# АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА RELY

Поставлена задача: **разработать архитектуру для программного комплекса управления рассылкой электронных сообщений.**

В процессе проектирования было принято следующее: система должна быть централизованной, т.е. все данные должны располагаться в центральном хранилище. Система должна иметь трехуровневую архитектуру, состоящую из следующих уровней: первый - клиент, второй - сервер, третий – хранилище в связи с ее масштабируемостью, конфигурируемостью, высокой безопасностью и надежностью, а также наличием большого объема документации.

## Клиентская часть

В процессе анализа было рассмотрено несколько альтернативных инструментов разработки web-интерфейса:

1. React;
2. Angular;
3. Vue.

Среди выбранных фреймворков предпочтение было отдано Angular благодаря его декларативности, наличию развитого сообщества, модульности и MVC из коробки. В качестве скриптового языка из-за ряда преимуществ над JavaScript был выбран TypeScript.

Для непосредственной вёрстки будет применяться классический набор инструментов: HTML + CSS.

## Серверная часть

Серверная часть включает в себя API, используемый для обработки запросов от клиентской части, набор функций для взаимодействия с хранилищем данных.

В результате проектирования было принято решение условно разделить серверную часть на ряд модулей:

1. модуль бизнес-логики;
2. модуль авторизации;
3. модуль балансировки.

Модуль бизнес-логики отвечает за работу с запросами, связанными с отправкой сообщений, просмотра журнала сообщений. Этот модуль взаимодействует с хранилищем данных для добавления, изменения и удаления клиентов, писем, сервисов доставки и других вспомогательных справочников; включает в себя реализацию очереди писем.

Модуль авторизации отвечает за авторизацию и аутентификацию, блокировку пользователей, а также за управление ролями зарегистрированных в системе пользователей.

Модуль балансировки обеспечивает корректную работу кластерной системы серверов: координирует работу нескольких серверов, позволяет проверять их работоспособность и эффективность их работы, назначать определенным единицам выбранные роли, собирает статистику запросов, позволяет задавать подходящий метод распределения нагрузки.

В качестве инструмента разработки серверной части была выбрана технология .NET Core, благодаря ее кроссплатформенности и удобству языка C#, а также благодаря детальной технической документации от официального разработчика.

## Хранилище данных

В ходе работы был проанализирован ряд различных СУБД, а именно:

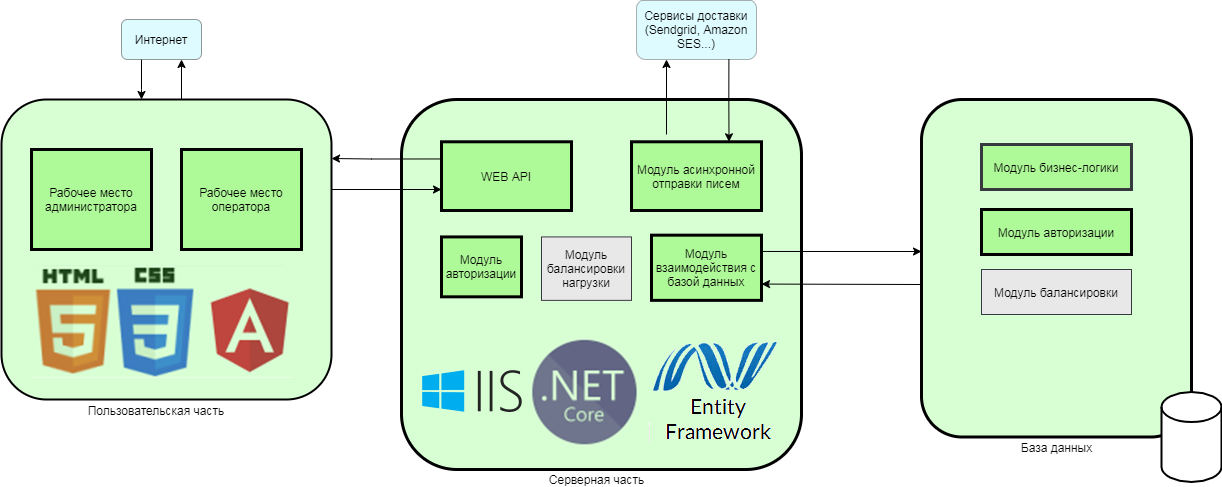
1. MySQL;
2. MS SQL;
3. PostgreSQL.

Предпочтение было отдано MS SQL в связи с его масштабируемостью и надежностью, высокой скоростью создания решений, а также возможностью обработки вычислений в оперативной памяти (in-memory OLTP).

В базе данных будут храниться все основные сущности системы:

1. сообщения;
2. пользователи;
3. сервисы доставки;
4. доступы и блокировки пользователей;
5. тарифы;
6. запланированные периоды
7. ссылки на прикрепленные файлы;
8. статусы доставки;
9. роли пользователей;
10. скидки.

## Схема архитектуры программного комплекса



**Рис. 1. Схема архитектуры проекта RELY**