

Рисунок 1. Схема "Клиент-Сервер"

**Назначение:**

1. Клиентская часть:

* **Frontend** взаимодействует с пользователем и предоставляет интерфейс редактирования. Валидация вносимых данных.
* **WebSocket** используется для двусторонней связи с сервером, что обеспечивает синхронизацию в реальном времени при наличии интернета.
* **Локальное хранилище** позволяет сохранять изменения в оффлайне и временно хранить данные до момента синхронизации.

1. Серверная часть:

* **SyncEngine** (синхронизирующий движок) обрабатывает изменения, отправленные клиентами и разрешает конфликты между пользователями.
* Сервер отправляет обновления другим клиентам через **WebSocket** или **API**, чтобы поддерживать актуальность данных.

1. Синхронизация изменений:

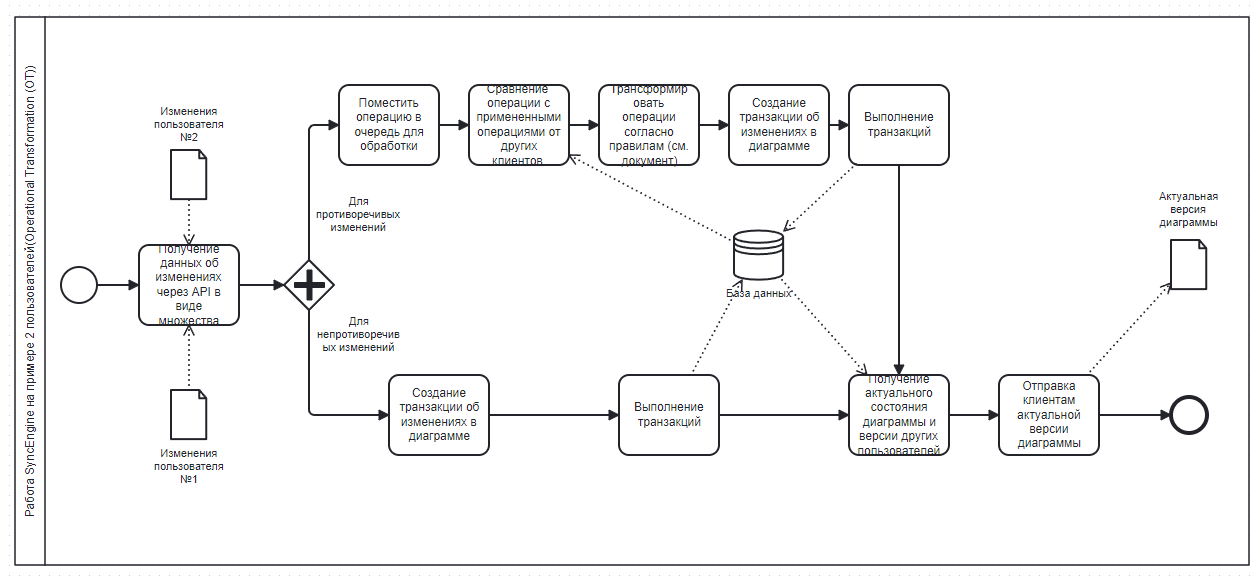
* Локальные изменения, сделанные в оффлайне, отправляются на сервер при восстановлении интернета. 

Рисунок 2. SyncEngine

**Правила трансформации операций:**

* Если две операции воздействуют на разные позиции, их можно применить независимо.
* Если две операции воздействуют на одну позицию, позиция новой операции должна быть скорректирована.
* **Insert vs Insert**: Если два символа вставляются в одну позицию, их порядок определяется по определенной стратегии (например, по ID клиента).
* **Insert vs Delete**: Если символ удален до вставки, операция вставки корректируется путем выбора актуальной версии (аналогично в git (merge))
* **Delete vs Delete**: Если два символа удаляются, дублирующие операции игнорируются.