



Урок 12 > Дизайн сервис такси

> Оглавление

> Оглавление

> Scope refinement

> Функциональные требования

Пешеход

Водитель

Дополнительно

> Нефункциональные требования

> Оценка нагрузки

Трафик

Запросы (load)

Сетевая нагрузка

Хранилище

> Затраты

> Высокоуровневый дизайн

> Дизайн компонент

Хранилища

> Масштабируемость и надежность

> Мониторинг и взаимодействие

> Scope refinement

1. Два типа пользователей сервиса
2. Входит поиск, совершение поездки и история поездок
3. Не входит поиск маршрута и расчет стоимости

> Функциональный требования

Пешеход

ДДД

1. Управление профилем (оплата, история поездок)
2. Выбрать точку старта поездки (текущая локация)
3. Увидеть машины поблизости и время ожидания
4. Выбрать пункт назначения и сделать заказ
5. Отслеживание и управление поездкой

Водитель

1. Управление профилем (оплата, история поездок)
2. Выход на смену и ее завершение
3. Принять входящий заказ поблизости
4. Отслеживание и управление поездкой

Дополнительно

1. Рейтинг водителей и пешеходов
2. Отслеживание спроса и прайсинг (ML)
3. Служба поддержки и контроль качества

> Нефункциональные требования

1. Доступность > консистентность
2. Пешеход быстро находит водителя
3. Водитель не простаивает без дела

4. Расстояние между пешеходом и водителем небольшое

> Оценка нагрузки

Трафик

1. 100M пешеходов всего
2. 4M водителей всего
3. В среднем за день 5% пешеходов оформляют по 2 поездки (5M daily active walkers (DAW) x 2 поездки) = 10M поездок в день
4. В среднем за день 1M активных водителей (1M daily active drivers)

Запросы (load)

1. (считая, что в дне около 100k секунд) 10M поездок / 100k секунд = 10k поисков в секунду
2. Поиск в среднем длится 10 секунд. 100k одновременных поисков
3. На каждый поиск опрашиваем 10 водителей = 1M соединений
4. (считая, что поездка в среднем длится 15 мин) 10M поездок / 24 часа / 4 части = 100k одновременных поездок
5. Каждые 5 сек водители сообщают свое местоположение. 0.2 R/D/s (запроса от водителя в секунду) x 1M водителей = 200k RPS

Сетевая нагрузка

1. Во время поиска водители отображаются на карте. Каждый водитель имеет 10 подписок на основе ближайших локаций: 10 подписок x 1M водителей = 10M
2. Предположим, что ID водителя вмещается в 4 байта, а локация - по 8 для lat и lng. По каждой подписке передаем эту информацию. 10M подписок x (4 + 8 + 8) = 200 MB/s
3. 200 MB/s x 100k секунд в дне x 2000 дней (в пяти годах) = 40PB информации

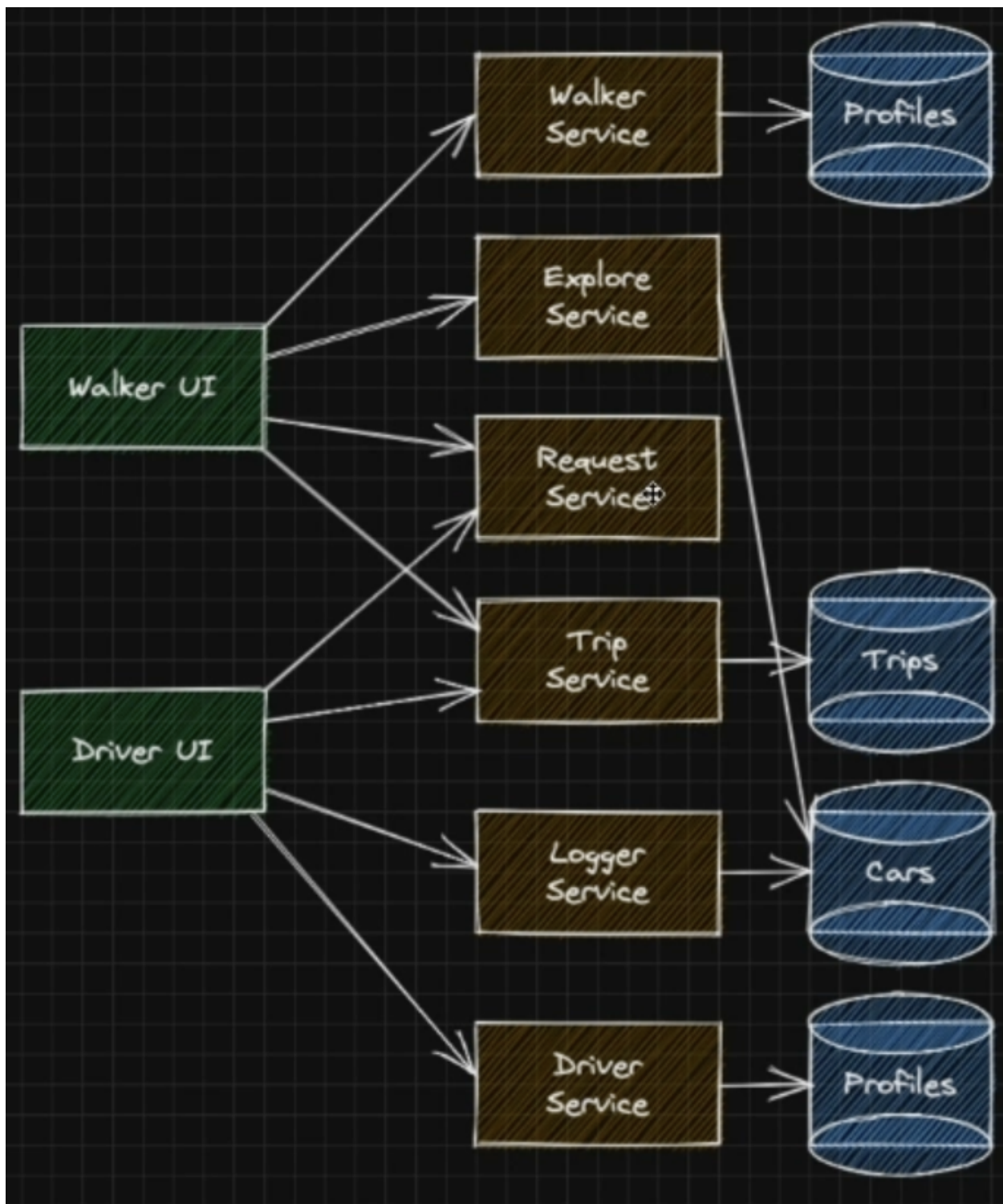
Хранилище

1. Профили водителей: $100\text{KB} \times 4\text{M водителей} = 400\text{ GB}$
2. Профили пешеходов: $100\text{KB} \times 100\text{M пешеходов} = 10\text{ TB}$
3. Местоположения: $1\text{M (активных водителей)} \times (4 + 8 + 8) + 5\text{M (активных пешеходов)} \times (8 \text{ (на ID, так как пешеходов больше)} + 8 + 8) = 200\text{ MB}$
4. Соединения по подпискам: $1\text{M} \times 4\text{B (на ID)} + 10\text{M} \times 8\text{B} = 100\text{ MB}$
5. Поездка: местоположение $(0.2\text{ R/D/s} \times 1000\text{ s (~15 мин)} \times 16\text{B} + 1\text{KB на инфо о водителе и тд)} \times 10\text{M поездок} \times 2000\text{ дней} = 100\text{ PB}$

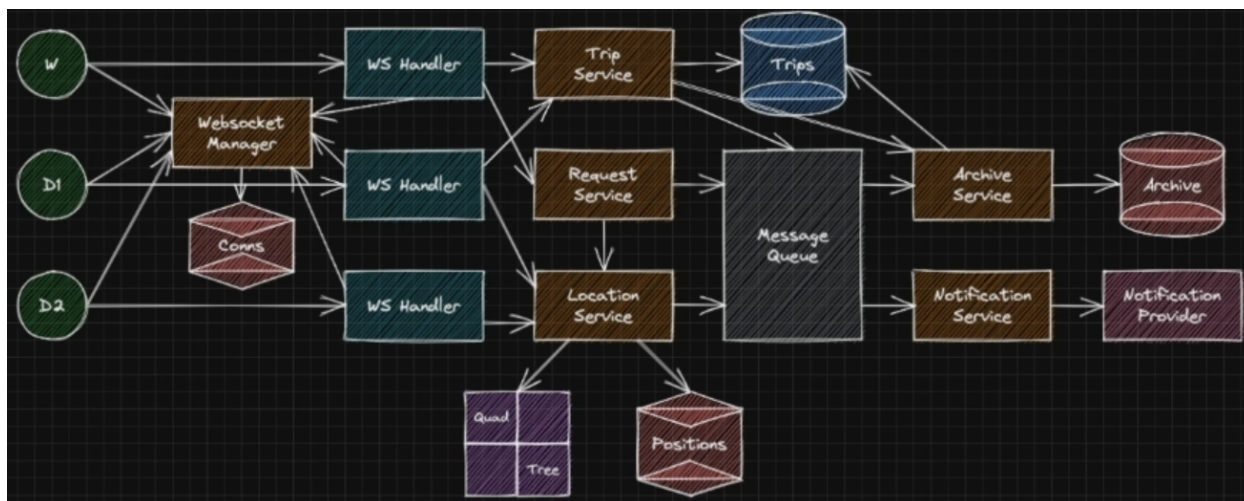
> Затраты

1. Conn/band: $100\text{ инстансов} \times \$100 \times 60\text{ месяцев} \times \text{перекрытие } 3 \text{ для надежности} = \2M на сервера
2. Трафик: $\$0.1 / \text{GB} \times 40\text{ PB} = \4M
3. Хранение: $\$300 / \text{TB} \times 200\text{ TB} = \$60\text{k на медленное хранилище} + \$20\text{k на быстрые хранилища}$
4. Итого около $\$6\text{M}$

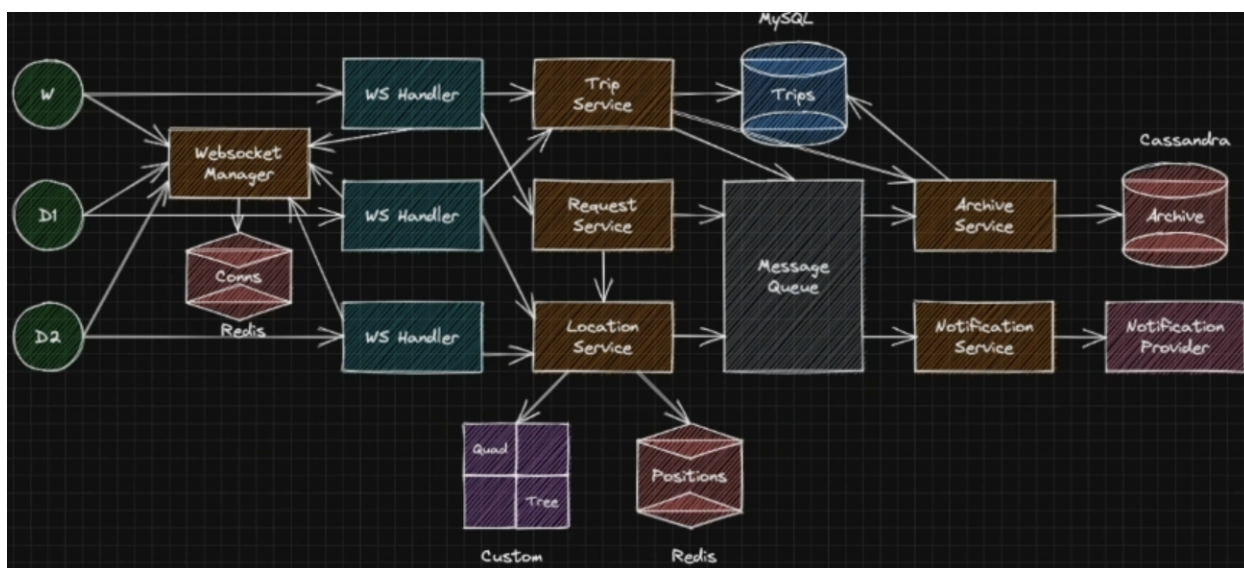
> Высокоуровневый дизайн



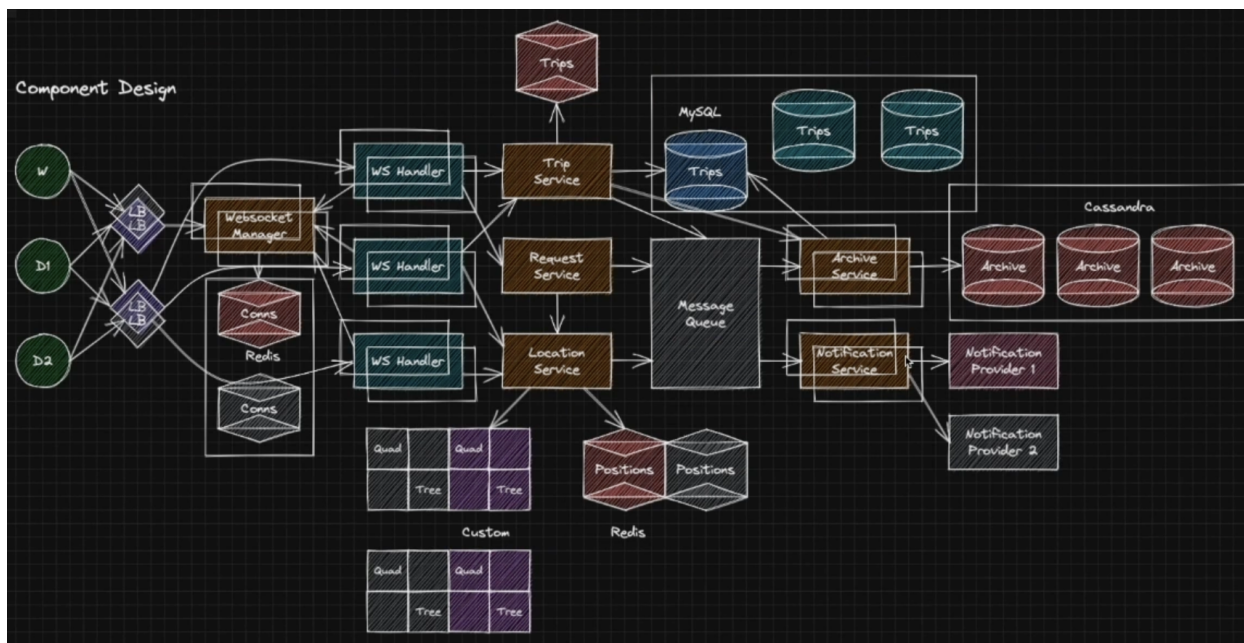
> Дизайн компонент



Хранилища



> Масштабируемость и надежность



> Мониторинг и взаимодействие

