

Давайте **выявим и решим компромиссы** относительно нашего продукта “Алёшка”.

Смотря на архитектурно значимые требования, можно выделить такие компромиссы (функциональные требования мы можем выполнить они несложные, поэтому рассмотрим НФТ):

- Производительность vs Стоимость: Для обеспечения высокой производительности (низкой задержки) в ответах чат-бота может потребоваться использование более мощной инфраструктуры или дорогостоящих технологий. Однако это может привести к увеличению стоимости проекта.
- Безопасность vs Удобство использования: Требования к высокой безопасности могут влиять на удобство использования системы. Например, если мы хотим защищать все данные пользователя, мы можем потребовать от них процесса аутентификации и авторизации (до обращения в чат), которые могут быть неудобны для пользователей.
- Масштабируемость vs Сложность системы: Обеспечение масштабируемости может потребовать использования микросервисной архитектуры, которая может быть сложнее в управлении и поддержке по сравнению с монолитной архитектурой (особенно учитывая что у нас немного функциональных требований и скорее всего нам подходит монолит).

Теперь нужно решить архитектурные компромиссы, чтобы спроектировать нужное нам решение. Тут могут быть разные варианты:

- либо вы со стейкхолдерами выясняете, какая характеристика важнее из компромисса (приоритеты), и двигаетесь в её сторону (например, делаете авторизацию несмотря на неудобства пользователей)
- либо вы находите решение, которое сможет на грани частично удовлетворить обе стороны

Для нашего проекта мы можем решить компромиссы так:

- Производительность vs Стоимость(сейчас важнее): Компромисс может заключаться в выборе более доступных технологий или инфраструктуры, которые обеспечивают приемлемую производительность в рамках бюджета проекта. На первом этапе запуска мы жертвуем стоимостью, так как мы за счёт недорого оборудования можем обеспечивать довольно долго нужные метрики

производительности.

- Безопасность vs Удобство использования(сейчас важнее): Компромисс может заключаться в использовании методов идентификации, которые обеспечивают хороший баланс между безопасностью и удобством использования (то есть пользователю нет необходимости авторизовываться на сайте, но необходимо представить своё имя и возможно доп.информацию по типу электронной почты).
- Масштабируемость vs Сложность системы(сейчас важнее): Можно начать с монолитной архитектурой, но проектировать ее с учетом возможности перехода на микросервисы в будущем по мере роста системы.

Отлично!

Получается, что вы, как архитектор, разобрались с требованиями, нашли компромиссы и даже решили их. Молодцы!