

УДК 69

DOI 10.21661/r-556565

Кравченко Д.А.

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА

Аннотация: статья посвящена понятию микросервисов, как подходу к разработке программного обеспечения. Приведены и описаны такие понятия как, микросервисы, монолитная архитектура, микросервисная архитектура. Приведены характеристики и преимущества микросервисной архитектуры.

Ключевые слова: микросервисы, микросервисная архитектура, монолитная архитектура, характеристики микросервисов, преимущества микросервисов, разработка программного обеспечения.

Микросервисы – это архитектурный и организационный подход к разработке программного обеспечения, при котором программное обеспечение состоит из небольших независимых сервисов, взаимодействующих через четко определенные API. Эти сервисы принадлежат небольшим автономным командам.

Архитектуры микросервисов упрощают масштабирование и ускоряют разработку приложений, позволяя внедрять инновации и ускоряя вывод новых функций на рынок.

Монолитная и микросервисная архитектура

В монолитных архитектурах все процессы тесно связаны и работают как единая служба. Это означает, что если один процесс приложения испытывает всплеск спроса, необходимо масштабировать всю архитектуру. Добавление или улучшение функций монолитного приложения усложняется по мере роста базы кода. Эта сложность ограничивает экспериментирование и затрудняет реализацию новых идей. Монолитные архитектуры повышают риск доступности приложений, поскольку множество зависимых и тесно связанных процессов увеличивают влияние сбоя одного процесса.

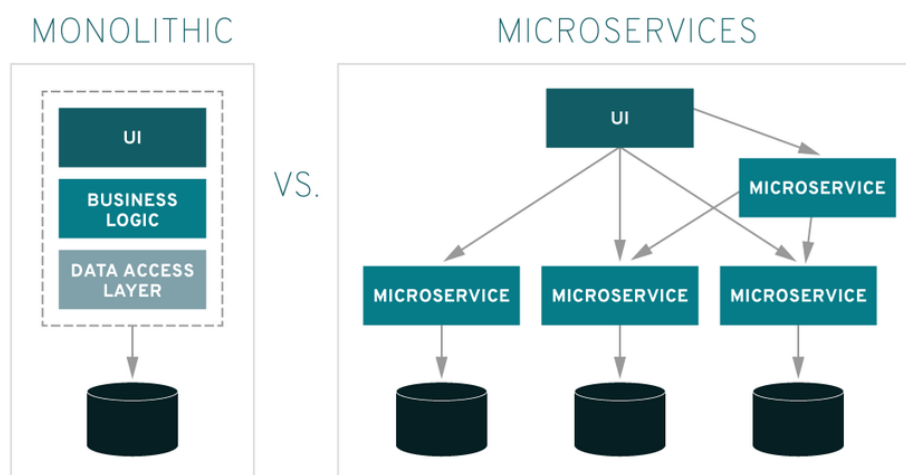


Рис. 1. Монолитная и микросервисная архитектура

В архитектуре микросервисов приложение строится как независимые компоненты, которые запускают каждый процесс приложения как службу. Эти сервисы взаимодействуют через четко определенный интерфейс с использованием облегченных API. Службы созданы для бизнес-возможностей, и каждая служба выполняет одну функцию. Поскольку они запускаются независимо, каждую службу можно обновлять, развертывать и масштабировать в соответствии со спросом на определенные функции приложения.

Характеристики микросервисов:

- Автономный

Каждую компонентную службу в архитектуре микросервисов можно разрабатывать, развертывать, эксплуатировать и масштабировать, не влияя на работу других служб. Сервисам не нужно делиться своим кодом или реализацией с другими сервисами. Любая связь между отдельными компонентами происходит через четко определенные API.

- Специализированный

Каждая услуга рассчитана на набор возможностей и ориентирована на решение конкретной задачи. Если разработчики со временем вносят в службу больше кода и служба становится сложной, ее можно разбить на более мелкие службы.

Преимущества микросервисов:

- Ловкость

Микросервисы способствуют организации небольших независимых команд, которые берут на себя ответственность за свои сервисы. Команды действуют в рамках небольшого и хорошо понятного контекста и имеют возможность работать более независимо и быстрее. Это сокращает время цикла разработки. Вы получаете значительную выгоду от совокупной пропускной способности организации.

– Гибкое масштабирование

Микросервисы позволяют независимо масштабировать каждую службу в соответствии со спросом на поддерживаемую ею функцию приложения. Это позволяет командам правильно определять потребности в инфраструктуре, точно измерять стоимость функции и поддерживать доступность, если спрос на услугу резко возрастает.

– Простое развертывание

Микросервисы обеспечивают непрерывную интеграцию и непрерывную доставку, упрощая опробование новых идей и откат, если что-то не работает. Низкая стоимость отказа позволяет экспериментировать, упрощает обновление кода и ускоряет вывод новых функций на рынок.

– Технологическая свобода

Архитектуры микросервисов не следуют универсальному подходу. Команды могут свободно выбирать лучший инструмент для решения своих конкретных задач. Как следствие, команды, создающие микросервисы, могут выбирать лучший инструмент для каждой задачи.

– Многоразовый код

Разделение программного обеспечения на небольшие четко определенные модули позволяет командам использовать функции для различных целей. Служба, написанная для определенной функции, может использоваться в качестве строительного блока для другой функции. Это позволяет приложению самопроизвольно загружаться, поскольку разработчики могут создавать новые возможности без написания кода с нуля.

– Устойчивость

Независимость от службы повышает устойчивость приложения к сбоям. В монолитной архитектуре отказ одного компонента может привести к отказу всего приложения. При использовании микросервисов приложения справляются с полным отказом службы за счет снижения функциональности, а не сбоя всего приложения.

Список литературы

1. Микросервисы (Microservices) [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/249183/>
2. Микросервисы и микросервисная архитектура [Электронный ресурс]. – Режим доступа: <https://www.atlassian.com/ru/continuous-delivery/microservices>
3. Микрослужбы.NET: Архитектура контейнерных приложений.NET [Электронный ресурс]. – <https://docs.microsoft.com/ru-ru/dotnet/architecture/microservices/>
4. Ричардсон Крис, Микросервисы. Паттерны разработки и рефакторинга/ Ричардсон Крис. – СПб, 2020. – С. 26–57.

Кравченко Даниил Андреевич –магистрант, ФГБОУ ВО «Санкт-Петербургский государственный морской технический университет», Санкт-Петербург, Россия.
