Бронирование билетов

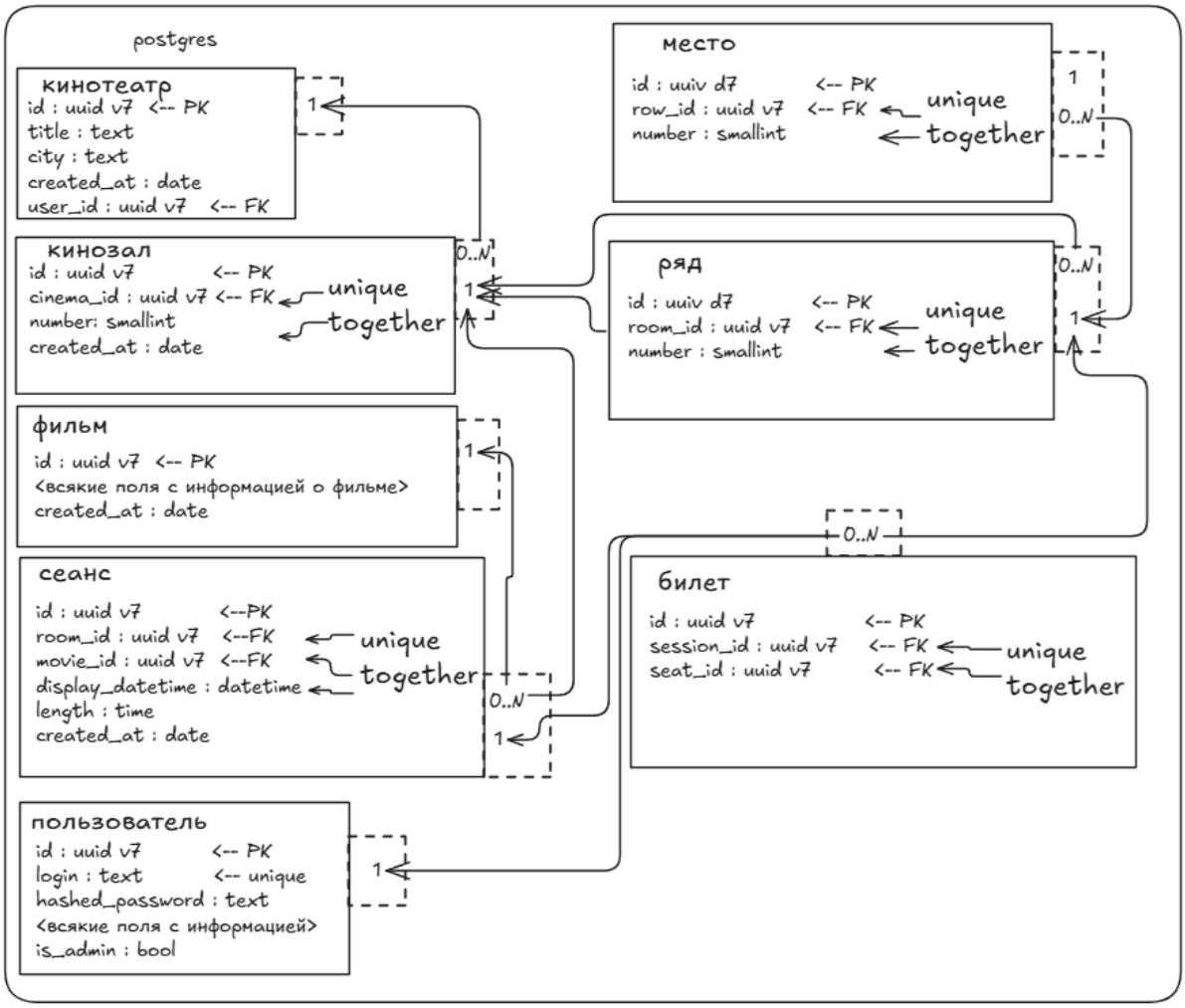
Функциональные требования:

1) получать расписание сеансов
2) покупать билеты (один или много)
3) много городов (и кинотеатров)

Нефункциональные требования:

1) 3 девятки
2) <500мс
3) согласованность данных
4) безопасность (это всё не ко мне)
5) одновременно:
* 10к пользователей
* 100 кинотеатров
* 1000 сеансов одновременно

Схема БД

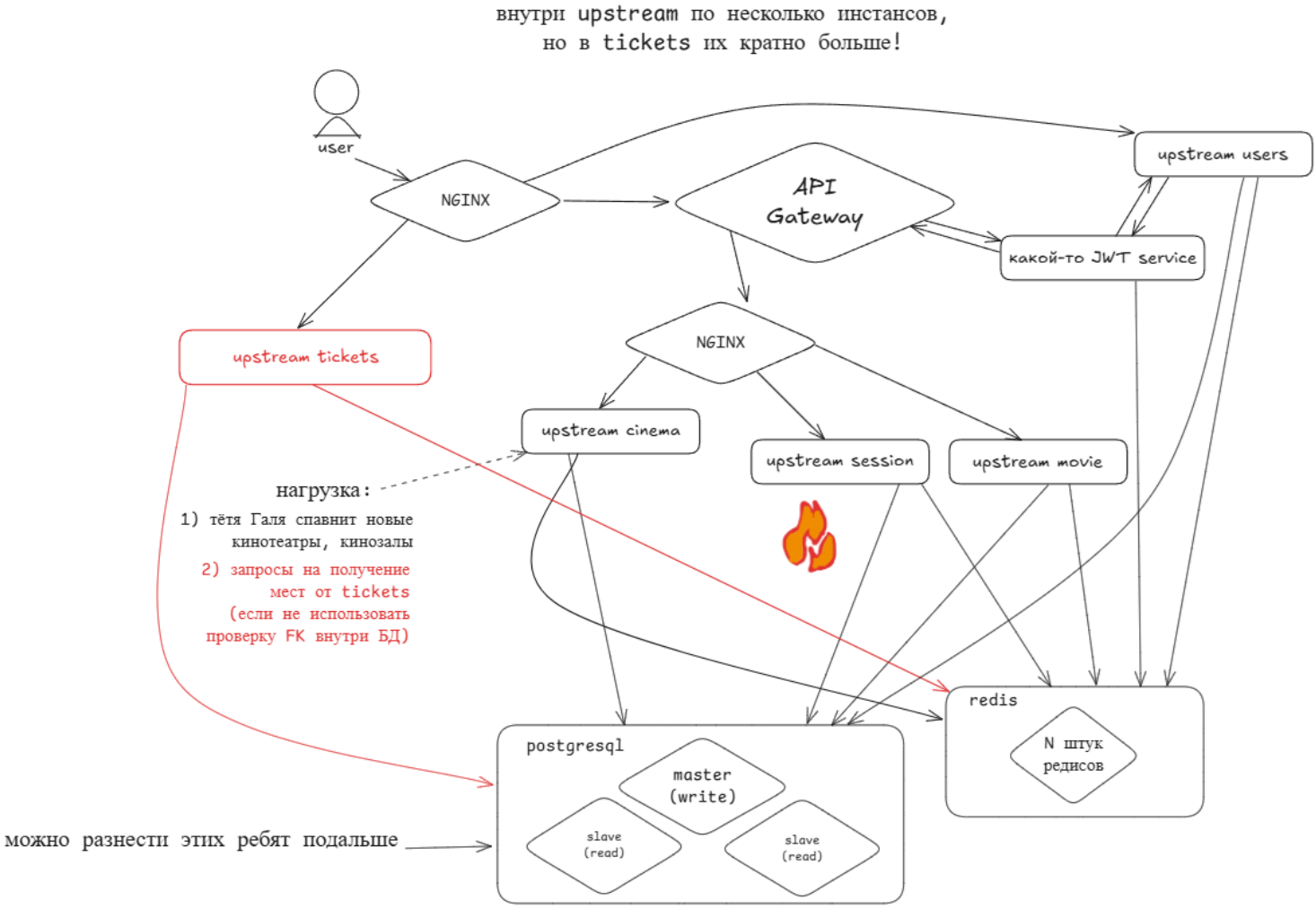


Функционал, если бы у сервиса был фронтенд:

Можно создавать свои кинотеатры, в них залы/ряды/места
Можно создавать фильмы и создавать показы фильмов
Можно покупать билеты
Можно получать данные обо всём что есть (как часть CRUD'ов)

Размещение

Всё поделено на микросервисы, а хостить их можно на разных машинах
*хостить по всему миру не нужно (максимум - в нескольких местах страны)



Многострадальный сервис оплаты

Tickets отвечает за покупку билетов, а это самая частая операция:
* на один фильм есть много сеансов
** а на один сеанс скупают сотни билетов
*** билеты надо проверять: не выкупили ли его уже

(это можно делать БД)
(но вдруг челам захочется разделить БД и сделать редис)

С БД есть 2 стула:

1) сделать одну схему БД с кучей таблиц
2) разделить все сущности какие есть по разным БД
(ага, и консистентность будут обеспечивать проверка через GET)

Куда сервис может расти?

Новые функции:

1) отмена билетов (мягкое удаление, например)

2) статистика

3) полные наборы GET для всех сущностей по всем зависимостям
(если тёте Гале в админ-панели кнопок побольше захочется)

4) хранение схем расположения мест в кинозале
(чтоб можно было на фронте строить планы застройки зала
для удобного выбора места)

5) поиск N соседних мест