



Розробка бібліотеки на .NET для роботи зі знімками (snapshots) для EventStoreDB

МАГІСТЕРСЬКА РОБОТА (НаУКМА, ФІ, МП-2, ІПЗ 2023)

ВИКОНАВ: Осадчук Володимир Ігорович

НАУКОВИЙ КЕРІВНИК: доцент, к.т.н. Франчук Олег Васильович

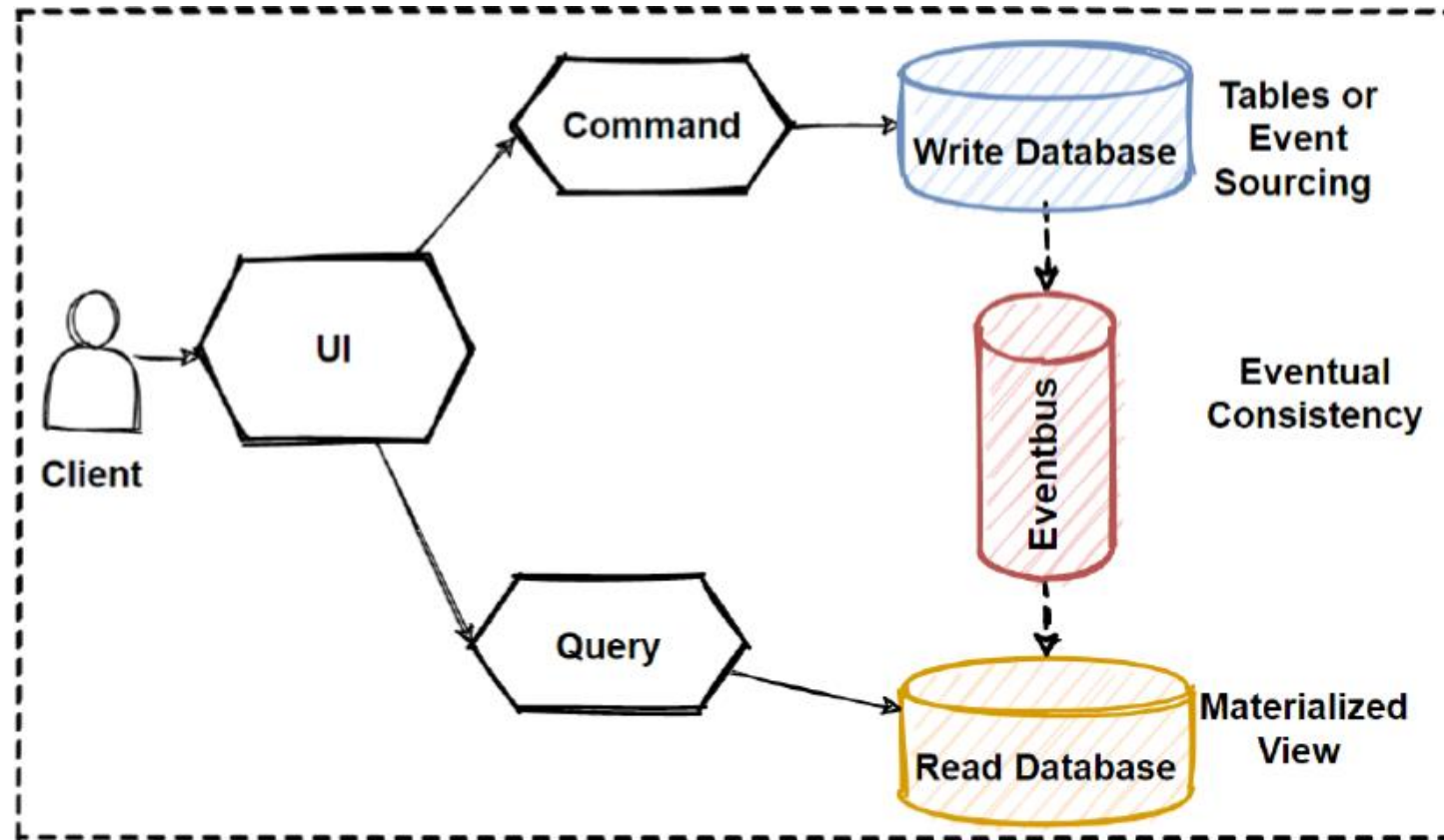
Мета

- Огляд EventStoreDB – бази даних, яка спеціалізується на збереженні подій (events) та їх історії, її переваги та недоліки.
- Огляд клієнтських бібліотек на .NET для роботи зі знімками (snapshots).
- Розробка власної бібліотеки для роботи зі знімками, порівняння з існуючими.
- Застосування EventStoreDB та створеної бібліотеки до веб-застосунку.

Event Sourcing + CQRS

- **Event Sourcing** – це підхід до зберігання даних, при якому замість збереження поточних станів об'єктів, генеруються події (events), які сталися в системі. Кожна подія містить інформацію про зміни в системі в певний момент часу.
- **CQRS** – це підхід до розподіленої моделі системи, де команди (commands), які змінюють стан системи, та запити (queries), які повертають поточний стан системи, обробляються окремо. Це дозволяє забезпечити більшу пропускну здатність, швидкість та масштабованість системи.

Event Sourcing + CQRS



EventStoreDB

- EventStoreDB (EventStore) – база даних, яка спеціалізується на зберіганні та управлінні подіями (events). Вона побудована з урахуванням принципів Event Sourcing та CQRS і розроблена для підтримки потокової обробки подій і забезпечення стійкості історії.

EventStoreDB

- Підтримка подійного архівування
- Незмінний журнал подій
- Гарантована послідовність подій
- Розподілена та масштабована архітектура
- Підтримка транзакцій
- Гнучкість схеми даних

EventStoreDB



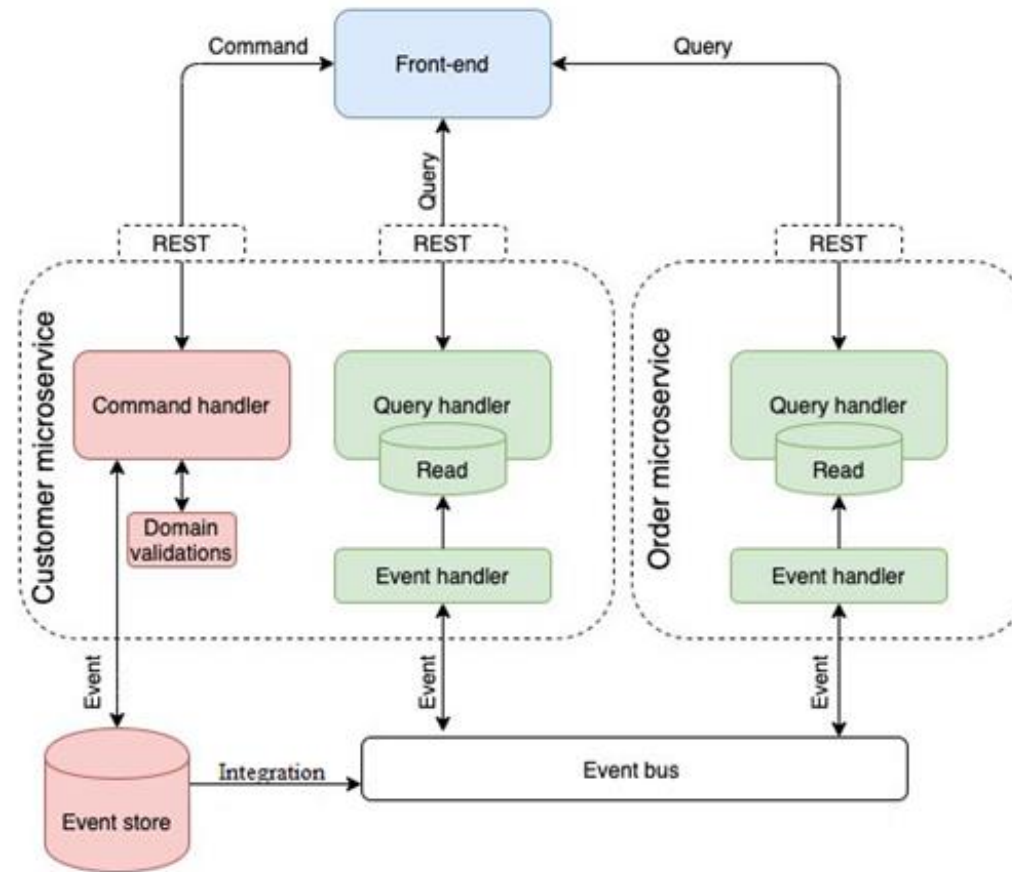
EventStoreDB

- Подія (event) – центральний елемент у EventStore, відображає зміну стану системи і зберігається в журналі подій. Кожна подія має унікальний ідентифікатор, дані та метадані.
- Потік подій (stream) – це послідовний запис подій в EventStoreDB. Він представляє собою послідовність подій, пов'язаних з певним агрегатом, об'єктом або сутністю.

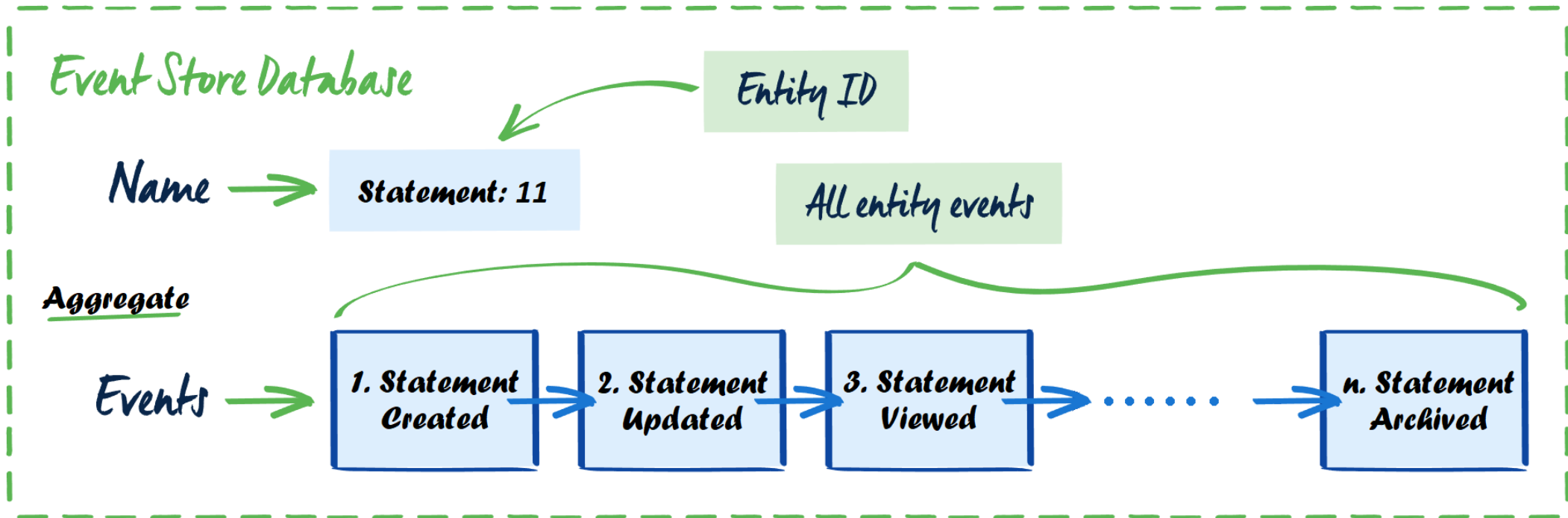
EventStoreDB

- Підписка (subscription) – дозволяє отримувати повідомлення про нові події, які додані до певного потоку. Це механізм, за допомогою якого можна реагувати на зміни в системі в реальному часі.
- Агрегат (aggregate) – логічний контейнер для подій, які пов'язані зі змінами стану певного об'єкта або сутності в системі. Він може включати один або більше потоків подій.

EventStoreDB



EventStoreDB



Знімки (Snapshots) агрегатів

- Знімок (Snapshot) – це згортка стану агрегата на певний момент часу, яка включає всю необхідну інформацію для відновлення стану агрегату без необхідності обробки всіх попередніх подій.
- Умови створення:
 - Визначена кількість подій, яку агрегат обробляє.
 - Визначений часовий інтервал.

Знімки (Snapshots) агрегатів

- «Anabasis.EventStore.Snapshot.InMemory» – 32.6K завантажень
- «Anabasis.EventStore.Snapshot.SqlServer» – 13.6K завантажень
- «EventStore.Repository.Snapshots.MSSQLServer» – 909 завантажень

Available Packages: Top 3

 Anabasis.EventStore.Snapshot.InMemory	1.0.185
 Anabasis.EventStore.Snapshot.SQLServer	1.0.185
 EventStore.Repository.Snapshots.MSSQLServer	0.1.2

Anabasis.EventStore.Snapshot.InMemory

- Переваги
 - Швидкодія
- Недоліки
 - Втрата даних
 - Відсутність спільного доступу
 - Неможливість відновлення системи
 - Затрати виділеної пам'яті мікросервіса
 - Відсутність зовнішнього доступу

Anabasis.EventStore.Snapshot.SqlServer

- Переваги
 - Довгострокове збереження даних
 - Наявність спільного доступу
 - Можливість часткового відновлення системи
 - Наявність зовнішнього доступу
- Недоліки
 - Збереження динамічних об'єктів в реляційну БД
 - Всі знімки зберігаються в одній таблиці
 - В якості первинного ключа використовується стрічка

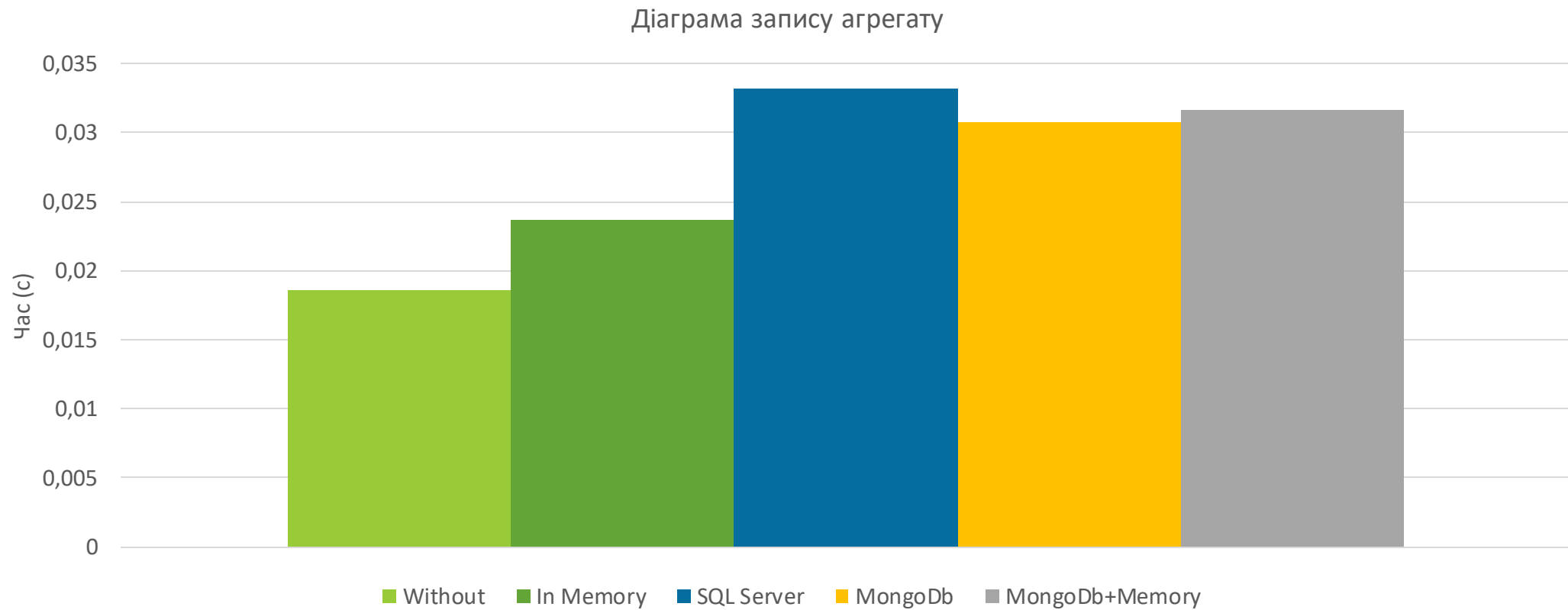
Власна бібліотека (MongoDB+Memory)

- Переваги
 - Швидкодія
 - Збереження динамічних об'єктів
 - Окрема колекція для кожного агрегату
 - Довгострокове збереження даних
 - Наявність спільного доступу
 - Можливість часткового відновлення системи
 - Наявність зовнішнього доступу

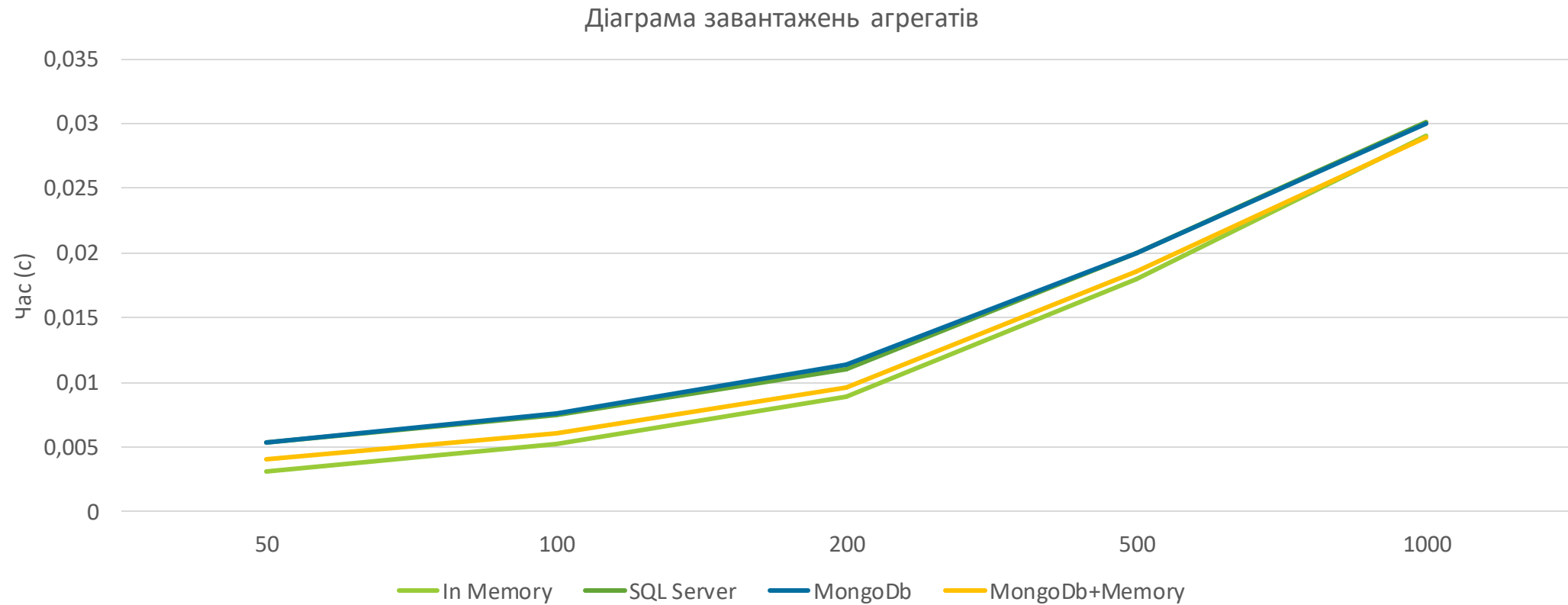
Власна бібліотека (MongoDB+Memory)

- Недоліки
 - Збереження даних в обидва місця
 - Неактуальність збережених даних в пам'яті

Порівняння бібліотек

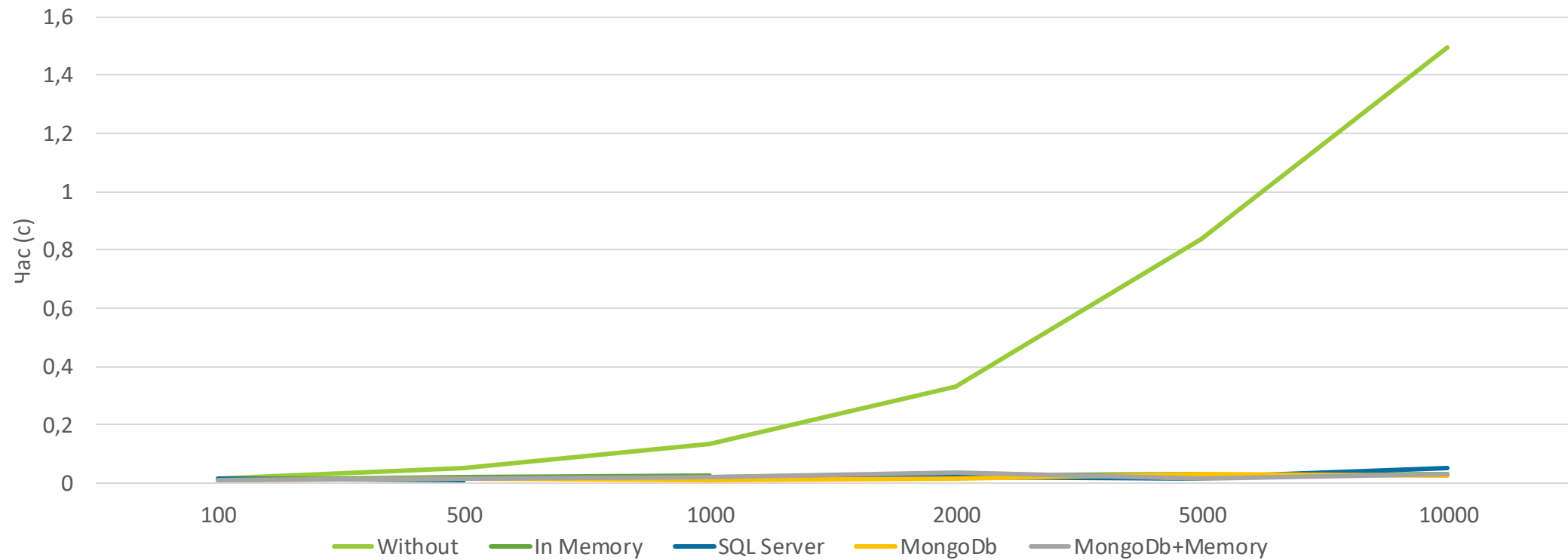


Порівняння бібліотек



Порівняння бібліотек

Діаграма завантаження агрегату за кількістю подій



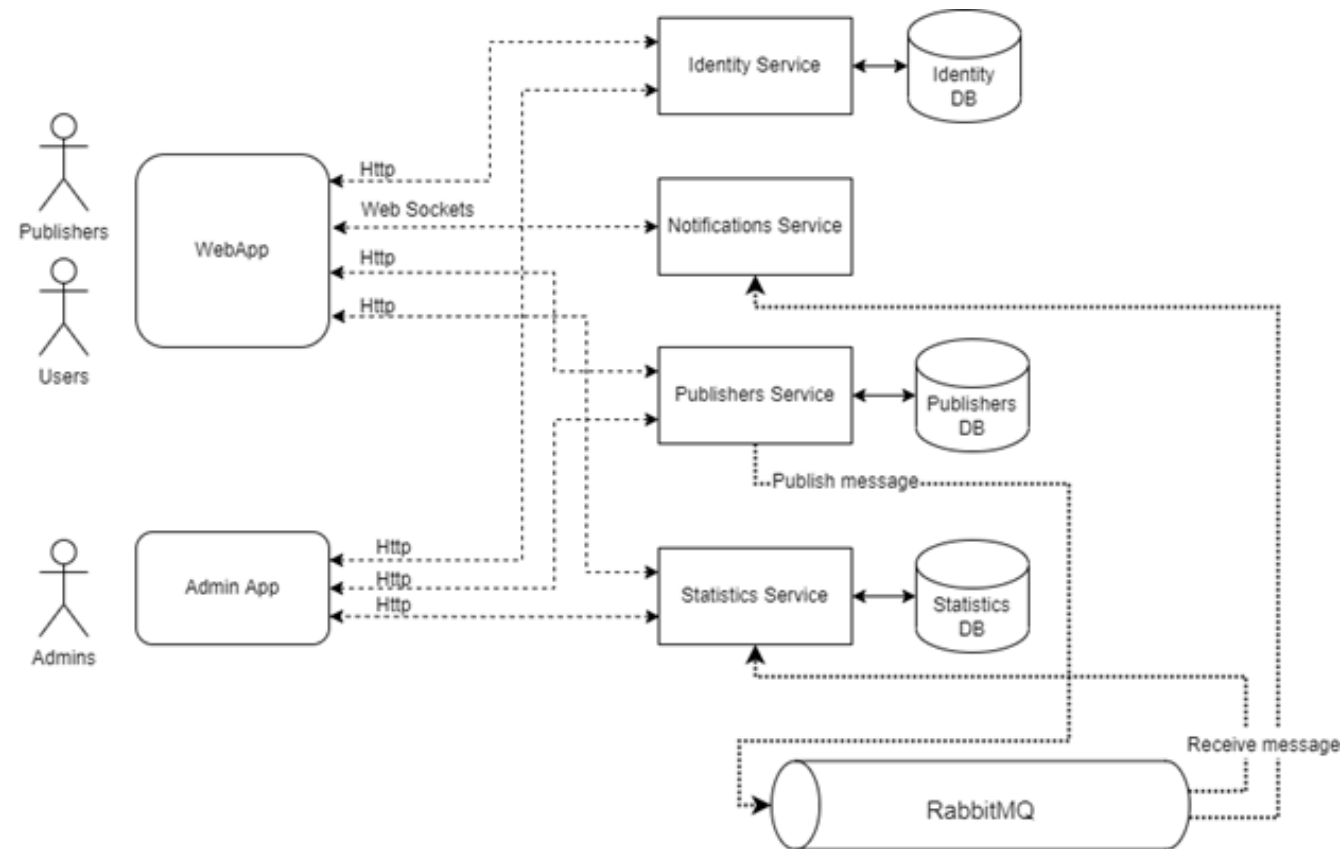
Порівняння бібліотек

- Середня швидкість запису агрегату без знімків та зі знімками майже не відрізняється.
- Середня швидкість запису з використанням різних бібліотек займає приблизно однаковий час.
- При завантаженні агрегату швидше працюють бібліотеки, що зберігають знімки безпосередньо в пам'яті.
- Побудова агрегату без бібліотек займає значно більше часу.
- Оптимальною умовою для зберігання нової версії агрегату є кожні 50-200 подій.

Застосування EventStore до застосунку

- Веб-застосунок з оголошень по прокату паперових книг.
- Технології
 - .NET
 - Angular
 - MS SQL Server
 - RabbitMQ
 - WebSockets

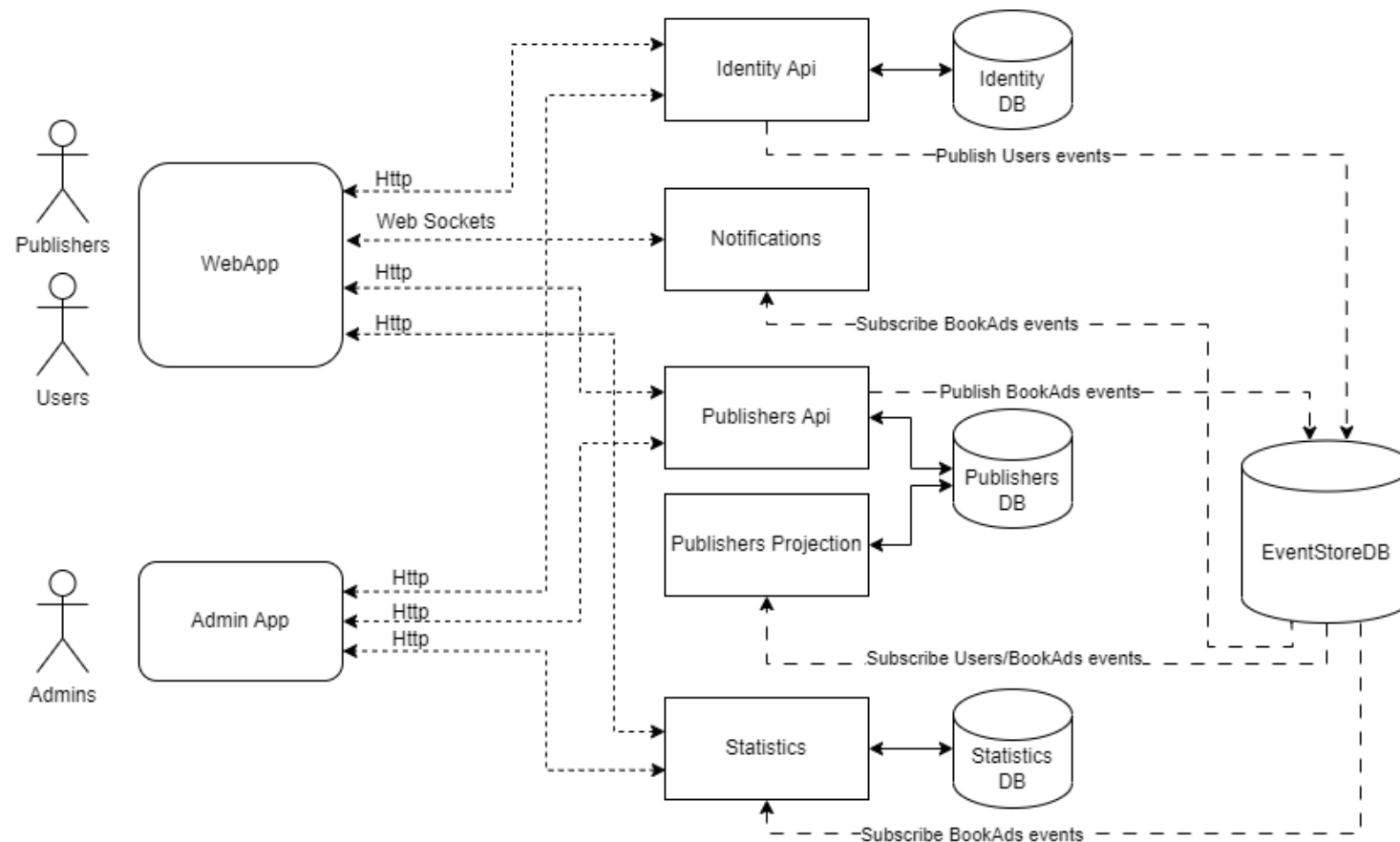
Застосування EventStore до застосунку



Застосування EventStore до застосунку

- **UserAggregate**
 - UserCreatedEvent
 - UserUpdatedEvent
- **BookAdAggregate**
 - BookAdCreatedEvent
 - BookAdUpdatedEvent
 - BookAdDeletedEvent
 - BookAdAvailabilityChangedEvent
 - BookAdViewedEvent

Застосування EventStore до застосунку



ВИСНОВКИ

- Розглянуто EventStoreDB в контексті патерна Event Sourcing для мікросервісної архітектури.
- Розглянуто переваги використання знімків та умови коли їх краще робити.
- Розроблено власну бібліотеку для збереження знімків (snapshots).
- Проведено порівняння розробленої бібліотеки з існуючими.
- Застосовано EventStoreDB та створену бібліотеку до веб-застосунку.



Дякую за
увагу!
