

Порівняння архітектур для застосунку автоматизованого планування та управління контентом у соцмережах

Софія Тригуб, студент¹ (ORCID: 0009-0007-8451-746X).

¹Київський національний університет будівництва і архітектури, Київ, Україна

АНОТАЦІЯ

У статті розглядається концепція та основні архітектурні підходи до побудови застосунків для автоматизованого планування та управління контентом у соціальних мережах. Основна ідея полягає у створенні інструменту, що дозволяє SMM-менеджерам, маркетологам та власникам бізнесу ефективно організувати роботу з публікаціями, координувати команду та контролювати результати. Проведено порівняння монолітної, мікросервісної, безсерверної та подієво-орієнтованої архітектур. Запропонована архітектура включає модулі календаря публікацій, управління завданнями, збереження медіа та шаблонів, систему сповіщень, а також базову аналітику для оцінки ефективності контенту. Система може бути масштабована за рахунок інтеграції із соціальними мережами, додавання розширеної аналітики та інтелектуальних підказок для створення контенту.

Ключові слова: клієнт-серверна архітектура, мікросервісна архітектура, подієво-орієнтована архітектура, безсерверні обчислення, монолітна система, автоматизація контенту, планування публікацій, соціальні мережі, вебзастосунок.

1. ВСТУП

Сучасні системи для автоматизованого планування публікацій у соціальних мережах є складними програмними комплексами, що обробляють велику кількість подій, повідомлень і запитів у режимі реального часу. Ефективність таких систем безпосередньо залежить від обраної архітектури. Використання традиційних інструментів, таких як звичайні календарі чи електронні таблиці, у таких умовах часто є неефективним: вони не забезпечують належної координації дій, не підтримують інтеграцію з соціальними платформами та не надають зручних засобів аналітики. Це створює труднощі для команд, що працюють над кількома проєктами одночасно.

Запропонована в даній роботі концепція полягає у створенні застосунку для автоматизованого планування та управління контентом у соціальних мережах. На відміну від базових інструментів організації роботи, такий застосунок має забезпечити зручний календар публікацій, систему ролей для команди, внутрішню комунікацію, нагадування про дедлайни та базову аналітику. Очікується, що це сприятиме підвищенню ефективності роботи, зниженню ризику пропуску важливих публікацій та покращенню загальної взаємодії всередині команди.

2. МЕТА ТА ЗАВДАННЯ ДОСЛІДЖЕННЯ

Метою роботи є концептуальна розробка прототипу застосунку для автоматизованого планування та управління контентом у соціальних мережах.

Основними завданнями є: порівняння та визначення архітектури системи та ключових модулів (календар, аналітика, управління завданнями, комунікація), розробка механізмів керування ролями та доступом користувачів, проектування інтерфейсу для створення, редагування та планування публікацій.

3. АНАЛІЗ ДЖЕРЕЛ

У публікації [1] подано детальний розгляд архітектури системи для планування контенту в соціальних мережах. Автори описують типові компоненти таких застосунків -

модулі автентифікації, планувальник, обробник подій та систему доставки публікацій. Стаття [2] зосереджується на принципах побудови подієво-орієнтованих систем. У ній розкриваються питання використання черг повідомлень, брокерів подій і шаблонів реактивної обробки даних.

У матеріалі [3] проведено порівняння трьох основних архітектурних моделей — монолітної, мікросервісної та безсерверної. Автори детально аналізують їхні переваги та обмеження у контексті швидкості розгортання, вартості утримання та масштабування.

Для оцінки актуальності проблеми було проведено опитування серед 40 SMM-спеціалістів, маркетологів та власників малого бізнесу, які регулярно працюють із соціальними мережами. Учасникам пропонувалося описати основні труднощі у процесі планування контенту, результати опитування представлені на рисунку 1.

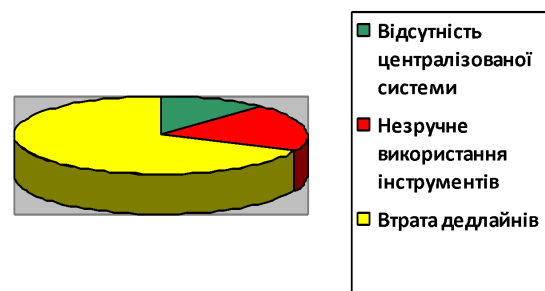


Рисунок 1. Результати опитування

Результати показали, що 68% респондентів стикаються з проблемою втрати дедлайнів або хаотичної організації постів, 20% вважають незручним використання стандартних інструментів, і ще 12% наголосили на відсутності централізованої системи комунікації з командою.

4. ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Архітектура застосунку базується на клієнт-серверній моделі, що передбачає поділ на клієнтську частину, серверну логіку та базу даних. Нижче розглянуто основні архітектурні підходи організації серверної частини -

монолітний, мікросервісний, безсерверний (serverless) і подієво-орієнтований - та їх придатність для подібних рішень. Монолітна архітектура передбачає об'єднання всіх компонентів системи (інтерфейс, логіка, база даних) у єдиному застосунку. Її головна перевага полягає у простоті реалізації та швидкому запуску продукту, що робить її доцільною для створення мінімально життєздатних прототипів (MVP). Разом з тим, із зростанням навантаження та кількості користувачів така архітектура втрачає гнучкість, стає складною для масштабування і супроводу.

Мікросервісна архітектура, на відміну від моноліту, розділяє систему на незалежні сервіси, кожен з яких виконує окрему бізнес-функцію: планування публікацій, обробку вебхуків, аналітику або управління користувачами. Це забезпечує можливість незалежного розгортання, масштабування і тестування модулів. Однак її впровадження потребує зрілої інженерної культури, системи моніторингу, оркестрації контейнерів (наприклад, Kubernetes) та налагоджених CI/CD-процесів.

Безсерверна (serverless) архітектура базується на використанні хмарних функцій (наприклад, AWS Lambda або Google Cloud Functions), що виконуються лише у відповідь на події. Цей підхід знижує витрати на підтримку інфраструктури та добре підходить для застосунків із нерівномірним навантаженням, наприклад, коли публікації плануються нерівномірно протягом доби. Основними недоліками є затримки через «холодні старти» функцій, складність тестування й обмеження часу виконання.

Подієво-орієнтована архітектура (event-driven) передбачає обмін повідомленнями між компонентами через шину подій або чергу (Kafka, RabbitMQ, AWS EventBridge). У такій моделі планувальник публікацій, модуль обробки контенту та модуль аналітики взаємодіють асинхронно через події, що дозволяє легко масштабувати систему та ізолювати навантаження. Цей підхід є найбільш придатним для високонавантажених систем управління контентом, де щосекунди відбувається велика кількість операцій публікації та обробки вебхуків. Водночас реалізація подієвої моделі вимагає ретельного проектування логіки доставки, дублювання і збереження подій. Узагальнений аналіз розглянутих архітектур наведено в таблиці 1.

Таблиця 1. Порівняльний аналіз архітектур

Архітектур	Переваги	Недоліки	Доцільність
Монолітна	Простота, швидкий старт	Важко масштабувати	MVP, малі системи
Мікросервісна	Гнучкість, незалежне розгортання	Висока складність	Великі системи
Serverless	Автоматичне масштабування, економія ресурсів	Затримки, обмеження часу виконання	Хмарні сервіси, нерівномірні навантаження
Подієво-орієнтована	Масштабованість, надійність	Складність у проектуванні	Високонавантажні платформи

Запропонований застосунок передбачає інтеграцію з основними соціальними мережами (Instagram, Facebook, TikTok, YouTube, LinkedIn та ін.), надаючи користувачам універсальну платформу для централізованого управління всіма акаунтами. Основна ідея полягає у створенні системи,

що поєднує календарне планування, автоматизований автопостинг, модулі аналітики та рекомендаційний механізм для підвищення ефективності комунікації. Користувач може формувати детальний контент-план на день, тиждень чи місяць, додавати тексти, візуальні матеріали та інтерактивний контент, а система подбає про своєчасну публікацію у вибраних соцмережах. Функція відкладених постів забезпечує охоплення аудиторій у різних часових зонах, а єдиний акаунт у застосунку дозволяє керувати кількома платформами одночасно, знімаючи потребу вручну перемикатися між сервісами.

На стороні клієнта реалізовано інтерфейс для створення постів, календар контенту, налаштування автопостингу й перегляд статистики. Бекенд відповідає за обробку запитів, API-інтеграцію з соціальними мережами, генерацію аналітики та управління завданнями публікацій. Для збереження інформації використовується база даних, де фіксуються ключові метадані: текст посту, зображення чи відео, дата та час публікації, платформа та статистичні показники ефективності. Така архітектура забезпечує масштабованість системи, її гнучкість та можливість розширення функціоналу в майбутньому.

5. ВИСНОВКИ

Запропонована концепція застосунку має потенціал стати ефективним інструментом для SMM-спеціалістів, бізнесів, блогерів та агенцій. Порівняння показало, що вибір архітектури має ґрунтуватися на масштабі продукту, очікуваному навантаженні та ресурсах команди. Для швидкого створення прототипів оптимальним є монолітний або serverless-підхід, тоді як для промислових систем із великою кількістю одночасних подій доцільно застосовувати подієво-орієнтовану архітектуру, що забезпечує найкраще співвідношення продуктивності, надійності та гнучкості. Система має об'єднувати ключові функції - від формування контент-плану й автопостингу до збору аналітики та генерації. Використання інтуїтивного інтерфейсу, інтерактивного календаря та колаборативних можливостей значно підвищує зручність роботи команд. Аналітичні модулі на основі метрик ефективності забезпечують можливість прийняття рішень, заснованих на даних, а рекомендаційна система допомагає оптимізувати час і формат публікацій. Подальший розвиток проекту передбачає розширення інтеграцій із новими платформами, удосконалення алгоритмів прогнозування та впровадження елементів штучного інтелекту для створення контенту. Це відкриває перспективи для формування універсальної екосистеми, здатної значно підвищити якість та результативність роботи у сфері цифрового маркетингу.

Список літератури

- [1] **AWS Architecture Blog.** *Building event-driven architectures.* – Amazon Web Services, 2024. – Режим доступу: <https://aws.amazon.com/architecture>.
- [2] **TechForProduct.** *System Design: Social Media Scheduler.* – 2024. – Режим доступу: <https://techforproduct.com/system-design-social-media-scheduler>.
- [3] **Harness.io.** *Monolithic vs Microservices vs Serverless: Which Is Right for You?* – Harness, 2023. – Режим доступу: <https://harness.io/blog/architecture>.