



Санкт-Петербургский государственный университет  
Кафедра системного программирования

# Переход ECS от нерективного монолита к реактивному монолиту.

Юсуп Амин Турмуди, группа 24.M71-мм

**Научный руководитель:** доцент кафедры системного программирования, к. Ф.-м. н.доцент Д. В. Луцив

Санкт-Петербург  
2025

- Развитие бизнеса через Интернет
- Проблемы возникают, когда пользователей становится больше

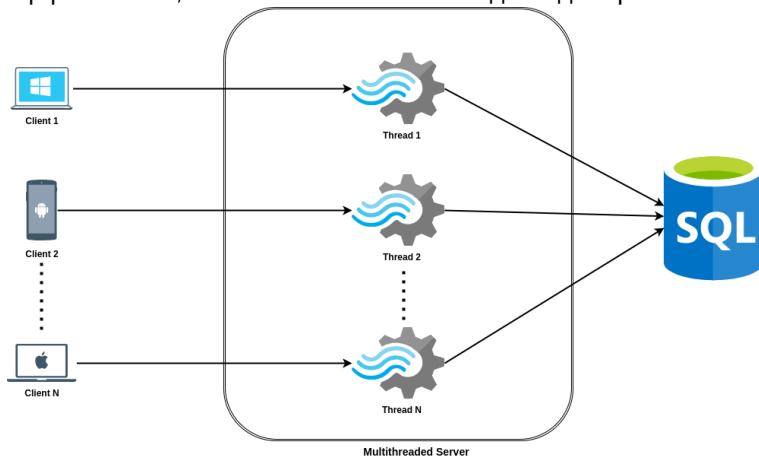
достижения этой цели были поставлены следующие задачи:

- Изучите существующие подходы к решению проблемы одновременных запросов.
- Внедрите фреймворк, обеспечивающий одновременные запросы.
- Проанализируйте и сравните время отклика, использование ЦП и памяти и измерьте, сколько запросов система может обработать в секунду.

- Многопоточность
- текущая архитектура
- вертикальное масштабирование
- горизонтальное масштабирование

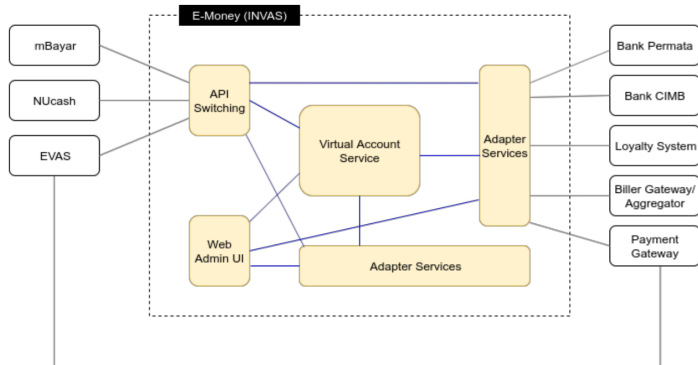
# Многопоточность

Многопоточность позволяет программе выполнять несколько задач одновременно, что может повысить производительность, особенно на многоядерных процессорах. В Java многопоточность используется для того, чтобы сделать программы быстрее и эффективнее, выполняя несколько задач одновременно.



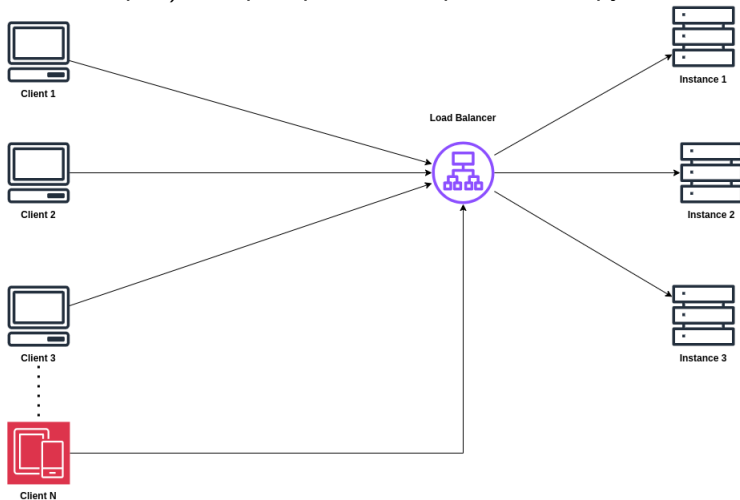
## текущая архитектура

Как автор уже писал во введении. Архитектура приложения emoney, которое в настоящее время работает, представляет собой монолитную архитектуру.

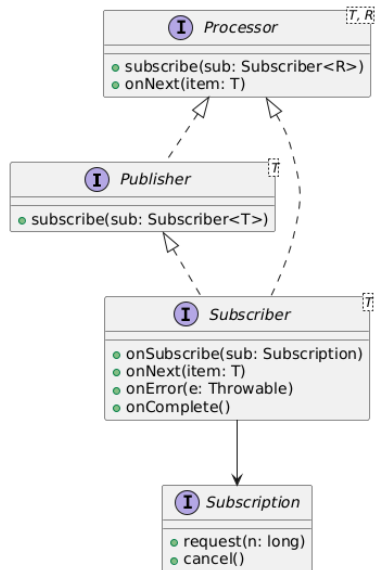


# горизонтальное масштабирование

Горизонтальное масштабирование (также известное как масштабирование наружу) означает добавление в систему большего количества машин (серверов, узлов или экземпляров) для распределения рабочей нагрузки



## Реактивное программирование.





Apache JMeter — это инструмент с открытым исходным кодом, используемый для тестирования производительности, нагрузочного тестирования и функционального тестирования приложений, особенно веб-приложений, API и серверов

# Определите основные метрики для измерения

- Задержка
- Коэффициент ошибок
- Использование ресурсов

В результате работы над учебными практиками были решены следующие задачи:

- ▶ Внедрение вертикального шкалирования.
- ▶ Горизонтальное шкалирование не проводилось, в связи с отсутствием средств на него.

Планирование дальнейшей работы:

- ▶ Реализация системы с использованием spring framework.
- ▶ Проанализируйте и сравните время отклика, использование ЦП и памяти и измерьте, сколько запросов система может обработать в секунду.