

# МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*Михневич А. В.*

*Институт информационных технологий Белорусского государственного университета  
информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Скудняков Ю.А. – канд. техн. наук, доцент*

**Аннотация.** Проведен анализ возможностей микросервисной архитектуры для ее применения в процессе создания современного программного обеспечения.

Целью данной работы является осуществление обзора о возможностях, достоинствах и недостатках микросервисной архитектуры для ее применения в последующей разработке современного программного обеспечения.

Микросервисная архитектура – это шаблон проектирования системы, при котором структура приложения представляет собой набор небольших сервисов, объединенных в так называемые домены (определенная бизнес-задача в рамках всей системы) [1].

Данный подход позволяет разделить функционал системы на автономно развивающиеся модули, которые разрабатываются и обновляются независимо друг от друга. Таким образом, система может быть отмасштабирована любым нужным образом в контексте функционала – критичные части можно зарезервировать, высоконагруженным – выделить больше ресурсов. неполадки и отказы отдельных частей никак не повлияют на работу системы в целом – будут недоступны лишь отдельные функции системы.

Немаловажной особенностью является возможность использования в принципе разных технологий и языков в рамках разрабатываемого программного продукта – если это позволит оптимально решить поставленные задачи.

Микросервис (или просто Сервис) – это компонент системы, обладающий следующими характеристиками:

- небольшой размер;
- фокусирование на задаче;
- слабая связанность;
- простота использования;
- независимость развертывания.

Формализация контракта взаимодействия.

Принципы и особенности микросервисной архитектуры

Разработка системы на основе микросервисных доменов дает в итоге четкое разделение на контексты, автономные по части кода, данных и интерфейса взаимодействия с конечным потребителем (как правило, это пользователи системы). Это делает систему децентрализованной, команды могут сами решать, как и с помощью чего реализовывать задачи в рамках своей зоны ответственности (домена). Одновременно решается проблема сбоев – даже при неполадках в каком-то домене или сервисе – остальные части продолжают работать.

Однако системы на основе микросервисов имеют ряд особенностей, которыми нельзя пренебрегать.

Микросервисная архитектура тесно связана с автоматизацией процессов сборки и развертывания сервисов. **CI/CD и контейнеризация** – ключевая особенность микросервисной архитектуры. Она позволяет оперативно разворачивать новые сервисы или обновлять старые, менять количество запущенных экземпляров сервисов для увеличения быстродействия системы.

Еще одной немаловажной особенностью является необходимость в **различных средствах мониторинга** всей системы. Это разнообразные механизмы трассировки, логирования, оценки состояния сервисов и так далее. Это позволяет отслеживать состояние компонентов системы, вовремя реагировать на нестандартные ситуации и анализировать возникающие проблемы.

Так как в реальной жизни сервисам все-таки приходится взаимодействовать между доменами, в микросервисной архитектуре широко и часто используется **асинхронный способ передачи данных** через очередь сообщений (шину). Данный способ обеспечивает независимость работы сервисов в разрезе доменов и позволяет оповещать о каком-то событии любое количество заинтересованных элементов. Немаловажно и то, что асинхронная передача позволяет управлять нагрузкой на сами сервисы, являясь своего рода, балансировщиком (сообщения в шине могут находиться некоторое время, пока не будут обработаны сервисами) Достоинства микросервисной архитектуры

Исходя из характеристик микросервисов, архитектура на их основе обладает следующими преимуществами:

- возможность непрерывного обновления, сокращение time-to-market;
- автономность команд разработки;
- независимость набора технологий;
- масштабируемость;
- простой канал взаимодействия;
- стабильная спецификация обмена сообщениями;
- изолированность доменов;
- простота подключения и отключения сервисов;
- асинхронность обмена сообщениями помогает управлять нагрузкой на систему.

Недостатки микросервисной архитектуры:

- высокая сложность эксплуатации;
- высокая ресурсоемкость;
- сложности с определением границ доменов;
- большое количество артефактов и библиотек;

*58-я научная конференция аспирантов, магистрантов и студентов БГУИР, 2022 г.*

- необходимость в асинхронном канале передачи данных;
- невозможность обеспечения мгновенной согласованности данных;
- частичное дублирование данных.

В заключение необходимо отметить, результаты обзора возможностей микросервисной архитектуры позволяют сформулировать вывод о целесообразности ее использования для разработки и применения в различных сферах человеческой деятельности современного программного обеспечения.

***Список использованных источников:***

*1. Ньюмен, С. Создание микросервисов. / С. Ньюмен. — СПб.: Питер, 2016. — 304 с.*