

УДК 612.8; 330.46 DOI: 10.14451/1.219.163

Реактивное программирование и микросервисы: описание методов использования реактивного программирования в архитектуре микросервисов и подробный анализ их эффективности

© 2023 **Гладун Анастасия Михайловна**

магистр по направлению «Прикладная математика и информатика», ведущий программист ООО «АТОН». Россия, Москва.

E-mail: netmislei@gmail.com

Реактивное программирование все чаще используется из-за его многочисленных преимуществ. В частности, это помогает избежать сбоев, вызванных проблемой проектирования, они больше не могут возникать, поскольку переменная не была обновлена. Время, которое экономится во время выполнения, значительно, и позволяет сосредоточиться на других задачах. Автор приходит к выводу, что микросервисы тесно связаны с предметно-ориентированным проектированием (Domain-Driven Design, DDD). Это такой подход к проектированию, при котором для программного обеспечения продумывается и постоянно развивается бизнес-модель — независимо от физической инфраструктуры, на которой все это работает. Всё чаще этот подход упоминается в связи с использованием реактивного программирования. Более того, что касается приложений, многие из них гораздо проще писать в реактивном программировании.

Вышесказанное позволяет сделать объективное заключение о том, что многие принципы проектирования DDD можно напрямую применять к системам на основе событий, в том числе не описанным в этой статье. В данной статье обо-

значены самые распространённые технические трудности: как выделить приложение в ограниченный контекст, почему так важны отдельные контексты, зачем нужны предметные модели и как всё это связано с обменом сообщениями, Apache Kafka и событиями.

Учитывая широкий набор доступных инструментов, архитектура микросервисов много выигрывает при использовании платформы потоковой обработки событий для разделения микросервисов. Можно использовать запросы, простые системы на основе событий, целые бизнес-приложения с потоками событий или комбинацию этих методов. DDD предоставляет фундамент, чтобы управлять взаимодействием между всеми компонентами, включая единый язык, ограниченные контексты, схемы и антикоррупционные слои. Как видно, события и сообщения помогают этим системам функционировать на практике.

Таким образом, создать архитектуру микросервисов можно разными способами, и предметно-ориентированное проектирование, несомненно, один из лучших подходов. Особенно если со-

здают системы со сложной предметной бизнес-областью.

Микросервисы не являются волшебной палочкой, но они решают ряд проблем, с которыми сталкиваются растущие компании при развитии ПО. Поскольку архитектура микросервисов состоит из независимо работающих модулей, каж-

дую службу можно разрабатывать, обновлять, развертывать и масштабировать отдельно от остальных. Обновления можно выполнять чаще, повышая надежность, время бесперебойной работы и производительность программного обеспечения. Архитектура микросервисов может оказаться полезной по мере роста организации и повышения требований к приложениям.

Библиографический список

1. Горбушин В. А., Самойлов Н. К. Особенности применения технологий в области реактивного программирования // / под ред. Д. Н. Борисова. — Воронеж : Вэлборн, 2022. — С. 42–48. — ISBN 978-5-6045486-5-3.
2. Карпович М. Н. Проектирование микросервисных архитектур информационных систем : материалы 86-й научно-технической конференции профессорско-преподавательского состава, научных сотрудников и аспирантов (с международным участием) // . — Минск, 2022. — С. 82–85.
3. Морозов Н. Д., Романов Д. Д., Пугин К. В. Введение в управление микросервисами // Современные проблемы лингвистики и методики преподавания русского языка в ВУЗе и школе. — 2022. — № 40. — С. 689–702.