та розробників             | Удосконалення системи з найбільше розширеним функціоналом                             |
| 2.               | Освоєння нових технологій          | Використання більш ефективних технологій для покращення та розширення функціоналу системи | Створення фокус-груп інженерів та розробників, які будуть вивчати та впроваджувати їх |
| 3.               | Розширення аналітичних можливостей | Використання інформації, яка надходить з датчиків з метою удосконалення системи           | Масштабування функціональності системи                                                |
| 4.               | Успішна маркетингова політика      | Висока зацікавленість користувачів в продукті                                             | Стабільна робота та розвиток системи                                                  |
| 5.               | Зворотній зв'язок від користувачів | Отримання статистичних даних на основі яких буде здійснено вдосконалення системи          | Задоволення потреби та бажань кінцевих користувачів                                   |

5.3.4) Надалі проведемо аналіз пропозиції: визначаємо загальні риси конкуренції на ринку (табл. 5.8).

Таблиця 5.8.

*Ступеневий аналіз конкуренції на ринку*

| Особливості конкурентного середовища                  | В чому проявляється дана характеристика                                         | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
|-------------------------------------------------------|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| 1. Вказати тип конкуренції<br>- монополістична        | Безліч компаній, які пропонують свої послуги                                    | Постійний розвиток                                                                     |
| 2. За рівнем конкурентної боротьби міжнародній        | Представники в різних країнах                                                   | Вихід на міжнародний ринок                                                             |
| 3. За галузевою ознакою<br>- міжгалузева              | Можливість використання подібної архітектури для автоматизації інших сфер життя | Постійний розвиток, пошук нових галузей побуту для автоматизації                       |
| 4. Конкуренція за видами товарів:<br>- товарно-видова | Різні способи побудови системи                                                  | Оптимізація рішень                                                                     |
| 5. За характером конкурентних переваг<br>- нецінова   | Головною перевагою є Унікальність та ефективність використання                  | Постійний розвиток                                                                     |
| 6. За інтенсивністю<br>- марочна                      | -                                                                               | -                                                                                      |

5.3.5) Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера) (табл. 5.9).

Таблиця 5.9.

*Аналіз конкуренції в галузі за М. Портером*

|                  | Прямі конкуренти в галузі            | Потенційні конкуренти                                                                  | Постачальники                                                           | Клієнти                                                                            | Товари-замінники |
|------------------|--------------------------------------|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------|
| Складові аналізу | Apple<br>Xiaomi<br>Samsung<br>Google | Необхідно запатентувати модифіковану архітектуру                                       | Значення розміру поставок, кількість постачальників, обмеження ресурсів | Користувачі можуть бути незадовільні функціональністю або швидкістю передачі даних | Відсутні         |
| Висновки:        | В галузі існує висока конкуренція    | Потенційним конкурентам потрібно запропонувати щось особливе, щоб створити конкуренцію | Незначна кількість постачальників, необмежені в об'ємах                 | Клієнти вносять побажання, які варто враховувати при побудові системи              | Відсутні         |

5.3.6) На основі аналізу конкуренції, проведеного в п. 3.5 (табл. 5.9), а також із урахуванням характеристик ідеї проекту (табл. 5.2), вимог споживачів до товару (табл. 5.5) та факторів маркетингового середовища (табл. №№ 5.6-5.7) визначаємо та обґрунтовуємо перелік факторів конкурентоспроможності. Аналіз оформлений в табл. 5.10.

Таблиця 5.10.

*Обґрунтування факторів конкурентоспроможності*

| № п/п | Фактор конкурентоспроможності                 | Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)                                                       |
|-------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.    | Наявність патенту на модифіковану архітектуру | Наявність такого документу не дає можливість конкурентам використовувати розроблену архітектуру                                                           |
| 2.    | Автономність                                  | Працює незалежно та забезпечує безперебійне обслуговування                                                                                                |
| 3.    | Інтеграція з корпоративними системами         | Співпраця з охоронними службами, міськими службами, як газова служба, пожежники і т.д.                                                                    |
| 4.    | Пропускна здатність                           | Дозволяє обробляти дані на основі вимог програми, наявних мережевих та обчислювальних ресурсів. Це зменшить обсяг даних, необхідних для передачі в хмару. |

5.3.7) За визначеними факторами конкурентоспроможності (табл. 5.10) проведемо аналіз сильних та слабких сторін стартап-проекту (табл. 5.11).

Таблиця 5.11.

*Порівняльний аналіз сильних та слабких сторін*

| № п/п | Фактор конкурентоспроможності                 | Бали 1-20 | Рейтинг товарів-конкурентів у порівнянні з запропонованим |    |    |   |    |    |    |
|-------|-----------------------------------------------|-----------|-----------------------------------------------------------|----|----|---|----|----|----|
|       |                                               |           | -3                                                        | -2 | -1 | 0 | +1 | +2 | +3 |
| 1.    | Наявність патенту на модифіковану архітектуру | 18        |                                                           |    |    |   | +  |    |    |
| 2.    | Автономність                                  | 19        |                                                           |    |    |   |    | +  |    |
| 3.    | Інтеграція з корпоративними системами         | 17        |                                                           |    |    |   |    | +  |    |
| 4.    | Пропускна здатність                           | 19        |                                                           |    |    |   |    |    | +  |

5.3.8) Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких

(Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 5.11).

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 5.12.

*SWOT- аналіз стартап-проекту*

|                                                                                                                                                                             |                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <p><b>Сильні сторони:</b></p> <p>Ефективність, економія пропускну здатності, автономія, інтеграція з корпоративними системами</p> <p>Зворотній зв'язок від користувачів</p> | <p><b>Слабкі сторони:</b></p> <p>Слабке знання ринкової ситуації, нестача стартових капіталовкладень</p> |
| <p><b>Можливості:</b></p> <p>Отримання необхідних інвестицій</p> <p>Освоєння нових технологій</p> <p>Розширення аналітичних можливостей</p>                                 | <p><b>Загрози:</b></p> <p>Крадіжка інтелектуальної власності, інфляція</p>                               |

#### 5.4. Висновок до 5 розділу

В даному розділі проведено маркетинговий аналіз стартап-проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження.

Переглянувши всі позитивні і негативні риси аналітичних систем, можемо зробити висновок, що представлена ідея є актуальною та цілком адекватною для застосування. Технологія є доступною.

Оскільки ринок монополістична конкуренція, то бар'єри входу на ринок – високі. Наявна конкуренція високої концентрації середнього рівня інтенсивності. Для ринкової реалізації проекту обрано стратегію диференційованого маркетингу.

На основі попередніх досліджень було визначено фактори конкурентоспроможності, за допомогою яких проведено аналіз сильних та слабких сторін проекту, та проведено SWOT-аналіз.

## ЗАГАЛЬНІ ВИСНОВКИ

Підбиваючи підсумки, можна сказати, що Інтернет речей дуже цікава та перспективна сфера. IoT використовує телекомунікаційні сучасні безпроводові технології, такі як Wi-Fi, Bluetooth, ZigBee, Z-Wave та інші, які ще не дуже відомі, для передачі даних, об'єднання їх в одну мережу. Інтернет речей не може бути реалізовано тільки на одній якійсь технології. Технології комбінуються, щоб краще покрити ті завдання та потреби, які потрібні користувачам.

При виконанні магістерської дисертації розглянуті загальні архітектури Інтернету Речей, такі як еталонна модель IoT від MCE-T та всесвітнього форуму IoT, та рівнева IoT архітектура. Також була розглянута архітектура Smart Building та архітектура управління «Розумним будинком».

В магістерській дисертації була модифікована архітектурна концепція розумної будівлі. З додаванням нового елемента fog server можна буде зберігати важливу інформацію та обробляти дані в реальному часі, і все це з малою затримкою.

Запропонована мною архітектура складається з: smart devices, шлюзу, fog gateway, small data center, streaming data processor, cloud gateway, big data center, control applications, машинне навчання, applications, аналіз даних. Також вона містить компоненти управління пристроями та користувачами, які забезпечують стабільне та безпечне функціонування речей та контролювати проблеми доступу користувачів.

Розробляючи архітектуру IoT для певного рішення, важливо зосередити увагу на:

- послідовності – приділяючи достатньо уваги кожному елементу архітектури IoT та змушуючи їх працювати разом;
- гнучкості – можливість додавання нових функцій та нової логіки;
- інтеграції з корпоративними системами – об'єднання нових рішень IoT з раніше впровадженими корпоративними IT-рішеннями, такими як ERP, MES, WMS, системи управління доставкою тощо.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Елементи системи розумний будинок, їх призначення та принцип роботи [Електронний ресурс] – режим доступу: <https://mastery-of-building.org/uk/sostavlyayushhie-elementy-sistemy-umnyj-dom-ix-naznachenie-i-princip-raboty/amp/>
- [2] Інтернет вещей: сетевая архитектура и архитектура безопасности [Електронний ресурс] – режим доступу: <http://internetinside.ru/internet-veshhey-setevaya-arkhitektura-i/>
- [3] Як створити систему розумний будинок. Розумний будинок: що це таке і як це працює? Як це працює [Електронний ресурс] – режим доступу: <https://nurkeevo.ru/uk/kak-sozdat-sistemu-umnyi-dom-umnyi-dom-chto-eto-takoe-i-kak-eto/>
- [4] Галузі майбутнього: «розумні» міста та будинки [Електронний ресурс] – режим доступу: <https://mind.ua/publications/20188390-galuzi-majbutnogo-rozumni-mista-ta-budinki>
- [5] «ІНТЕРНЕТ РЕЧЕЙ» ЯК ПРАВОВИЙ ТЕРМІН [Електронний ресурс] – режим доступу: <http://ipp.kpi.ua/wp-content/uploads/2016/04/Баранов-2.pdf>
- [6] Створення системи розумний будинок. Мі Home настройка сценаріїв автоматизації [Електронний ресурс] – режим доступу: <https://giropark.ru/uk/nasosy-dlya-kolodcev/sozдание-sistemy-umnyi-dom-mi-home-nastroika-scenariev-avtomatizacii.html>
- [7] ITU-T, “Overview of the Internet of Things,” Recommendation Y.2060, June 2012.
- [8] ITU-T, “Common Requirements and Capabilities of a Gateway for Internet of Things Applications,” Recommendation Y.2067, June 2014.
- [9] IoT архитектура [Електронний ресурс] – режим доступу: <https://habr.com/ru/post/455377/>

- [10] Гапотій В.Д., Інформаційне право. Навчальний посібник [Електронний ресурс] – режим доступу: [http://eprints.mdpu.org.ua/id/eprint/8306/1/Информ\\_право\\_макет.pdf](http://eprints.mdpu.org.ua/id/eprint/8306/1/Информ_право_макет.pdf)
- [11] Smart home or building (home automation or domotics) [Електронний ресурс] – режим доступу: <https://internetofthingsagenda.techtarget.com/definition/smart-home-or-building>
- [12] The Current State of Security in Smart Home Systems/Daniel Schwarz, SEC Consult Vulnerability Lab - Vienna
- [13] Овчинников А.В. Архитектура удаленной системы управления умным домом посредством мобильного приложения // Научный журнал. 2017. №5 (18). С. 30-34.
- [14] Встраиваем своё устройство «Умного дома» в экосистему SmartThings [Електронний ресурс] – режим доступу: <https://habr.com/ru/company/samsung/blog/489834/>
- [15] Глупый умный дом. Часть 2, Apple против Samsung [Електронний ресурс] – режим доступу: <https://mobile-review.com/articles/2019/smart-home-2.shtml>
- [16] Samsung SmartThings. Классическая система от лидера рынка [Електронний ресурс] – режим доступу: <https://gadgetpage.ru/smart-house/3200-samsung-smarththings-klassicheskaja-sistema-ot-lidera-rynka.html>
- [17] Samsung SmartThings [Електронний ресурс] – режим доступу: <http://www.rovdo.com/smart-home-samsung-smarththings>
- [18] ЧТО ТАКОЕ УМНЫЙ ДОМ APPLE И КАК ОН РАБОТАЕТ [Електронний ресурс] – режим доступу: [https://strojdvor.ru/umnyj-dom/umniy-dom-apple/#h2\\_3](https://strojdvor.ru/umnyj-dom/umniy-dom-apple/#h2_3)
- [19] HomeKit - что это такое? Умный дом от Apple [Електронний ресурс] – режим доступу: <https://sprut.ai/client/article/1039>
- [20] Apple HomeKit: все что вам нужно знать [Електронний ресурс] – режим доступу: <https://ilounge.ua/review/apple-homekit-vse-chto-vam-nuzhno-znat>

- [21] Глупый умный дом. Часть 3, IKEA против Xiaomi [Електронний ресурс] – режим доступу: <https://mobile-review.com/articles/2019/smart-home-3.shtml>
- [22] Умный дом Xiaomi. Как все работает? [Електронний ресурс] – режим доступу: [https://newsmarthome.ru/xiaomi-smart-home/umnyj-dom-xiaomi#\\_\\_Xiaomi](https://newsmarthome.ru/xiaomi-smart-home/umnyj-dom-xiaomi#__Xiaomi)
- [23] Bluetooth шлюзы Xiaomi [Електронний ресурс] – режим доступу: <https://xiaomi-smarthome.ru/bluetooth-shlyuzu-xiaomi/>
- [24] Что такое ZigBee протокол в системе "умный дом" Xiaomi? [Електронний ресурс] – режим доступу: <https://xiaomi-smarthome.ru/zigbee/>
- [25] Умный дом Xiaomi и Apple: сравнение систем [Електронний ресурс] – режим доступу: [https://m-strana.ru/articles/umnyy-dom-xiaomi-i-apple-sravnenie/?utm\\_source=copy&utm\\_medium=direct&utm\\_campaign=copy\\_from\\_site](https://m-strana.ru/articles/umnyy-dom-xiaomi-i-apple-sravnenie/?utm_source=copy&utm_medium=direct&utm_campaign=copy_from_site)
- [26] IoT architecture: building blocks and how they work [Електронний ресурс] – режим доступу: <https://www.scnsoft.com/blog/iot-architecture-in-a-nutshell-and-how-it-works>
- [27] IoT, туман и облака: поговорим про технологии? [Електронний ресурс] – режим доступу: <https://habr.com/ru/company/cloud4y/blog/467711/>
- [28] Розроблення стартап-проекту. Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с
- [29] Баранов О.А., ІНТЕРНЕТ РЕЧЕЙ І ШТУЧНИЙ ІНТЕЛЕКТ: збірник матеріалів II-ї Міжнародної науково-практичної конференції (Львів, 17 листопада 2017 р.). – Львів : НУ «Львівська політехніка», 2017. – 318 с. (С. 18-42).
- [30] Сергієнко А.В., Губар В.Г. (2020), «Дистанційно-програмований контролер датчиків та пристроїв». Матеріали XIV Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ», с. 199-201
- [31] Могильний С.Б., Мініч М.А. (2020), «Дослідження ефективності методів моніторингу рухомих об'єктів в системах розумного будинку». Матеріали XIV

Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ», с. 240-242

[32] Пороло Є.О., Курдеча В.В. (2020), «Удосконалена архітектура мережі для хмарного інтернету речей», Матеріали XIV Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ», с. 219-221

[33] Осипчук С.О., Мошинська А.В., Кіращук В.В. (2019), «Прикладні аспекти реалізації рішень передавання інформації в технологіях інтернету речей». Матеріали XIII Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ», с. 17-21

[34] Кравчук С.О. (2019), «Інтелектуальні M2M-шлюзи для безпроводових доменів приладів». Матеріали XIII Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ», с. 181-184

[35] Міхненко Я.О., Курдеча В.В. (2019), «Порівняння видів хмарних сервісів в IoT». Матеріали XIII Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ», с. 207-209

[36] Ковбаса А.О., Петрова В.М. (2019), «Методи балансування навантаження в бездротових сенсорних мережах». Матеріали XIII Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ», с. 301-303

[37] Borovskyi B. M. A, «Comparison between open-source platforms for cloud computing». Proceedings of the XIII International Scientific and Technical Conference "PROSPECTS OF TELECOMMUNICATIONS", pp. 311-313

[38] Л.О.Уривський, С.О. Осипчук. Концептуальные аспекты организации IoT сетей в Украине / Наукоемкие технологии в инфотелекоммуникациях: обработка информации, кибербезопасность, информационная борьба. Монография. Под общ. ред. В.М. Безрук, В.В. Баранник. - Харьков: Харьков: ФОП Бровін О.П., 2018. – с. 89-112.