

# РЕАЛІЗАЦІЯ СТРУКТУРИ РОЗПОДІЛЕНИХ СИСТЕМ ВИКОНАННЯ ЗАПИТІВ ОБРОБКИ ДАНИХ В ХМАРНИХ РІШЕННЯХ

Вінницький національний технічний університет

## **Анотація**

*Запропоновано реалізацію “Столичної” структури мікросервісів в розподілених системах, що дозволяє оптимізувати необхідну кількість ресурсів для реалізації та підтримки високонавантажених систем розміщених в хмарному середовищі*

**Ключові слова:** мікросервіси, високонавантажені, хмари, хмарне середовище, оптимізація

## **Abstract**

*In work proposed implementation of “Capital” structure of microservices in decentralized systems. It allows optimize amount of required resources needed for realizations and support of high-loaded systems in cloud environment.*

**Keywords:** microservices, high-load, clouds, cloud environment, optimization

## **Вступ**

Зазвичай рішення побудови мікросервісної реалізації структури розподілених систем для хмарних рішень реалізується за практикою DDD (Domain Driven design) [1] і вимагає створення окремих сервісів [2] для кожних окремих сутностей, що наявні у домені програми. Для прикладу можна розглянути уніфіковані підходи до їх розробки, такі як Clean Architecture [3]. Такі реалізації вже використовують підходи DDD, CQRS [4], Unit testing, integration testing [5].

## **Реалізація структури мікросервісів**

Однією із основних особливостей реалізації мікросервісної структури є використання Onion Architecture [6]. Цей підхід дозволяє поділити весь сервіс на декілька шарів:

1. Domain layer – шар, що відповідає за логіку саме на рівні домену. Тобто моделі, процеси та все, що не прив'язано до конкретного мікросервісу, а більше до сфери в якій функціонує розподілена система.
2. Application layer – шар, що відповідає за логіку конкретної частини системи, та дозволяє декларувати зовнішні залежності мікросервісу такі як бази даних, інші сервіси та
3. Representation Layer – шар, у якому міститься опис контракту для комунікації з ним або в деяких випадках інтерфейс для користувача
4. Infrastructure layer – шар, у якому міститься загальні залежності мікросервісу та алгоритми взаємодії з ними. Його задача конкретна імплементація декларованих раніше залежностей на рівні application шару.

Проблема в тому, що при побудові реальних розподілених систем вдаються до дуже дрібного рівню розподілення зон відповідальностей мікросервісів. Це призводить до створення окремих сервісів на незначну частину процесу системи. В таких випадках ресурси, такі як час розробників, на базову частину яка не несе ніякої логіки за собою, а є лише фундаментом для її створення та надає змогу розширювати її надалі, сильно перевищують корисну частину.

Для вирішення цієї проблеми, запропонований варіант нової, “столичної” архітектури розподіленої системи виконання запитів на базі мікросервісів. Основа цієї концепції в знаходженні балансу і в отриманні швидкості розробки моноліту [7] та можливістю розширення мікросервісів.

Для досягнення цього, спочатку створюється головний мікросервіс, який слугує основним хабом для усіх таких частин функціоналу системи. Завдяки використанню CQRS технології, є можливість зробити coupling [8] менше, адже кожні окремі дії вже поміщаються в незалежні один від одного місцях. Також так як Domain логіка є спільною для всієї розподіленої системи, є можливість її винести

в окремий пакет, який можуть імпортувати собі інші сервіси. Прикладом технологій для цієї реалізації може бути npm пакети [9] який чудово підходить для NodeJS мікросервісів, NuGet [10] для .NET мікросервісів, sub-module git система.

Загалом, при розширенні певної функціональності, усе, що потрібно зробити - це перенести ізольований модуль в особистий сервіс, враховуючи те, що більшість залежностей вже відв'язані або знаходяться в загальному місці (рис.1).

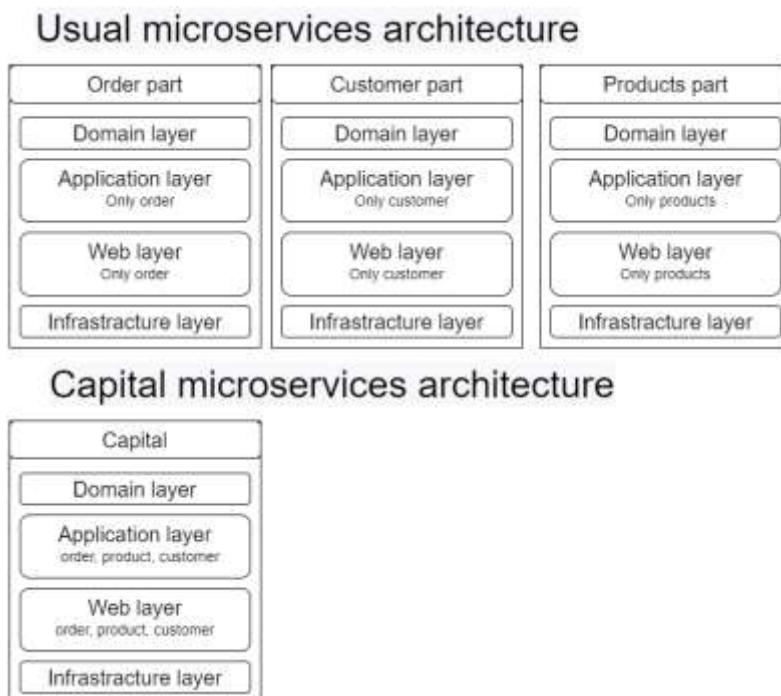


Рис. 1 – Фрагмент схеми мікросервісної архітектури

### Висновок

Розглянувши даний фрагмент схеми мікросервісної архітектури, можна зробити висновок, наскільки зменшилась кількість модулів необхідних для її реалізації, що на довготривалій перспективі дозволяє зберегти і час і можливість масштабування системи.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем - Эрик Эванс.: Пер. С англ. – М. : ООО «И.Д. Вильямс», 2021. -448с. ISBN 978-5-8459-1597-9.
2. Ричардсон Крис «Микросервисы. Паттерны разработки и рефакторинга» Питер, 2019 год, 544 стр., ISBN 978-5-4461-0996-8
3. Комплект из четырех книг. Чистая архитектура. Чистый код. Чистый Agile. Идеальный программист. - Роберт Мартин – 2020
4. Greg Young: [CORS Documents by Greg Young](#). (PDF) In: *WordPress*. November 2010, abgerufen am 13. Juli 2014 (englisch).
5. Искусство автономного тестирования с примерами на C#. 2-е издание / пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2016. – 360 с.: ил. ISBN 978-5-94074-945-5
6. Tapas Pal. Understanding Onion Architecture. [Електронний ресурс]. Режим доступу: <https://www.codeguru.com/csharp/understanding-onion-architecture/>. Дата звернення: грудень. 2021
7. Ньюмен С. От монолита к микросервисам: Пер. с англ. – СПб.: БХВ-Петербург, 2021. – 272 с.: ил. ISBN 978-5-9775-6723-7.
8. Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization 1st Edition - Susan J. Fowler – 2016
9. Node.js в действии. 2-е издание - Кантелон М.Хартер М.Головайчук Т.Райлих Н. – 2018
10. ASP.NET Core in Action, Second Edition - Andrew Lock – 2018

**Озеранський Володимир Сергійович** — кандидат технічних наук, старший викладач кафедри комп'ютерних наук, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця

**Ішук Владислав Вікторович** — студент групи 2КН-20м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: [vladyslav.ishchuk@gmail.com](mailto:vladyslav.ishchuk@gmail.com)

**Volodymyr Ozeransky** — Candidate of Science (Engineering), associate professor of Computer Science Department, Informations Technologies and Computer Engineering Faculty, Vinnytsia National Technical University, Vinnytsia

**Vladyslav Ishchuk** — student of Informations Technologies and Computer Engineering Faculty, Vinnytsia National Technical University, Vinnytsia, email : [vladyslav.ishchuk@gmail.com](mailto:vladyslav.ishchuk@gmail.com)