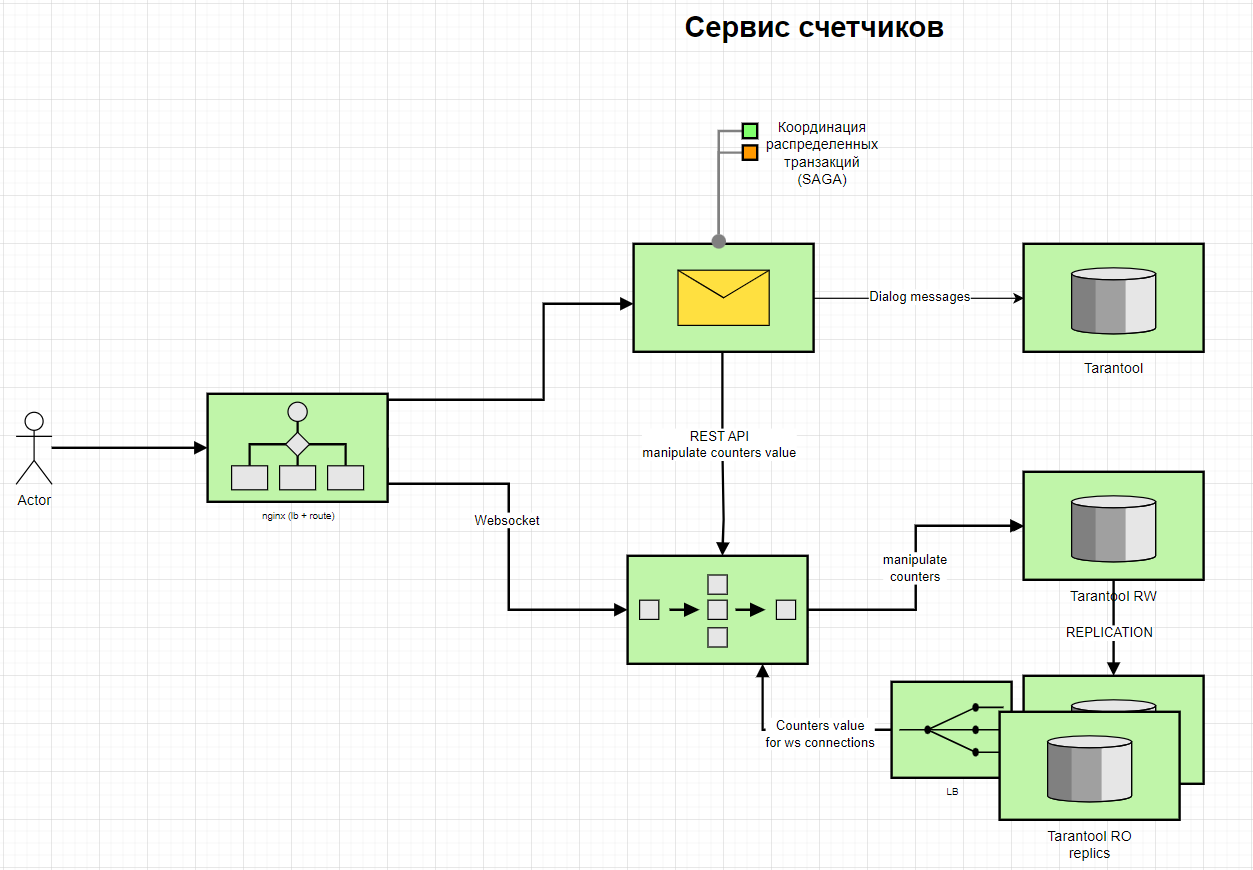
ДЗ: Сервис счетчиков

Для реализации сервиса счетчиков выбрана следующая схема:  


Сервис счетчиков:

* Сервис хранит число непрочитанных сообщений каждого пользователя часа
* на сервис предполагается большая нагрузка на чтение
* консистентность между счетчиком и реальным числом непрочитанных сообщений

Получение текущего значения счетчиков реализовано опроса по websocket, для регулярного опроса счетчиков и отображения в UI пользователя.

Для хранения использует in-memory СУБД Tarantool. Для распределения нагрузки на чтение предполагается использование кластера readonly реплик СУБД

Задачу обеспечения консистентности предлагается решить патерном SAGA - атомарные транзакции с компенсирующими изменениями в случае ошибки на очередном шаге (упрощенная реализация для учебного проекта в 2-3 шага в цепочке).

Цепочки обеспечения консистентности данных в БД счетчиков и диалогов:

1. **Отправка сообщения:**

* Создание счетчика, если его еще нет с значением 0
* increment\_counter(1)
  + if not increment\_counter – вернуть ошибку всей цепочки
* SendMessage() - (Поворотная транзация, если сообщение успешно отправлено, то цепочка успешная)
  + If not SendMessage.status – компенсация счетчика decrement\_counter(1)

1. **Чтение диалога:**

* Получить текущее значение счетчика -> unread
  + подсчет необходимого изменения счетчика с учетом offset:

change = unread – offset (если значение меньше 0, то изменение = 0)

* decrement\_counter(change)
  + if not decrement\_counter – вернуть ошибку всей цепочки
* GetDialoglist() - (Поворотная транзация, если чтение успешно, то цепочка успешная)
  + If not SendMessage.status – компенсация счетчиков increment\_counter(change)
  + If len(Dialoglist) < change (получили сообщений меньше чем изменили счетчик)
    - Компенсируем разницу increment\_counter(change – len(Dialoglist))