

Монолитная архитектура (без разделения, как спагетти-код или "комок грязи")

Определение: Монолитная архитектура — это подход к структурированию приложения как единого, не разделённого на части блока, где все компоненты взаимодействуют в рамках одного процесса.

Структура: В монолитной архитектуре, приложение строится как единый, цельный блок. Все функциональные элементы (пользовательский интерфейс фронтенд, серверная логика бэкенд и даже иногда базы данных и т.д.) интегрированы в одно приложение.

Взаимодействие компонентов: Компоненты в монолитном приложении тесно связаны и взаимодействуют друг с другом напрямую, без необходимости использования сетевых протоколов или API. Они обычно разделяют общую память и ресурсы.

Преимущества:

- Простота разработки: Монолитные приложения часто проще разрабатывать, так как все кодовая база находится в одном месте, и не требуется настраивать сложную систему взаимодействия между модулями.
- Простота развертывания: В отличие от микросервисов, монолитное приложение можно развернуть одним целым блоком, не требуется организовывать сложную инфраструктуру с несколькими сервисами, работающими вместе.
- Совместное использование ресурсов: В монолитном приложении, все компоненты могут легко делить ресурсы между собой, что может быть более эффективно с точки зрения производительности.

Недостатки:

- Сложность поддержки: Когда приложение растет, кодовая база становится все более сложной и трудной для понимания и поддержки, особенно без четкой модульной структуры.
- Ограниченная гибкость: При изменении одного компонента приложения вам, возможно, придется изменить и другие части, что может привести к "эффекту домино". Это проблематично, особенно в больших командах.
- Сложность масштабирования: Монолитные приложения могут быть сложными для масштабирования, так как придется масштабировать все приложение целиком, а не отдельные части.
- Риск сбоев: Если одна часть монолитного приложения выходит из строя, вся система может стать недоступной.

