

NOSQL-АРХІТЕКТУРА FIREBASE В АВТОМАТИЗОВАНИХ ІОТ-СИСТЕМАХ ПІДРАХУНКУ ГОЛОСІВ

Світлана Лавренчук, Ніна Здолбіцька, Сергій Здолбіцький

FIREBASE NOSQL ARCHITECTURE IN AUTOMATED IOT-BASED VOTING SYSTEMS

Svitlana Lavrenchuk, Nina Zdolbitska, Serhii Zdolbitskyi

Луцький національний технічний університет, вул. Львівська, 75, Луцьк, 43018, Україна

Abstract. *This paper examines the architectural features and practical implementation of an automated electronic voting system based on the Internet of Things concept and the Firebase Realtime Database cloud NoSQL database. A three-tier "Edge-Cloud-Web" architecture is proposed, which addresses the latency issue and ensures instantaneous synchronization of results. Special attention is dedicated to the methods of hardware-software sensor noise filtering at the peripheral level (ESP8266) and ensuring data integrity within the NoSQL structure using atomic increment operations. Conducted experiments have demonstrated that the overall end-to-end system latency does not exceed 300 ms, and the architecture completely eliminates the race condition problem.*

Традиційні системи голосування (паперові бюлетені) характеризуються значними часовими витратами на підрахунок, високою ймовірністю людських помилок та вразливістю до маніпуляцій на етапі консолідації даних. З іншого боку, повністю цифрові системи [1] (веб-форми, онлайн-опитування) позбавлені фізичної можливості верифікації присутності виборця, що підвищує ризики масових бот-атак або фальсифікацій за допомогою скриптів.

Перспективним напрямком вирішення цієї дилеми є створення гібридних автоматизованих систем на базі концепції Інтернету речей [2, 3], де фізичне опускання бюлетеня фіксується сенсорами, а агрегація та відображення даних відбуваються у хмарі в режимі реального часу.

Більшість сучасних IoT-рішень для моніторингу подій використовують реляційні бази даних або брокери повідомлень типу MQTT. Проте класичні РБД (на кшталт MySQL, PostgreSQL) демонструють низьку продуктивність при високій інтенсивності одночасних запитів, яка є типовою для фінальних етапів голосування. Крім того, вони вимагають налаштування складних ORM-прошарків та серверної інфраструктури, що здорожує систему.

Мета – розробка та дослідження ефективності трьохрівневої архітектури автоматизованої системи підрахунку голосів на базі мікроконтролера ESP8266 та NoSQL сховища Firebase Realtime Database, що забезпечує високу швидкість, масштабованість та відмовостійкість без використання традиційних серверних архітектур.

Для досягнення поставленої мети розроблено трьохрівневу архітектурну модель «Edge-Cloud-Web» (рисунк 1), яка мінімізує обчислювальне навантаження на кінцеві пристрої та переносить логіку синхронізації у хмару.

Апаратну основу периферійного вузла (виборчої урни) становить мікроконтролер ESP8266, який інтегрує в собі 32-бітний процесор та вбудований Wi-Fi стек стандарту 802.11 b/g/n. Як сенсор виявлення бюлетенів використано інфрачервоний датчик відбиття HW-201.

Основна проблема апаратного рівня – оптичний та електричний шум. Падіння паперового листа через зону дії датчика викликає явище «брязкоту контактів», коли сенсор генерує серію імпульсів замість одного сигналу. Для усунення цієї проблеми було розроблено програмний алгоритм фільтрації без зупинки процесора, що базується на фіксації часових міток:

$$t_{next} > t_{last} + T_{debounce} \quad (1)$$

де t_{next} – час поточного спрацювання, t_{last} – час попереднього підтвердженого голосу, а $T_{debounce}$ – вікно блокування, яке експериментально встановлено на рівні 300 мс. Всі імпульси всередині цього вікна ігноруються мікроконтролером як шум.

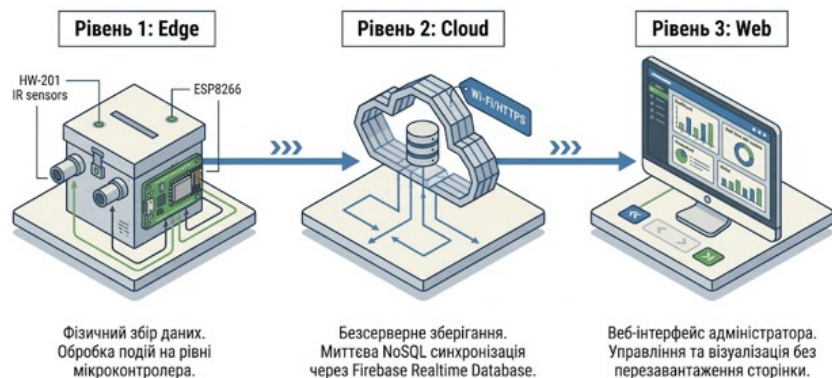


Рисунок 1 – Тривірнева архітектура IoT-системи

Центральним елементом системи є Firebase Realtime Database. На відміну від реляційних баз даних, Firebase зберігає дані як єдине неструктуроване JSON-дерево. Це дозволяє динамічно змінювати структуру опитувань, додавати нові виборчі дільниці або варіанти відповідей без необхідності проведення міграцій схем даних.

Для забезпечення гнучкості системи реалізовано механізм activePoll. Контролер ESP8266 перед початком роботи періодично виконує HTTP GET-запит до ключа activePoll. Це дозволяє адміністратору централізовано змінювати тему або ID поточного голосування через веб-панель. Контролери автоматично перемикаються на нову гілку БД без необхідності їх фізичного перепрошивання.

Клієнтський веб-додаток (панель адміністратора) реалізований за допомогою технологій HTML5, CSS3 та JavaScript. Зв'язок із Firebase RTDB відбувається через офіційний SDK за допомогою протоколу WebSockets. Використання методу onValue() дозволяє веб-інтерфейсу підписатися на оновлення конкретної гілки JSON-дерева. Будь-яка зміна лічильника голосів у хмарі ініціює push-повідомлення до клієнта, що забезпечує реактивне оновлення графіків у режимі реального часу без постійного опитування сервера.

У системах електронного голосування критично важливою є проблема стану гонитви – ситуації, яка виникає, коли кілька паралельних процесів (або транзакцій) одночасно намагаються прочитати та модифікувати одні й ті самі дані. Якщо дві різні виборчі урни одночасно фіксують голос і надсилають запит до БД, виникає ризик втрати даних. Хмара може обробити обидва читання одночасно, і в результаті два голоси запишуться як один.

Для вирішення цієї проблеми в NoSQL архітектурі Firebase застосовано атомарні оновлення на стороні сервера за допомогою спеціального токена інкременту. Запит від ESP8266 формується у вигляді HTTP PATCH-запиту:

```
HTTP
PATCH /polls/poll_id_2026/candidates/candidate_01.json
Content-Type: application/json
{
  ".sv": "increment",
  "value": 1
}
```

У цьому випадку операція збільшення лічильника на 1 виконується безпосередньо всередині ядра СУБД Firebase як атомарна транзакція. Це гарантує, що навіть при тисячах одночасних запитів від різних Edge-пристроїв кожен голос буде враховано коректно, а черга запитів побудується автоматично на рівні хмари.

Для тестування розробленої архітектури створено випробувальний стенд, що складався з трьох периферійних модулів на базі ESP8266 та веб-панелі моніторингу. На першому етапі тестувалася стійкість алгоритму фільтрації шумів.

Експеримент показав, що при інтенсивному киданні макетів бюлетенів датчик HW-201 генерував від 3 до 7 хибних імпульсів на одну дію через вібрацію паперу. Програмний фільтр з вікном 300 мс продемонстрував 100 % точність, відсікаючи всі шуми та зафіксувавши рівно ту кількість голосів, яка фактично була опущена.

На другому етапі вимірювався життєвий цикл голосу (end-to-end delay) – час від моменту перетину ІЧ-променя до відображення результату на графіку веб-інтерфейсу (таблиця 1).

Таблиця 1

Розподіл затримок у розробленій системі голосування

Етап життєвого циклу голосу	Середній час затримки (мс)
Фіксація сигналу датчиком та Edge-обробка	10 мс
Формування HTTPS-пакета та передача у мережу Wi-Fi	85 мс
Маршрутизація та атомарний інкремент у Firebase RTDB	45 мс
Передача оновлення через WebSocket на Web-клієнт	70 мс
Загальний час (End-to-End)	210 мс

Загальний час реакції системи склав близько 210 мс (з максимальними піками до 300 мс при погіршенні якості Wi-Fi сигналу), що є чудовим результатом для систем реального часу і повністю задовольняє вимоги до автоматизованого підрахунку голосів.

Отож, можемо зробити висновок, що успішно розроблено та досліджено NoSQL-орієнтовану IoT-архітектуру для електронної системи голосування. Використання Firebase Realtime Database дозволило побудувати повністю Serverless-систему, яка не потребує окремого сервера додатків, легко масштабується та забезпечує реактивне оновлення даних за 210 мс.

Застосування апаратно-програмного дебаунсингу на рівні ESP8266 вирішило проблему фізичних шумів датчика HW-201, гарантуючи точність первинних даних. Впровадження серверних атомарних інкрементів повністю нівелювало загрозу втрати голосів через стан гонитви, забезпечуючи цілісність даних NoSQL-структури в умовах високої щільності запитів.

Перспективами подальших досліджень є інтеграція у периферійний рівень RFID-зчитувачів (наприклад, RC522) для персональної авторизації виборців [4, 5] за електронними ID-картками, а також впровадження алгоритмів шифрування AES-128 для захисту трафіку між Edge та Cloud рівнями.

Бібліографія

1. Гусаревич Н. Електронне голосування: концептуальні підходи. Аспекти публічного управління, 9(4), 2021. С. 104-110. <https://doi.org/10.15421/152142>
2. Intelligent Digital Voting System Using ESP32 with Biometrics and Camera Based Authentication Framework / Vuppula Manohar et al. International Journal of Engineering Research and Science & Technology. 2026. Vol. 22, no. 2(1). P. 802-807. [https://doi.org/10.62643/ijerst.2026.v22.n2\(1\).2676](https://doi.org/10.62643/ijerst.2026.v22.n2(1).2676)
3. Wahyu Andre Wibowo, Erik Iman Heri Ujjianto. The IoT-Based E-Voting System Using Fingerprint Biometrics for School Elections. Journal of Information Systems and Informatics. Vol. 7, No. 4, 2025. Pp. 3645-3667. DOI: 10.63158/journalisi.v7i4.1328.
4. Anuja Pawar, Kirti Pujari, Prachi Pawar, Prof. A.D. Harale. Literature Review on Fingerprint based Electronic Voting System . International Journal of Advanced Research in Science, Communication and Technology. Volume 5, Issue 4, 2025. P. 1095-1109. DOI: 10.48175/IJARST-29461
5. Здолбіцька Н.В., Здолбіцький А.П., Семенко О.О. Системи електронної ідентифікації і управління доступом користувачів. Науковий журнал «Комп'ютерно-інтегровані технології: освіта, наука, виробництво». Луцьк: Видавництво ЛНТУ. Вип. 36. 2019. С. 103-108.