# OSPAB.VPN

Разработка устойчивого сетевого протокола на базе VLESS

---

**Автор:** Сыралёв Георгий (7 "М")

Гимназия №2, Великий Новгород

# ВАЖНО: ДИСКЛЕЙМЕР

## ⚠ Научное исследование

Данный проект является исключительно **научно-исследовательской работой**, посвященной изучению сетевых протоколов. Он не предназначен для коммерческого использования или нарушения законодательства РФ.

## Цель работы

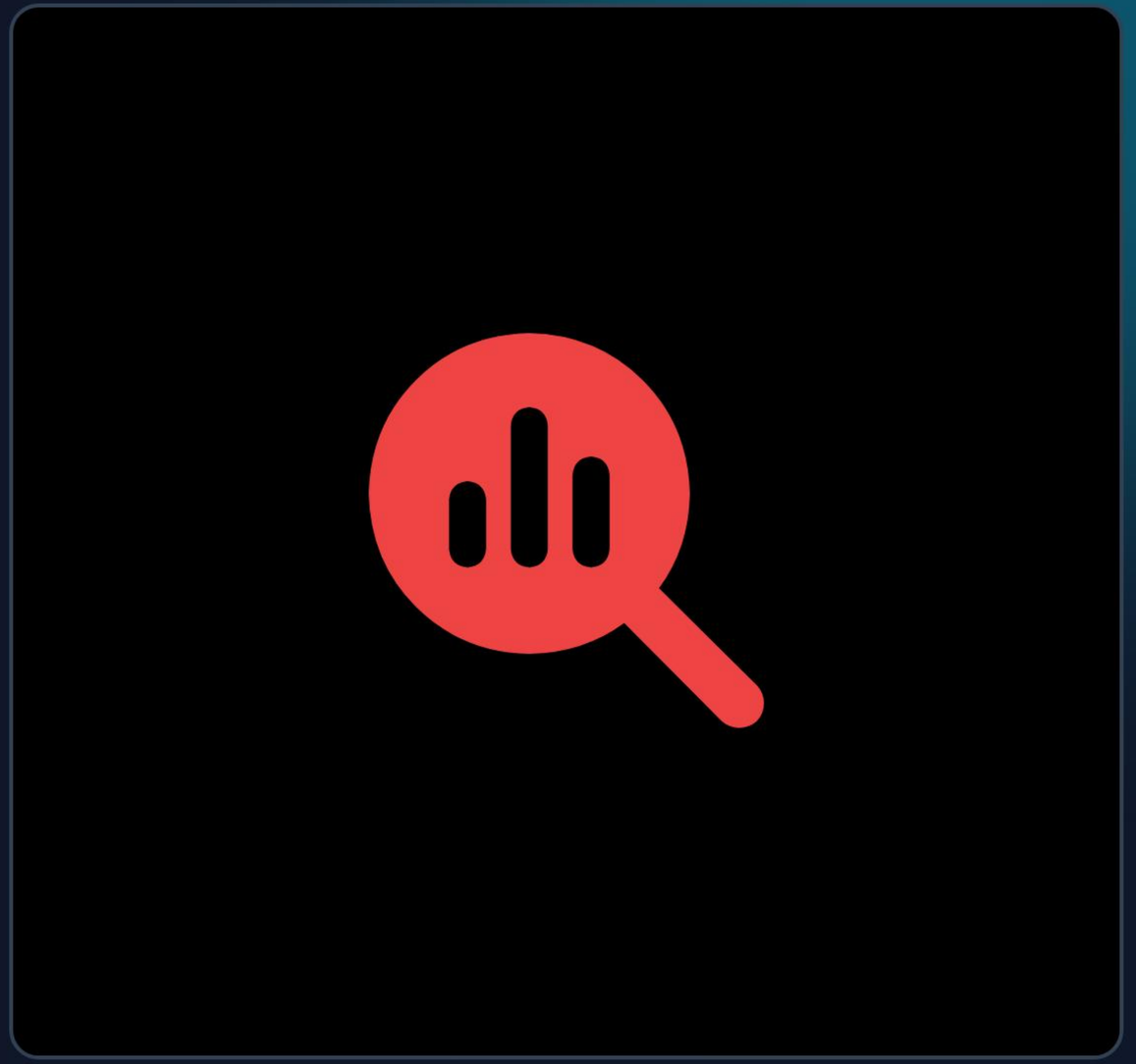
- Изучить методы фильтрации трафика.
- Разработать прототип с повышенной сетевой адаптивностью.
- Доказать концепцию на языке Python.

# ПРОБЛЕМА: СЕТЕВАЯ ФИЛЬТРАЦИЯ

Система **DPI** (Глубокий Анализ Пакетов) отвечает за **безопасность** и **соответствие** сетевого трафика установленным нормам.

Она работает как строгий контролёр, который проверяет все данные, проходящие через сеть.

> DPI идентифицирует и отфильтровывает трафик с подозрительными или несовместимыми сигнатурами.



# НЕДОСТАТКИ ТРАДИЦИОННЫХ ПРОТОКОЛОВ


Классические VPN протоколы такие как OpenVPN устарели с точки зрения адаптивности:

✕ Имеют **уникальные сигнатуры**, которые легко обнаружить.

✕ Используют **устаревшие методы маскировки**.

**Итог:** Протокол быстро идентифицируется и ограничивается.

# МОЁ РЕШЕНИЕ: МАСКИРОВКА ТРАФИКА

Мы используем метод мимикрии для **повышения адаптивности** протокола.

**Традиционный VPN**

Выглядит как VPN.

**ФИЛЬТРУЕТСЯ**

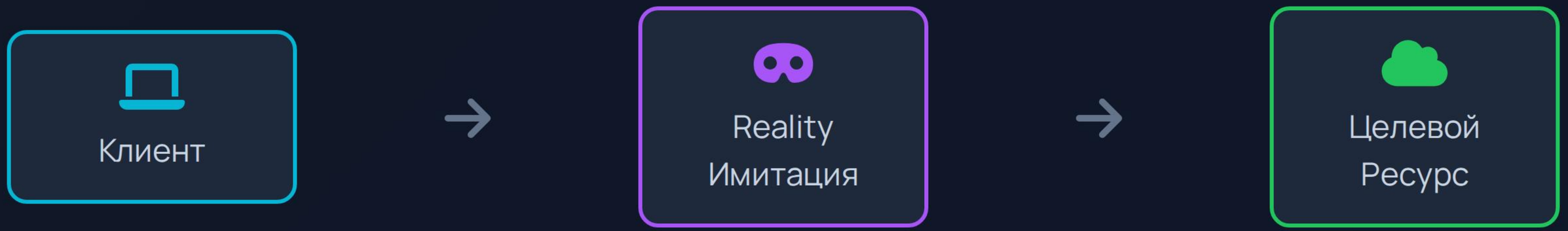
**Мой проект**

Выглядит как Google/Apple.

**ПРОПУСКАЕТСЯ**

# СХЕМА РАБОТЫ ПРОТОТИПА

Трафик имитирует стандартное HTTPS-соединение к легитимному ресурсу.




✓ DPI видит стандартный HTTPS-трафик и классифицирует его как доверенный.

# ПРЕИМУЩЕСТВА VLESS + REALITY

**Традиционный VPN**

- Низкая скорость
- Обнаруживаемая сигнатура
- Высокий расход энергии

**Мой подход**

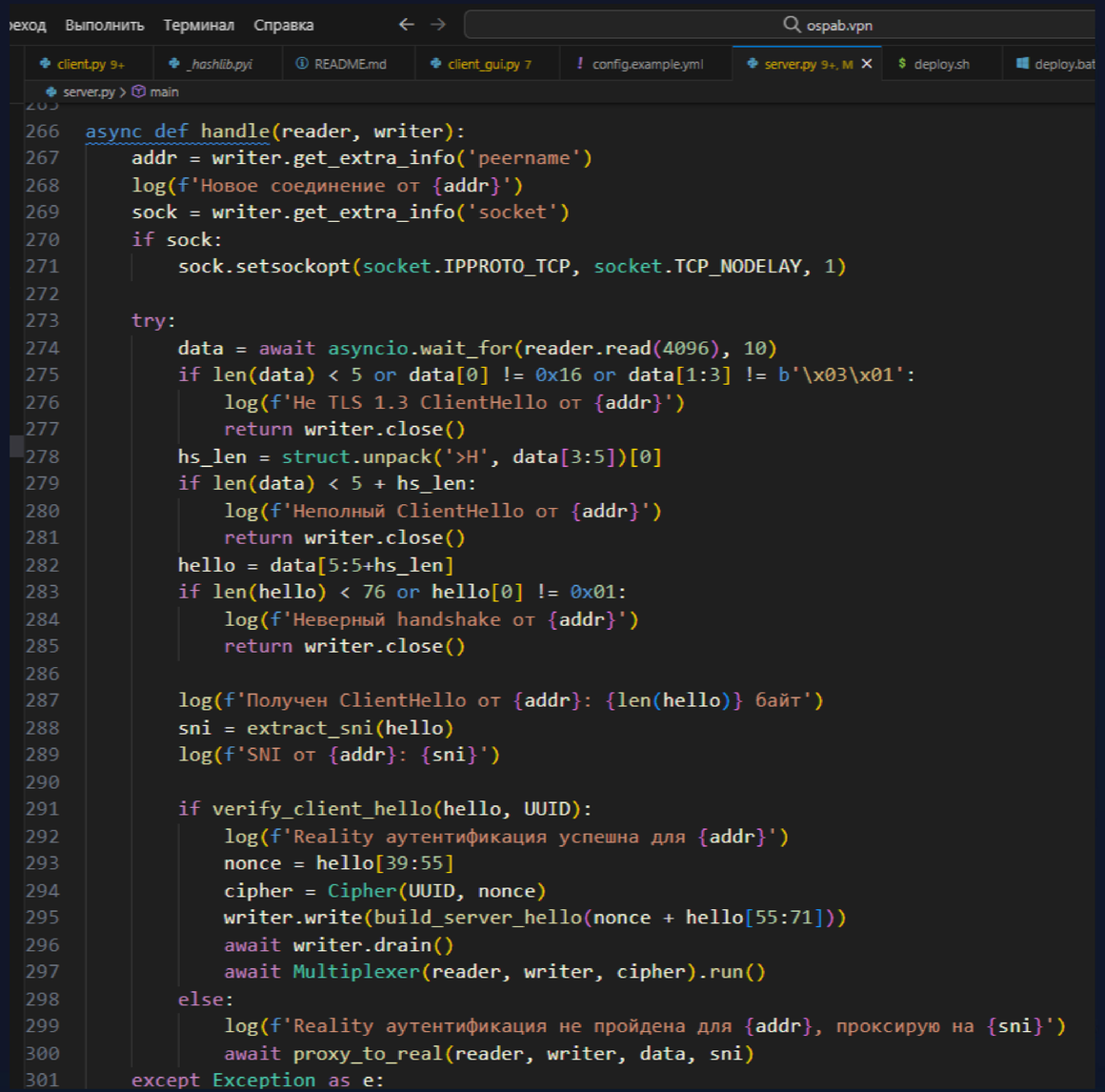
- **Высокая производительность**
- **Неопределяемая сигнатура**
- **Низкая нагрузка**

# АВТОРСКАЯ РЕАЛИЗАЦИЯ

Прототип был разработан на языке **Python**.

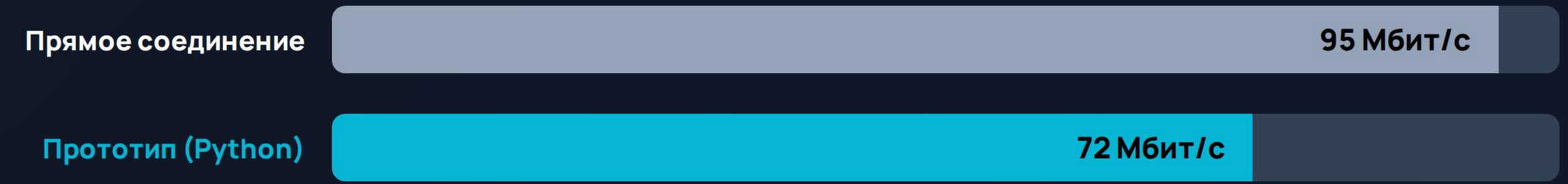
Это позволило детально изучить процесс создания сетевых сокетов и механизмов TLS-рукопожатия.

> Я использовал архитектуру VLESS как основу для своего прототипа, реализовав всю логику работы на чистом Python.



# РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Сравнение скорости передачи данных в Мбит/с:




Наблюдаемая потеря скорости (24%) обусловлена особенностями языка Python, что является важным выводом для дальнейших разработок.

# ЗАКЛЮЧЕНИЕ

## Проект успешно завершен

Разработан прототип, подтверждающий высокий уровень адаптивности сетевого трафика с использованием технологии Reality.

Спасибо за внимание!