Мультиагентная транспортная модель для ЧМ-2018 в России на Java за 4 месяца, с блэкджеком и эволюционными алгоритмами

Ярослав Смирнов
j.s@optimal-drive.ru
Facebook: Jaroslav.smirnov.9

Bio

- Образование RTOS & DCD
- 10 лет Developer
- 6 лет Java Developer

- Главный по науке и инновациям в лаборатории «Оптимальные транспортные системы» в Университете ИТМО.
- Сооснователь компании «OdgAssist» (LIMS & eQMS software)
- Проектировал и создавал транспортные макромодели городов и всего мира для крупных мероприятий.

План выступления

- Немного результатов, чтобы было понятно, о чём речь
- Общие слова про задачу
- Стек использованных технологий
- Проблемы и их решения
- Результаты

Дисклеймер

- Все цифры, связанные с проведением Чемпионата мира или Кубка конфедераций, показанные в данной презентации, не являются реальными, все они были специально изменены.
- Все цифры, связанные со стоимостью проекта, и шутки, связанные с этим, не имеют под собой никаких оснований и сказаны для красного словца.

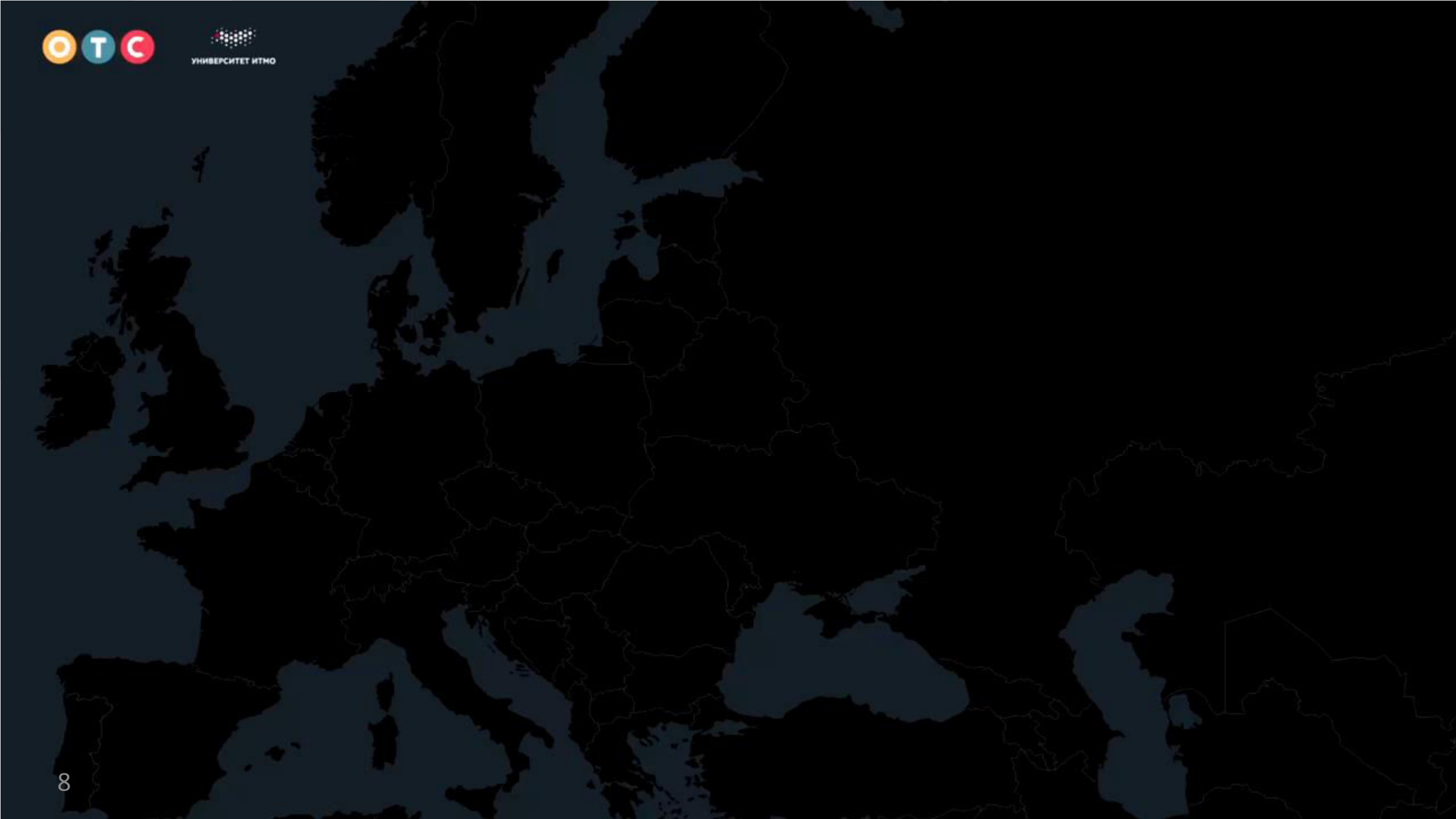
Для чего этот доклад?

- Уникальный проект написаный на JAVA
- Проект, который родился и сразу оптимизировался
- Может быть наши специфичные решения будут полезны в других отраслях

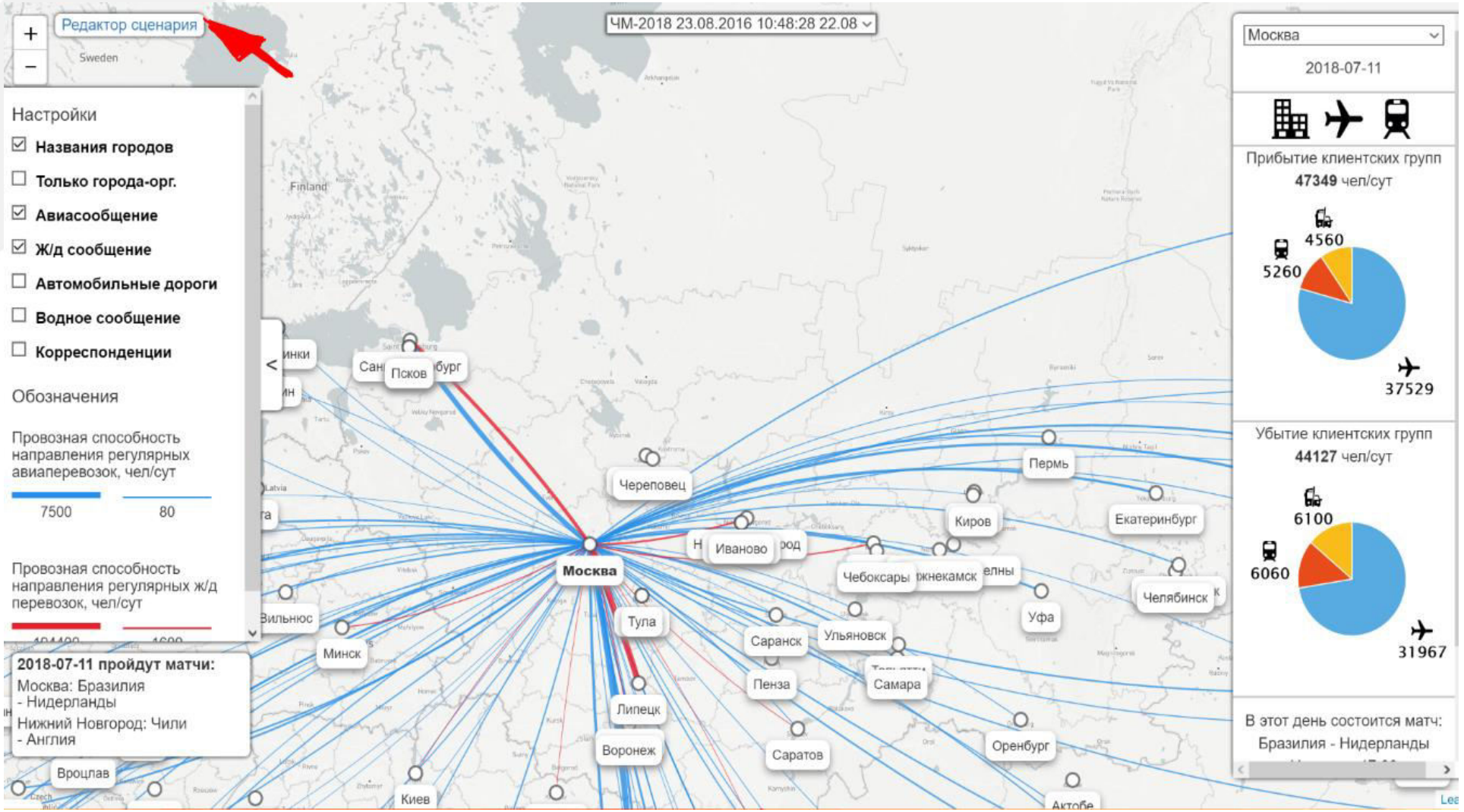
Попробуем описать задачу

- Хорошо провести Чемпионат мира по футболу
- 5 000 000 зрителей должны иметь возможность попасть на игры
- Найти ограничения, которые не позволят зрителям попасть на игры
- Произвести компьютерное имитационное моделирование и найти узкие места

Давайте начнём с конца.
Что получилось в результате ?


УНИВЕРСИТЕТ ИТМО

Пользовательская визуализация



Назначение системы

- Дать пользователям систему поддержки принятия управленческих решений по перевозке пассажиров на основе моделирования
- Учесть все ограничения: наличие транспорта, наличие гостиничных номеров, пропускные способности дорог
- Постепенно уточняя распределение играющих команд, получить нагрузку на разные транспортные узлы
- Найти проблемные места и не облажаться

Теперь немного назад.
За три месяца до этого...

Бэкграунд команды на тот момент

- Макромодели городов
- Микромодели участков городов
- Аналитика скоростных магистралей

Что от нас требовалось?

Описание задачи

Исходные данные

- Предыдущие чемпионаты
- Билетная политика FIFA

Модель

- Из каких стран поедут болельщики
- В какие города они поедут
- Где возникнут узкие места

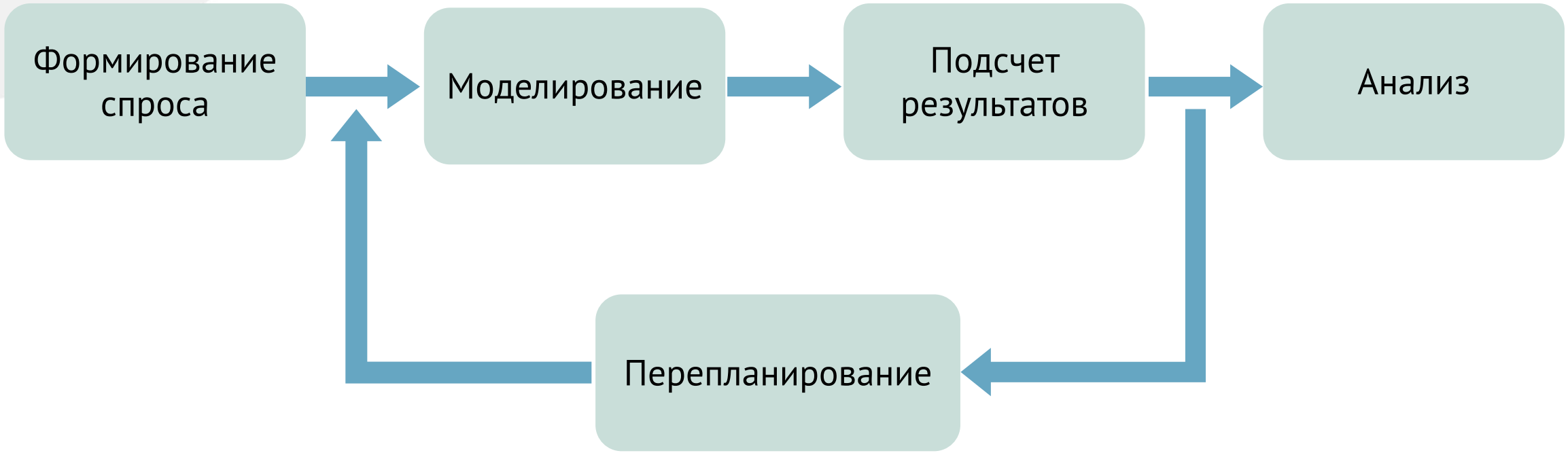
Приложение

- Ввод уточнённых данных
- Получение обновлённых результатов

Термины и определения

- **План** – последовательность из городов, которые хочет посетить болельщик. Начиная от страны вылета, через все игры, заканчивая возвращением домой
- **Роутинг** – процесс нахождения доступных видов транспорта и выделения на них билетов для каждого агента
- **Моделирование** – имитация реального траспортного мира с учётом всех возникающих ограничений: пропускной способности терминалов и взлетно-посадочных полос
- **Постпроцессинг** – обработка событий, произошедших в моделировании, получение итоговых результатов

Мультимодальное мультиагентное транспортное моделирование



План агента

Транспорт:

- Самолет
- Поезд
- Автобус
- Паром

Время прибытия/убытия:

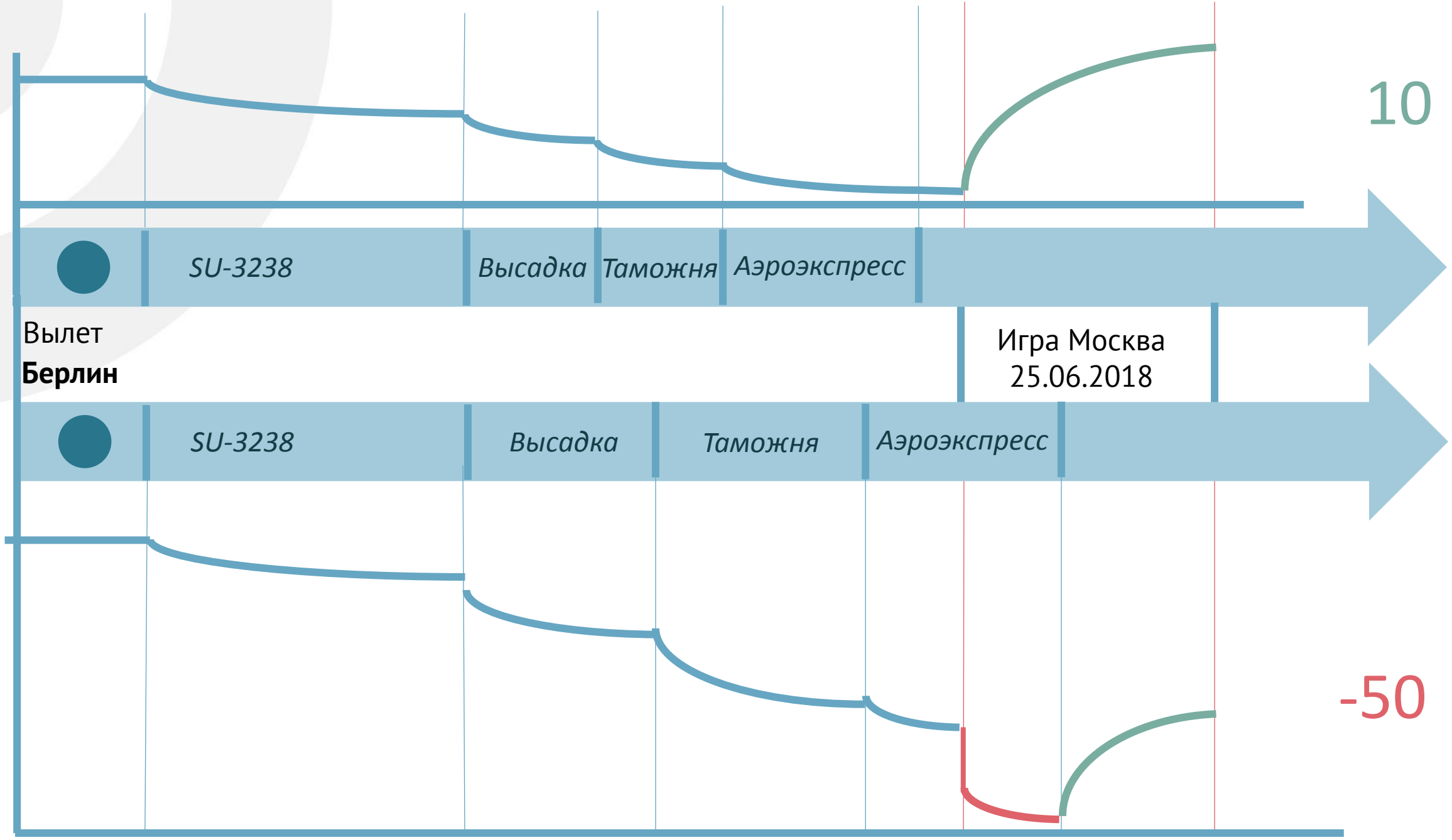
- Желание
- Гостиничный фонд
- Наличие возможности

Роутинг


Берлин
Москва
Санкт-Петербург
Москва
Берлин
SU-3238
Автомобиль
E-95
Автомобиль
E-95
SU-3239
Вылет
25.06.2018
10:00
Игра
25.06.2018
Приезд
26.06.2018
Игра
30.06.2018
Выезд
02.07.2018
Вылет
03.07.2018
13:00

Моделирование, набор очередей

- Агент сел в самолет SU-3238
- Самолет встал в очередь в Шереметьево на посадку
- Самолет встал в очередь на высадку
- Агент встал в очередь на прохождение границы
- Агент встал в очередь на получение багажа
- Агент прибыл в гостиницу
- Агент прибыл на игру
- Агент прибыл в гостиницу
- Агент встал в очередь на выезд из Москвы (пробка)
- Агент выехал на E-95
- Агент встал в очередь на въезд в СПб
- …


10
SU-3238
Высадка
Таможня
Аэроэкспресс
Вылет
Берлин
Игра Москва
25.06.2018
SU-3238
Высадка
Таможня
Аэроэкспресс
-50

Анализ

- Поиск дополнительных авиарейсов
- Поиск дополнительных ж/д отправлений
- Уничтожение ненужных авиарейсов

Исходные данные

- Количество зрителей по клиентским группам
- План продажи билетов клиентским группам
- Существующий мировой граф транспорта

Что является результатами

Последовательность событий:

- ActivityEndEventHandler - Матч или проживание в гостинице завершено
- ActivityStartEventHandler - Матч или проживание в гостинице начато
- GenericEventHandler - Дженериковое событие
- LinkEnterEventHandler - Попадание на граф транспортной сети
- LinkLeaveEventHandler - Покидание графа транспортной сети
- PersonArrivalEventHandler - Прибытие
- PersonDepartureEventHandler - Отправление
- PersonEntersVehicleEventHandler - посадка в машину
- PersonLeavesVehicleEventHandler - высадка из машины
- PersonStuckEventHandler - Зритель не может исполнить свой план
- TransitDriverStartsEventHandler - Самолет или поезд выехал из депо
- VehicleAbortsEventHandler - Транспортное средство было отменено
- VehicleEntersTrafficEventHandler - Транспортное средство пострадало из за ограничения
- VehicleLeavesTrafficEventHandler - Транспортное средство покинуло ограничение

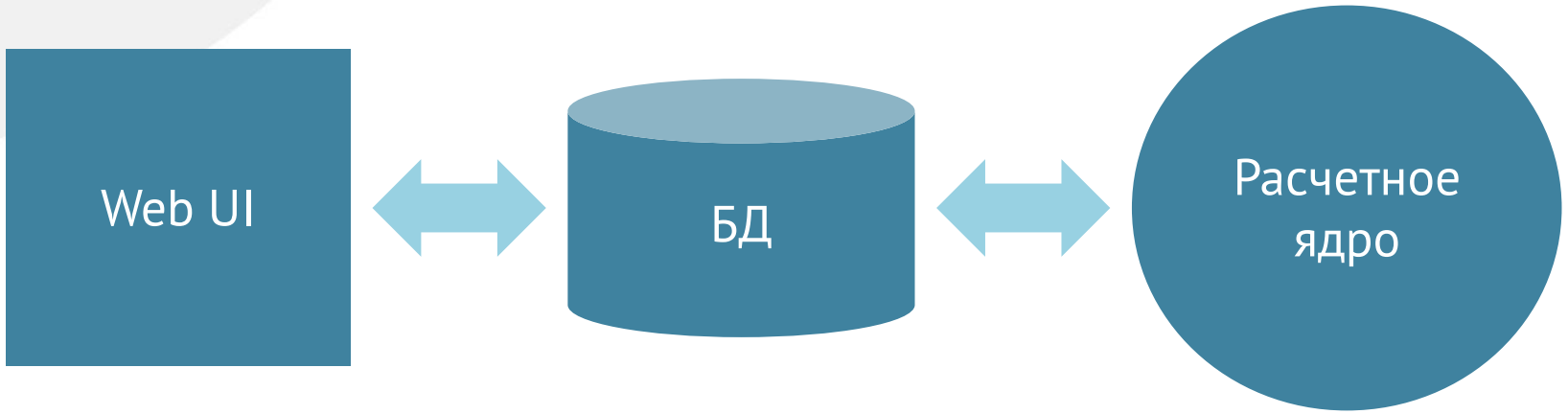
Спонтанные ограничения у самолетов

Откуда могут взяться ограничения у самолетов?

- Плохая погода
- Неисправность
- В город прилетает президент (небо закрывают на два часа)

Реализация проекта

Первая блок схема элементов



Postgresql

Vaadin

Matsim

Spring boot

Почему JAVA, SPRING Boot, VAADIN?

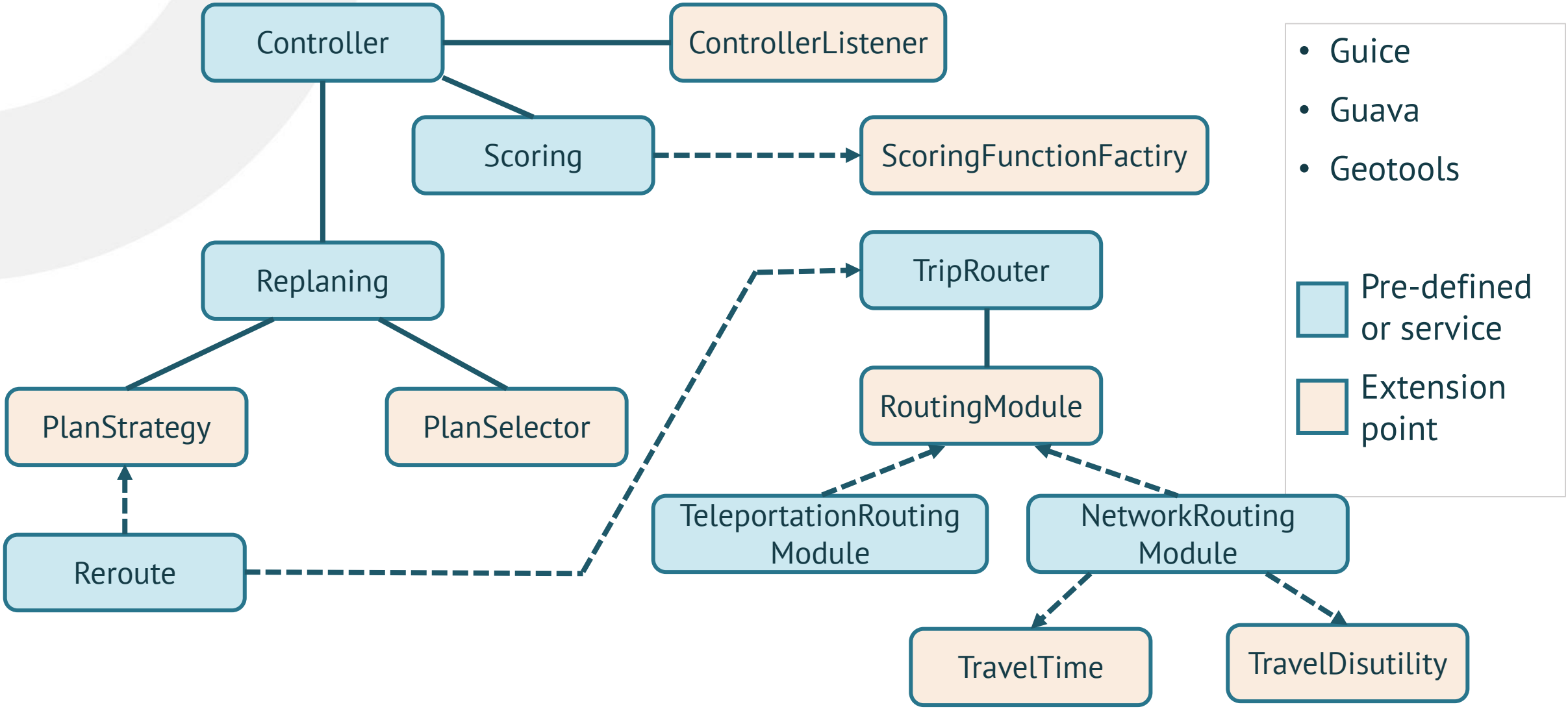
- Кросплатформенность, масштабируемость и разнообразие
- Дает возможность быть быстрым в разработке и быстрым в работе
- Spring Boot дает быстрое расширение технологий
- Vaadin дает быструю разработку

Мы также использовали TDD на спецификацию

и код ревью

Расчётное ядро

Библиотека MATSim и ее стек технологий



Как происходит моделирование шаг за шагом: полное описание процесса

Ветвь графа

Отстойник

Capacity – пропускная способность
Единиц в час

Freespeed –время без ограничений

Моделирование очереди (одна итерация)

T < tmax

Пройти по всем агентам и переместить в отстойник по времени

Пройти по всем узлам и переместить агентов на следующую ветвь графа

T++

Моделирование в несколько потоков

T < tmax

Случайное распараллеливание агентов

Пройти по всем агентам и переместить в отстойник по времени

Пройти по всем агентам и переместить в отстойник по времени

Случайное распараллеливание ветвей графа

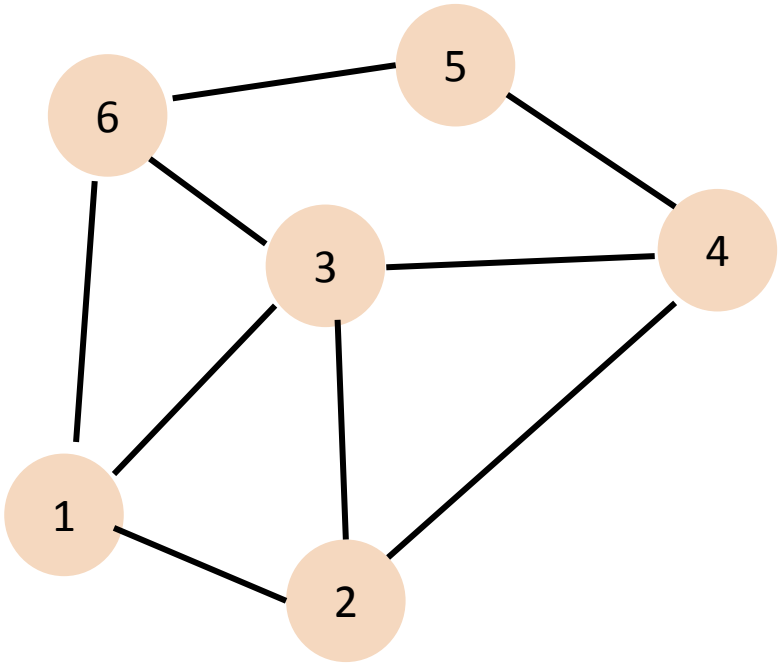
Пройти по всем узлам и переместить агентов на следующую ветвь графа

Пройти по всем узлам и переместить агентов на следующую ветвь графа

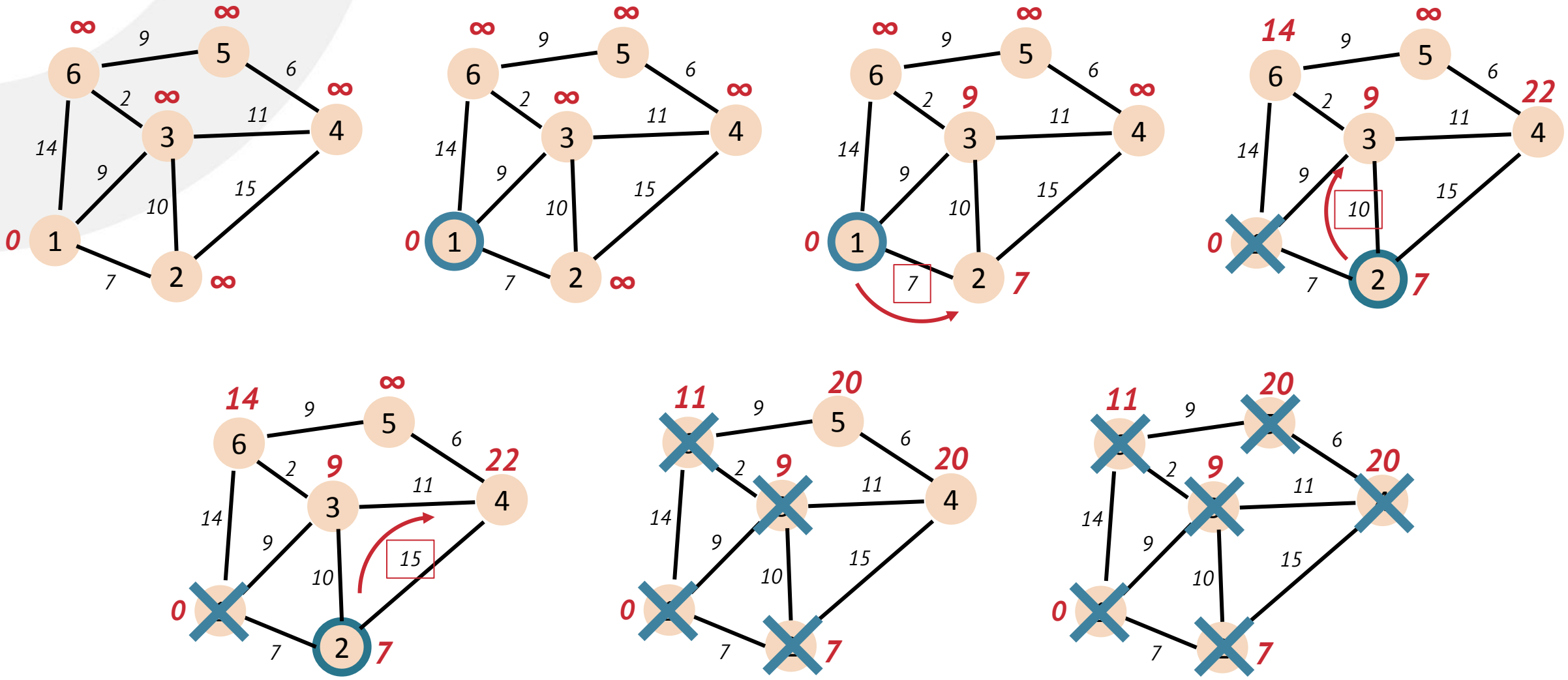
T++

Роутинг. Нахождение пути

Задача поиска маршрута

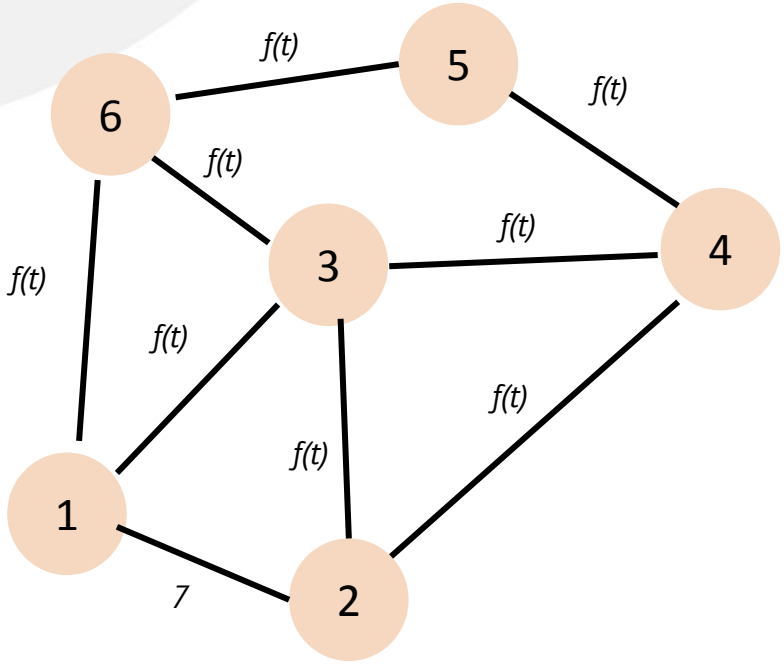


Как работает алгоритм Дейкстры?

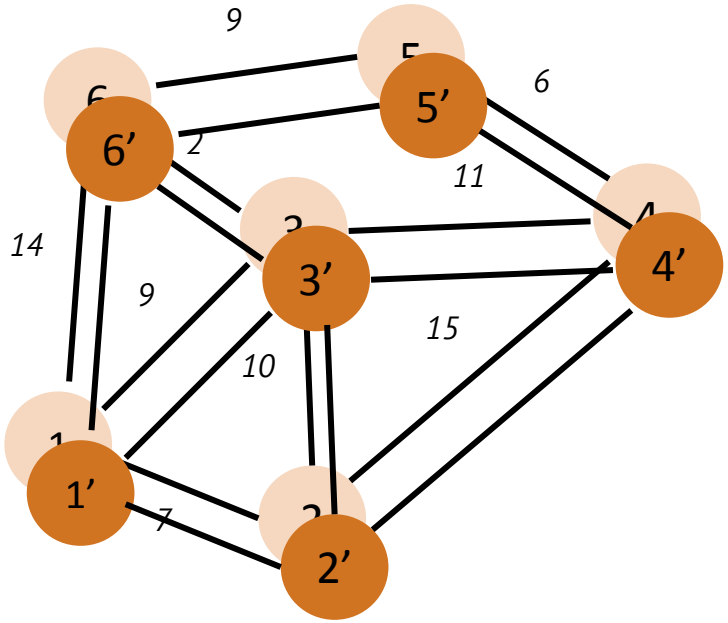


Как применить Дейкстру к транспорту

Время-зависимый подход:

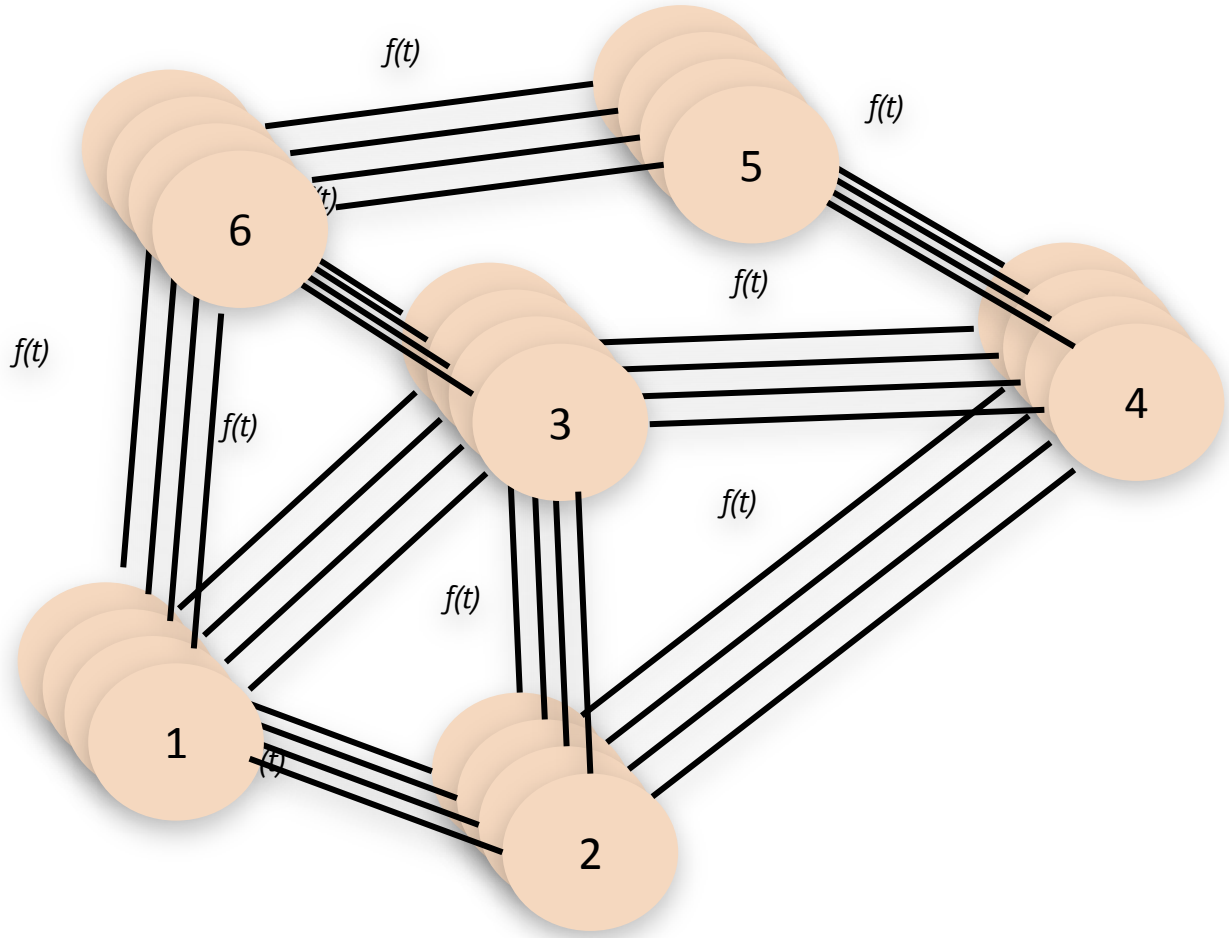


Время-расширительный подход:



Мультикритериальность

- Позволяет учесть при максимальном количестве пересадок в 3



Почему это может быть долго?

Для Чемпионата мира:

- 250 000 вылетов
- 1000 станций
- 5 000 000 агентов

Время-зависимый подход:

n = 1000, m = 999 000

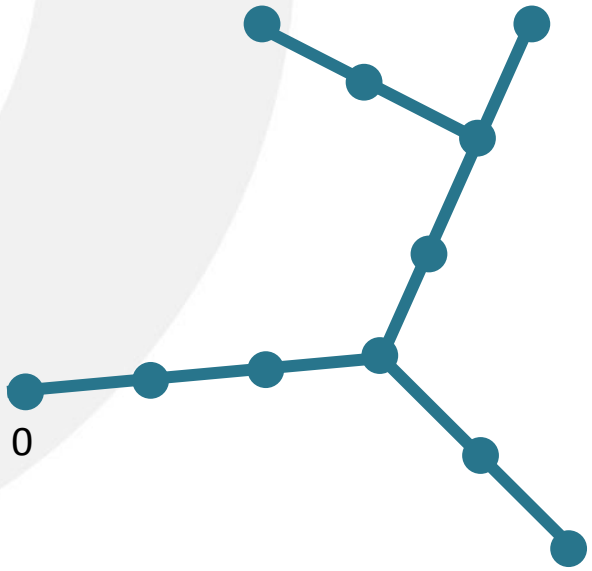
Время-расширительный подход:

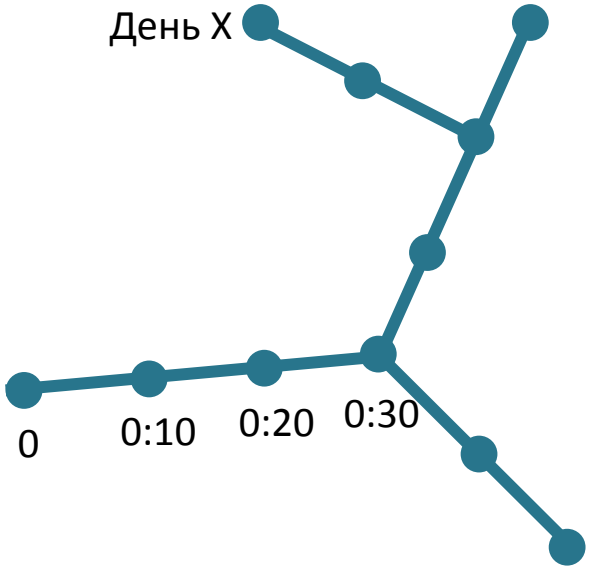
n = 250 000, m = 999 000

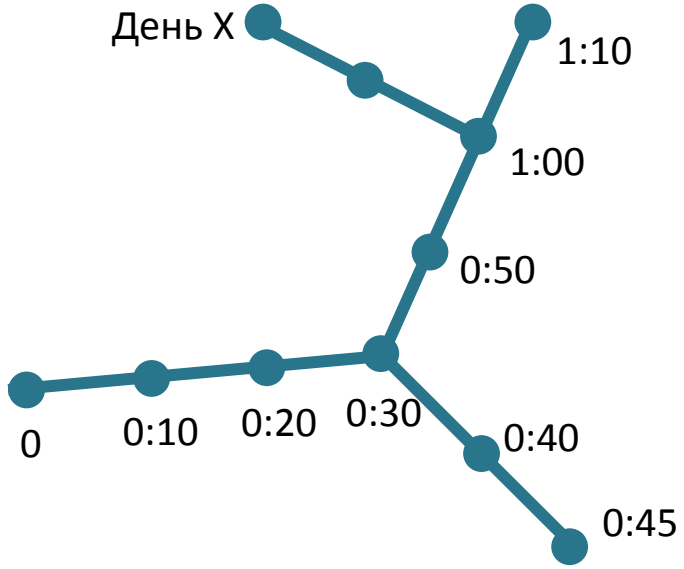
- Вычислительная сложность $O(n^2)$
- Для разреженных графов можно применить кучу. В случае с Фибоначиевой кучей сложность $O(n \log n + m)$

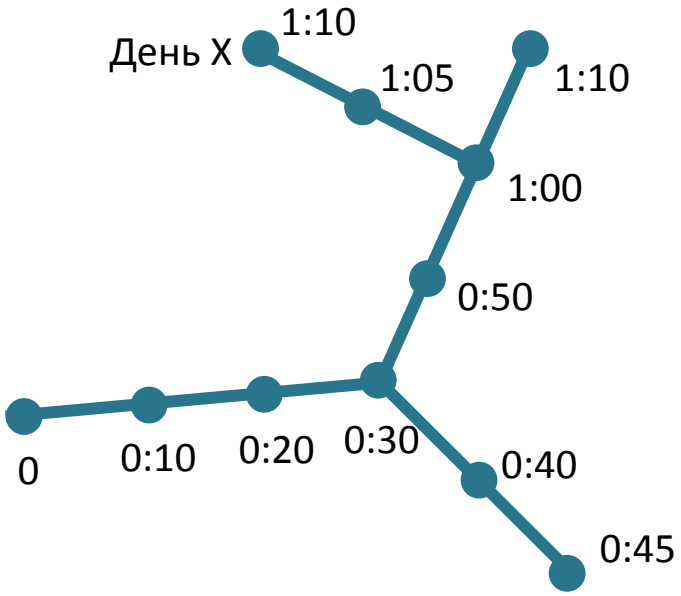
Что можно изменить

- Кэшировать результаты по типу точка-точка
- Сколько может быть пересадок в международном сообщении?
- 1
- 2
- 3
- А зачем вообще использовать Дейкстру?


0


День Х
0
0:10
0:20
0:30


День Х
1:10
1:00
0:50
0
0:10
0:20
0:30
0:40
0:45


1:10
День Х
1:05
1:10
1:00
0:50
0
0:10
0:20
0:30
0:40
0:45

Описание псевдокодом

Input: Source and target stops p_s, p_t and departure time τ.

// Initialization of the algorithm
1 **foreach** i **do**
2 $\tau_i(\cdot) \leftarrow \infty$
3 $\tau^*(\cdot) \leftarrow \infty$
4 $\tau_0(p_s) \leftarrow \tau$
5 mark p_s
6 **foreach** $k \leftarrow 1, 2, \ldots$ **do**
  // Accumulate routes serving marked stops from previous round
7 Clear Q
8 **foreach** *marked stop* p **do**
9 **foreach** *route* r *serving* p **do**
10 **if** $(r, p') \in$ Q *for some stop* p' **then**
11 Substitute (r, p') by (r, p) in Q if p comes before p' in r
12 **else**
13 Add (r, p) to Q
14 unmark p
  // Traverse each route
15 **foreach** *route* $(r, p) \in$ Q **do**
16 $t \leftarrow \perp$ // the current trip
17 **foreach** *stop* p_i *of* r *beginning with* p **do**
  // Can the label be improved in this round? Includes local and target pruning
18 **if** $t \neq \perp$ *and* $\text{arr}(t, p_i) < \min\{\tau^*(p_i), \tau^*(p_t)\}$ **then**
19 $\tau_k(p_i) \leftarrow \tau_{\text{arr}}(t, p_i)$
20 $\tau^*(p_i) \leftarrow \tau_{\text{arr}}(t, p_i)$
21 mark p_i
  // Can we catch an earlier trip at p_i?
22 **if** $\tau_{k-1}(p_i) \leq \tau_{dep}(t, p_i)$ **then**
23 $t \leftarrow \text{et}(r, p_i)$
  // Look at foot-paths
24 **foreach** *marked stop* p **do**
25 **foreach** *foot-path* $(p, p') \in \mathcal{F}$ **do**
26 $\tau_k(p') \leftarrow \min\{\tau_k(p'), \tau_k(p) + \ell(p, p')\}$
27 mark p'
  // Stopping criterion
28 **if** *no stops are marked* **then**
29 **stop**

Изменение скорости роутинга

Алгоритм	Время, мс
RAPTOR	7.3
TD	14.2
LD	44.5
MLC	67.2

Городской транспорт сильно отличается от международного

В чем проблема?

И стандартная реализация Дейкстры,

и стандартная реализация Раптора

не учитывают, что мест может уже не быть

Как учесть количество свободных мест?

Решения

- Массив с количеством свободных мест на каждом сегменте каждого отправления Int [routes*departures]
- Нельзя параллелизовать и получается медленно
- Массив AtomicInteger
- AtomicInteger [] freeSeats = new AtomicInteger[segments.size];

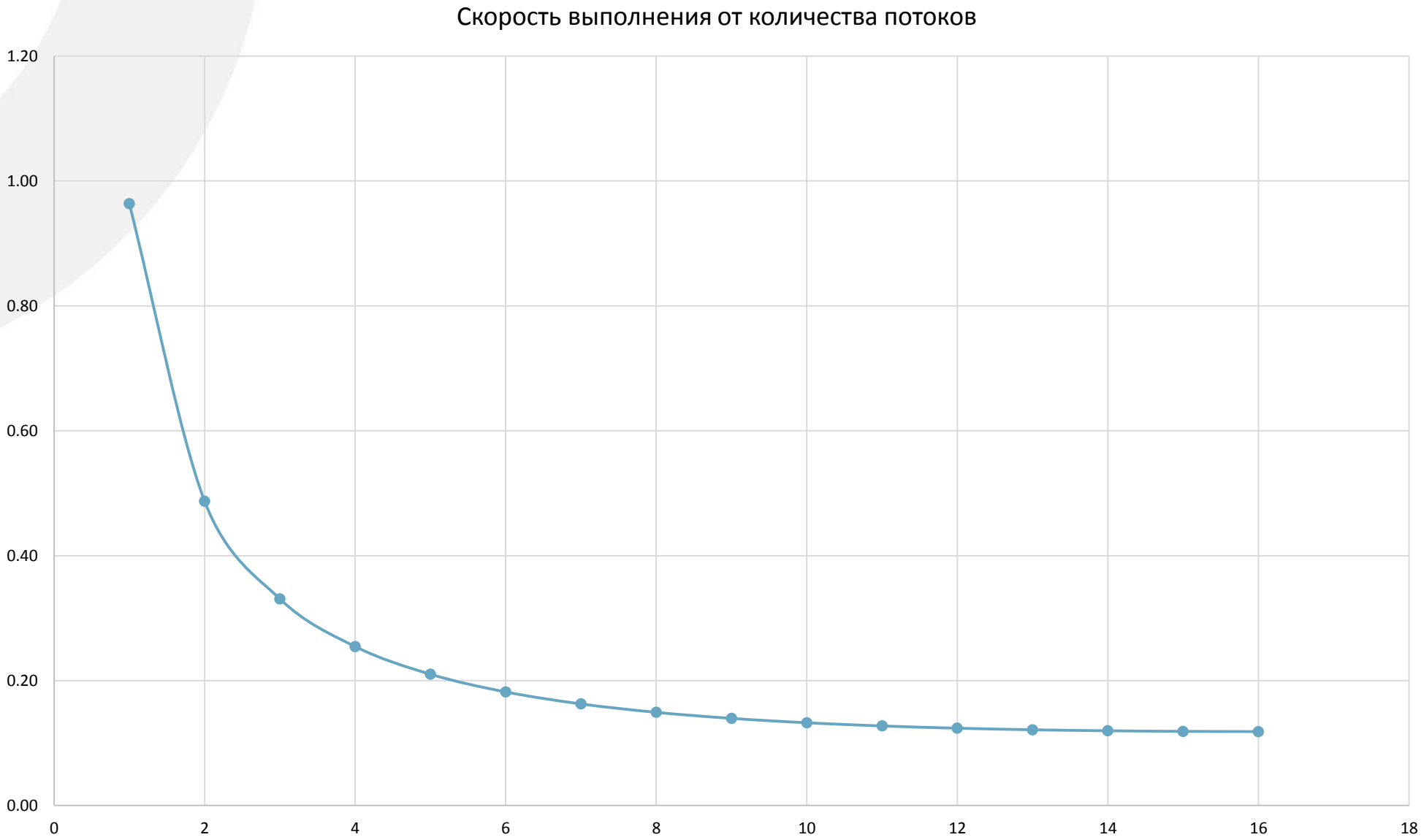
Результаты

- Медленно
- Узкое место при доступе к общим ресурсам
- Это связано с тем, что все летят через Москву
- В один поток: 19 единиц
- В четыре потока: 15 единиц

Решение № 3. Разделение общих ресурсов с их общим слиянием

Описание алгоритма параллельной работы:

- Разделить ресурсы на каждый поток
- Сохранить общее хранилище всех ресурсов
- При получении агентом маршрута менять состояние общего хранилища
- При достижении остаточной ёмкости менее 20% в общем количестве отключать маршрут у всех
- Если у одного из воркеров один из маршрутов иссяк, отправлять сигнал на перераспределение свободных мест этого маршрута


Скорость выполнения от количества потоков
1.20
1.00
0.80
0.60
0.40
0.20
0.00
0
2
4
6
8
10
12
14
16
18

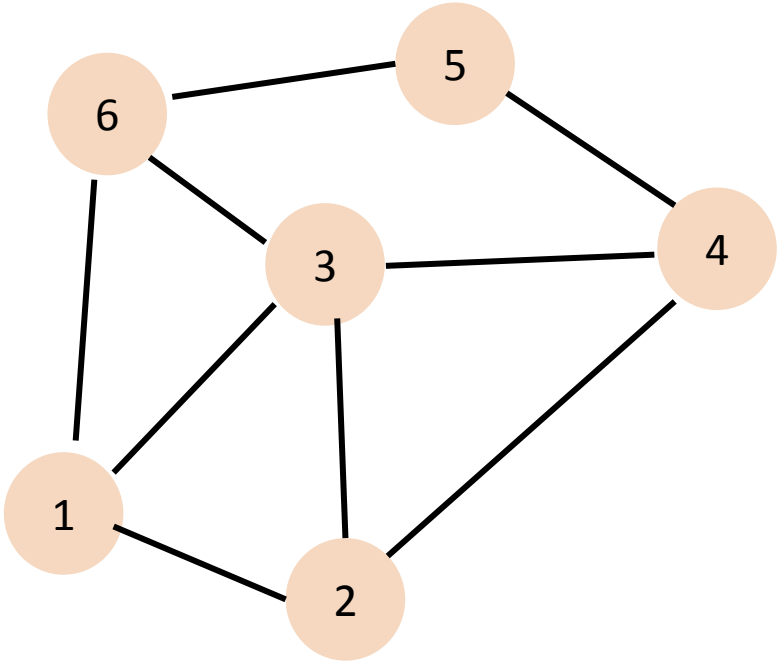
Описание ограничений: как мы с ними мирились

На стадии проекта было заложено, что фоновый спрос составляет 20%.

За счет этого удалось достигнуть практически полного отсутствия операций синхронизации.

Поиск новых рейсов

Задача поиска маршрута



Простое решение удовлетворения спроса

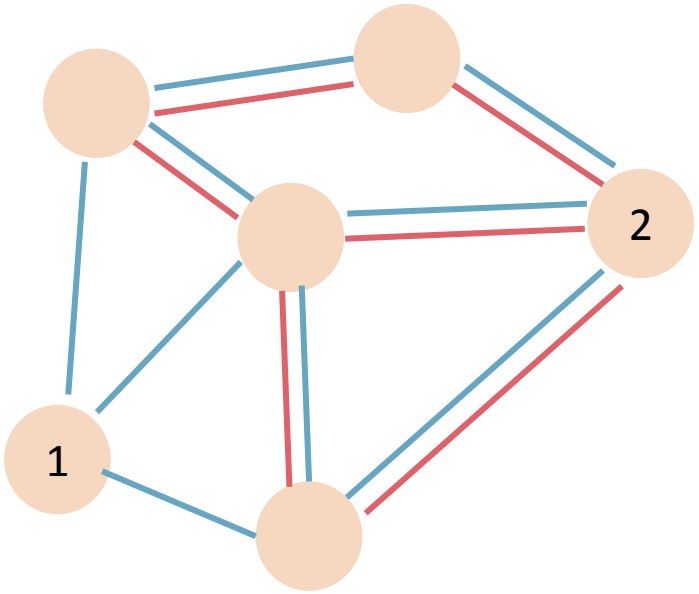
- Добавление чартеров
- Сложно реализуется из-за ограничения парковок
- Финансово не самый эффективный вариант
- Только для организованных групп

Что делать?

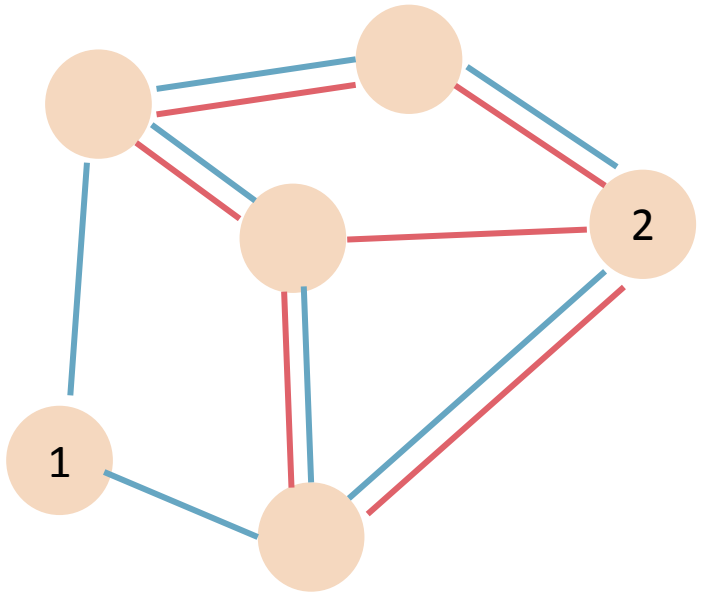
поезд

самолет

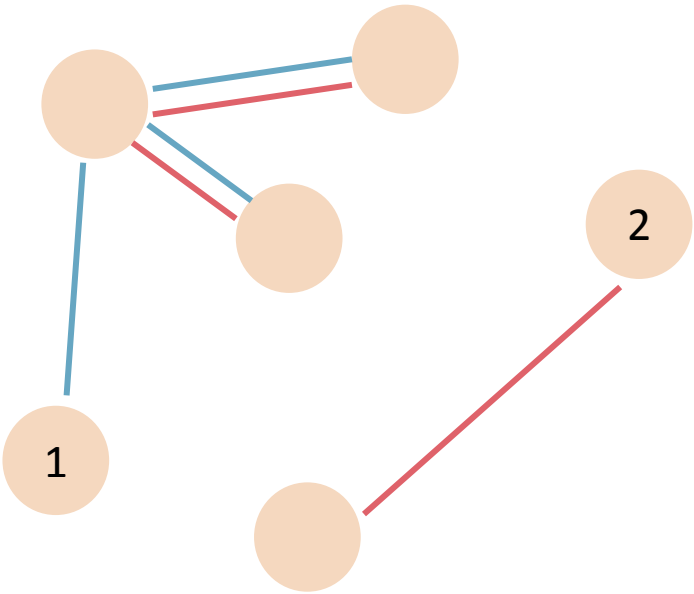
100 людей в день



1000 людей в день



5000 людей в день

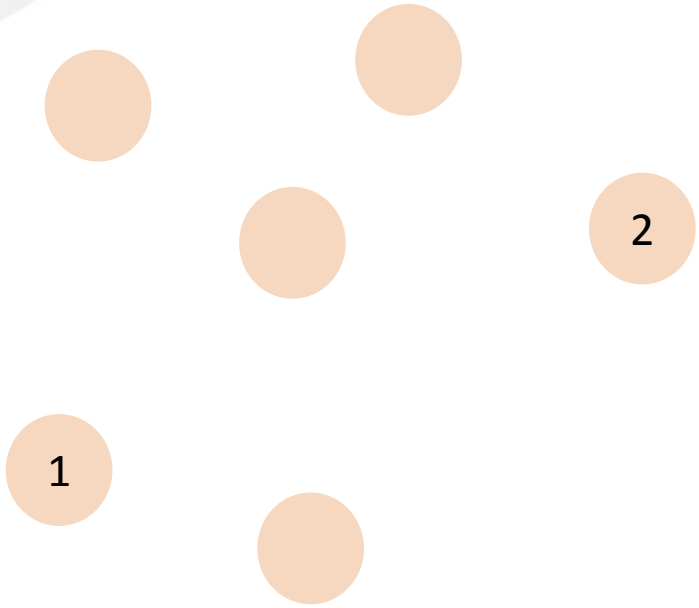


Поиск дополнительных ветвей графа

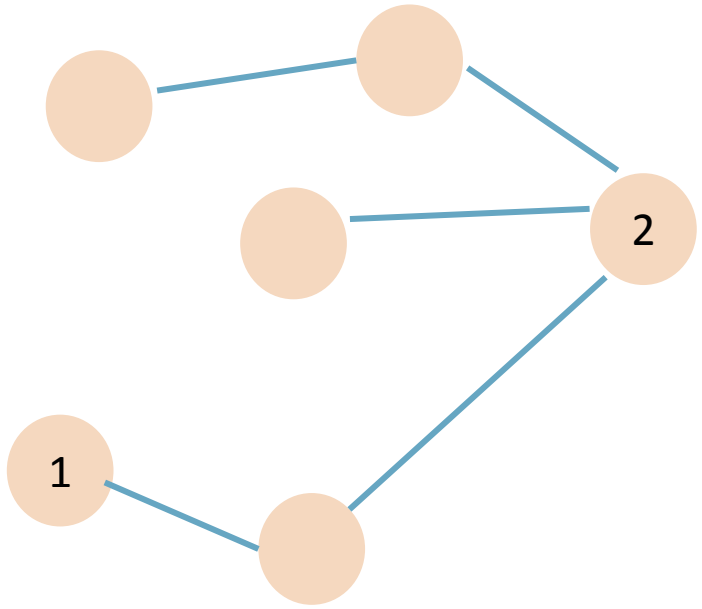
- Данные:
  - Дата игры
  - Группа болельщиков (Например Германия из Берлина)
  - Количество недоехавших болельщиков
- Полный перебор вариантов
- Эвристические алгоритмы
  - Теория перколяции
  - Генетические алгоритмы

Теория перколяции

N = 0%

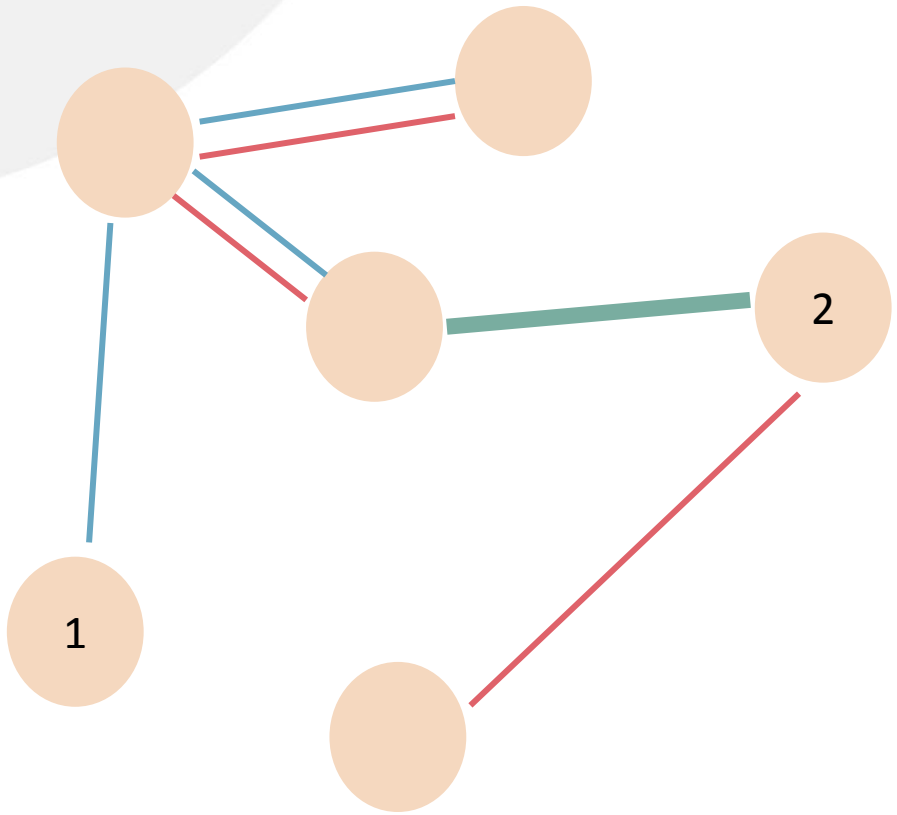


N = 50%

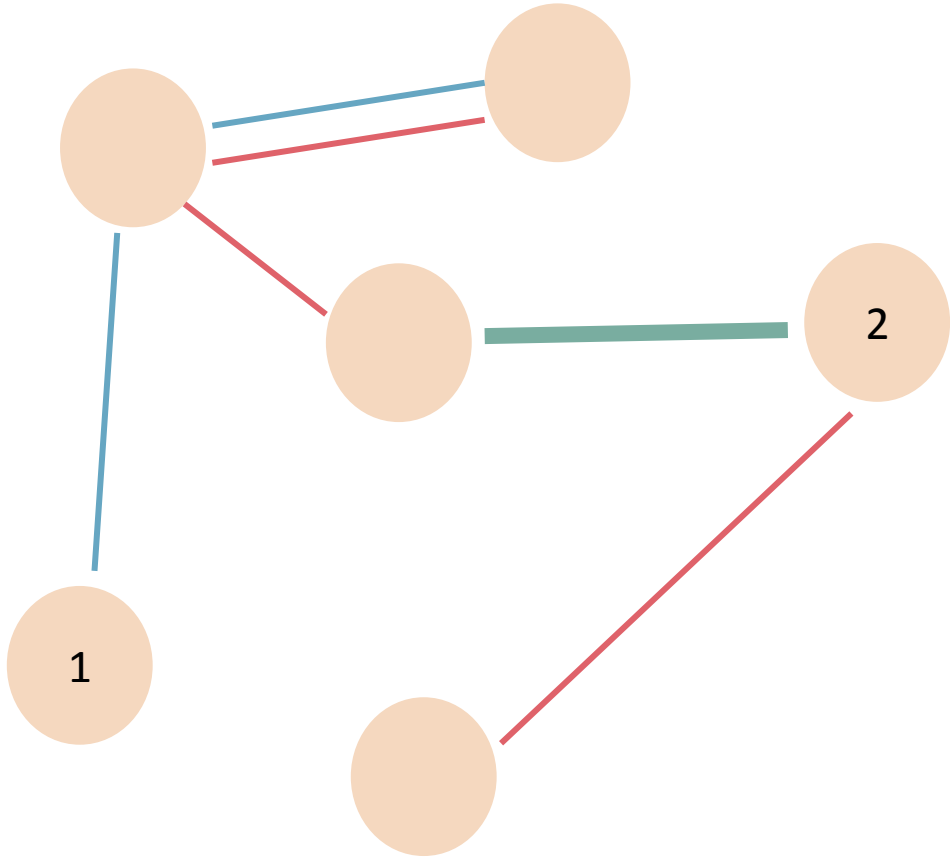


Использование генетического алгоритма

Мутация времени



Мутация станции



Использование эвристического алгоритма

- Случайный выбор алгоритма генерации новых рейсов
- Генерация новых рейсов
- Моделирование в течение 5 итераций
- Уничтожение по критерию выживаемости. Критерий выживаемости: загруженность более 80%

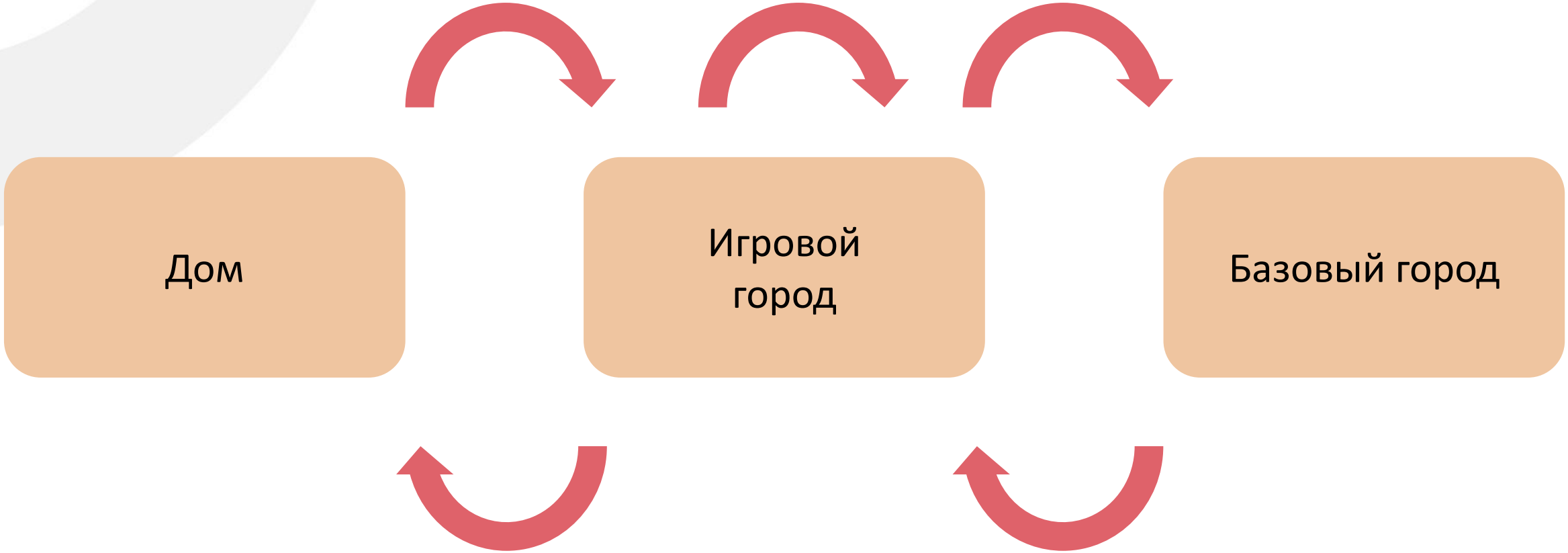
Каннибализм

- Какую проблему создают мутанты?
- Они съедают пользователей у нормальных рейсов
- Каким образом можно решить эту проблему?
- Решение: добавление штрафного времени для мутантов (4 часа)

Поведение зрителя

Описание задачи

Маковская модель

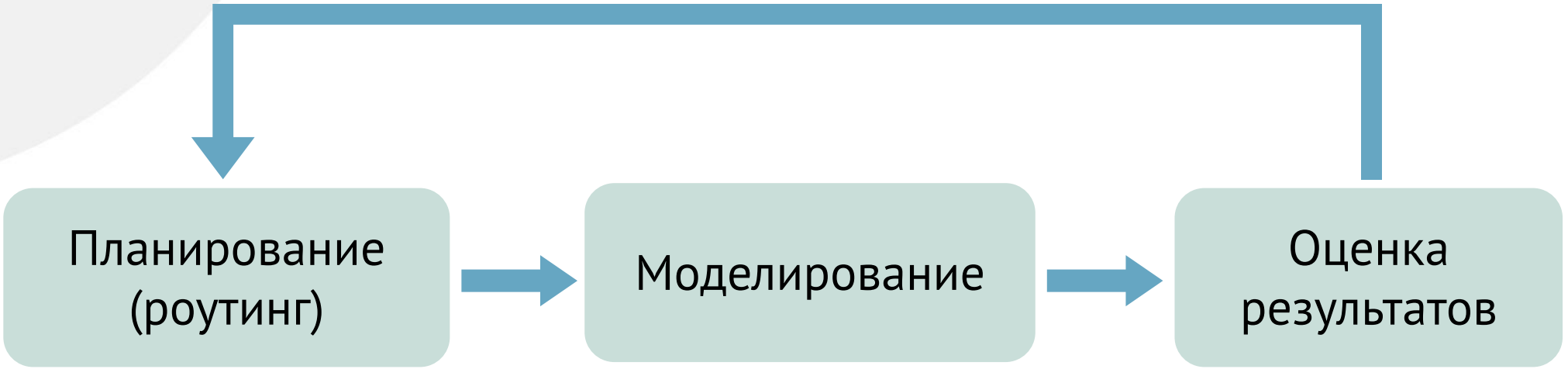


Признаки

- Количество гостиничных мест в текущем городе
- Количество гостиничных мест в следующем городе
- Количество гостиничных мест в базовом городе
- Вероятность долгой игры команды на турнире
- Откуда взять вероятность долгой игры?
- Букмекеры как не очень надежный, но все-таки источник

Уменьшение времени расчета

Последовательный процесс в MATSim



20 единиц
в 6 потоков

20 единиц
в 16 потоков

5 единиц

Эволюционный алгоритм создает много мусора

- Сборка мусора JVM занимает 5% от времени работы системы
- Вынос расчетного ядра в отдельное приложение
- По возможности отключить сборщик мусора

GC Epsilon

Управляет аллокацией памяти

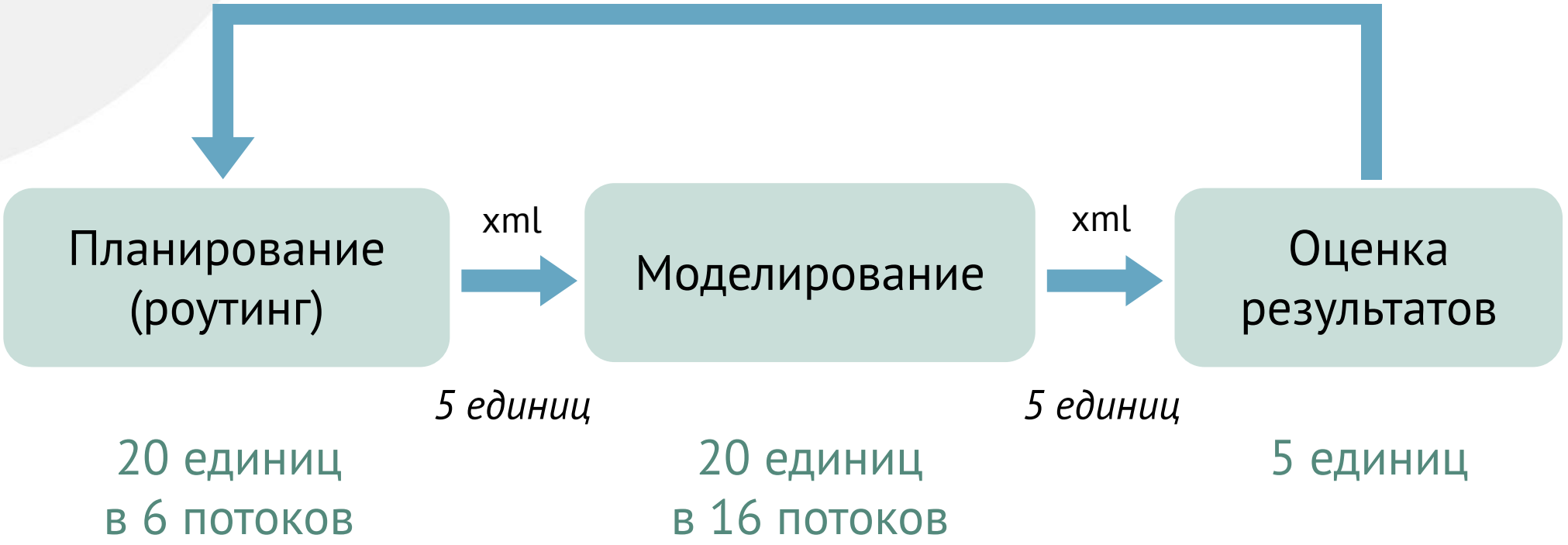
Не управляет ее освобождением

Как только память закончилась, сразу отрубается

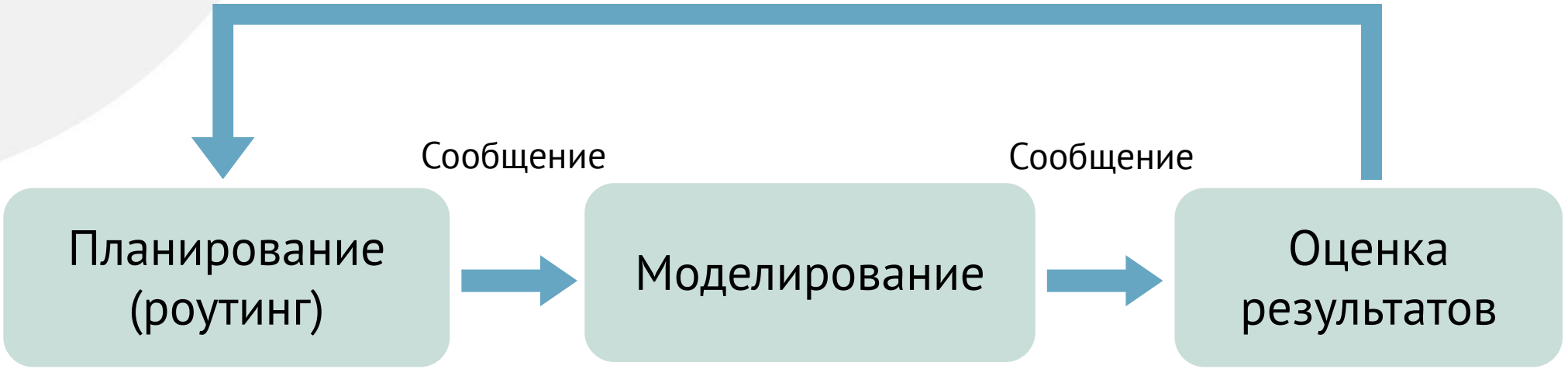
Включается через -XX:+UnlockExperimentalVMOptions -XX:+UseEpsilonGC

Мы им не пользовались, но это было бы идеальным решением

Выделение моделирования в отдельный модуль

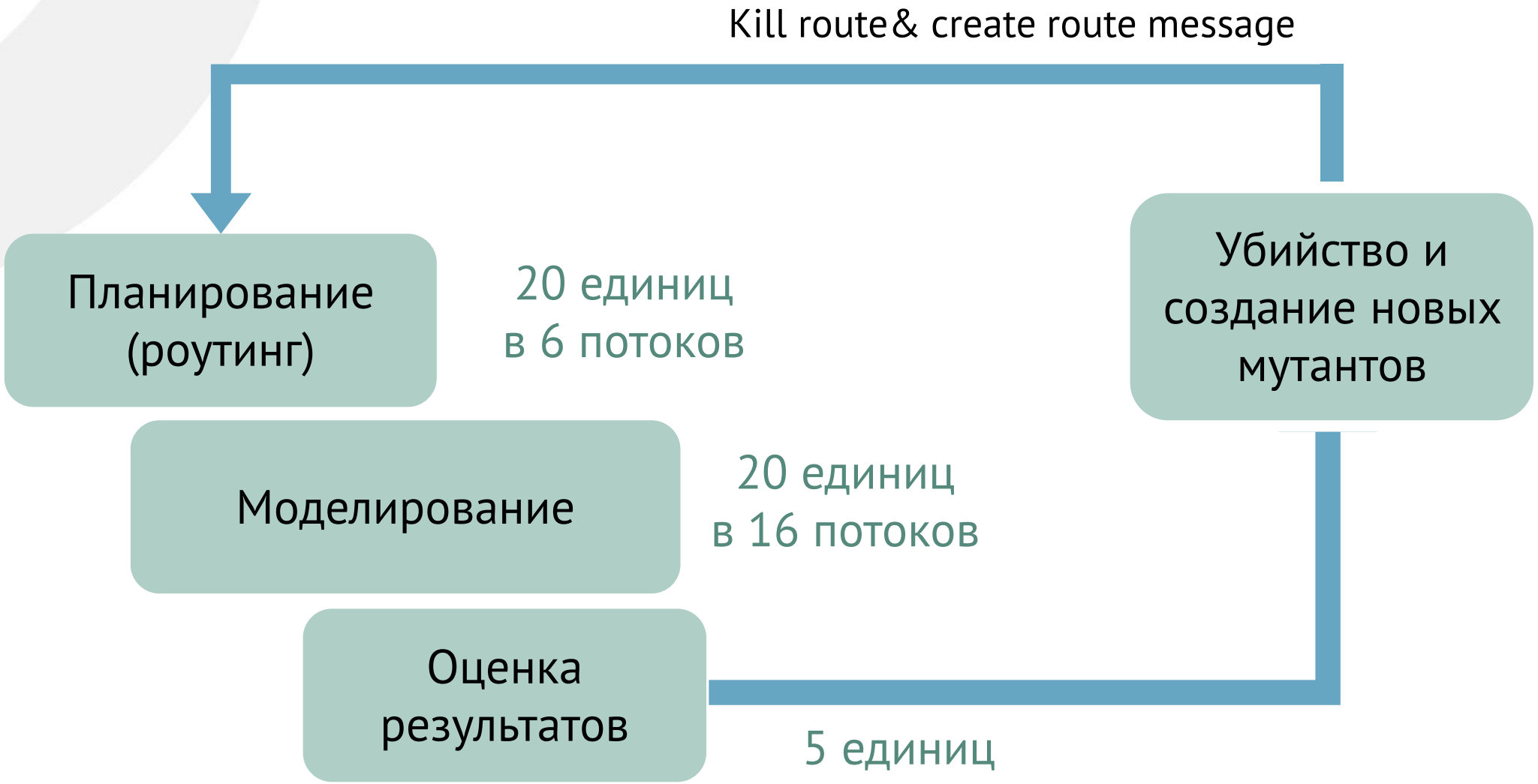


Отказ от xml


Сообщение
Сообщение
Планирование (роутинг)
Моделирование
Оценка результатов
20 единиц в 6 потоков
20 единиц в 16 потоков
5 единиц

Если мы отказались от xml, что можно улучшить еще?

Можно все и сразу



Некорректность подхода с точки зрения моделирования

Какая есть некорректность?

- Люди не планируют свои поездки в порядке путешествий
- Разрезали поездку на отдельные корреспонденции и перемешали

Проблема большого количества вариантов

Комбинаторика

126 команд участвуют в отборе, лучшие 32 разместятся по 8 группам

$6\ 10^{113}$ вариантов – на ранней стадии отбора

20736 вариантов на финальной стадии отбора

4096 вариантов после отбора

1 вариант после финальной жеребьевки

Что делать?

- Моделировать 20736 вариантов?
- Это займет 28 лет
- Что сделать?
- Уменьшить количество команд до 4-х.
- Как это сделать?

Транспортные кластеры

- Гипотеза «С точки зрения России существуют транспортные кластеры, для которых осуществление транспортной корреспонденции – гомогенно»
- Аналитическое решение. Кластеризовать по расстоянию и провозной способности на размер страны.
- Численное решение. Полученные кластеры были проверены на кратком сценарии.

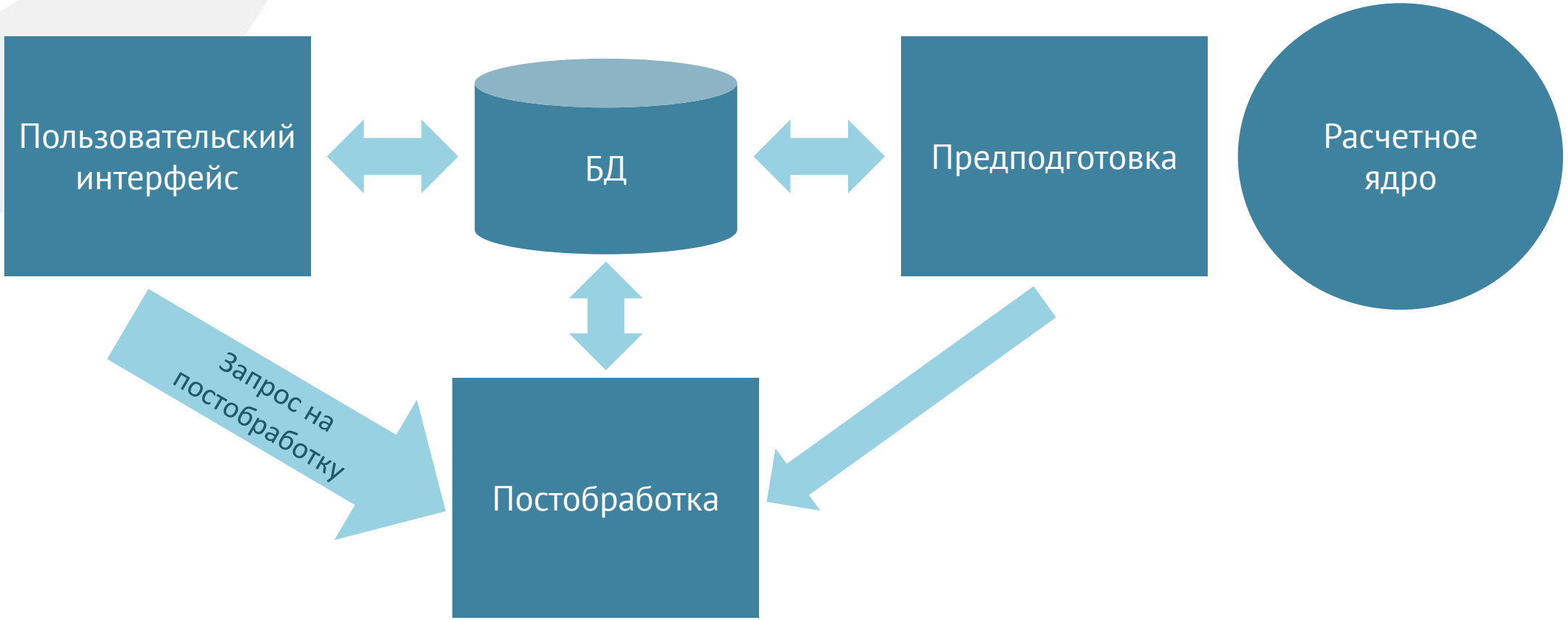
Визуализация

Обработка данных по моделированию

- Последовательность событий
- Результаты хранятся в xml файле размером 9GB

```xml
<?xml version="1.0" encoding="utf-8"?>
<events version="1.0">
    <event time="21600.0" type="actend" person="1" link="5" actType="h" />
    <event time="21600.0" type="departure" person="1" link="5" legMode="car" />
    <event time="21600.0" type="PersonEntersVehicle" person="1" vehicle="1" />
    <event time="21600.0" type="vehicle enters traffic" person="1" link="5" vehicle="1" networkMode="car" relativePosition="1.0" />
    <event time="21601.0" type="left link" vehicle="1" link="5" />
    <event time="21601.0" type="entered link" vehicle="1" link="1" />
    <event time="21601.0" type="personMoney" amount="-10000.0" person="1" />
    <event time="21652.0" type="left link" vehicle="1" link="1" />
    <event time="21652.0" type="entered link" vehicle="1" link="2" />
    <event time="21703.0" type="left link" vehicle="1" link="2" />
    <event time="21703.0" type="entered link" vehicle="1" link="6" />
    <event time="21713.0" type="vehicle leaves traffic" person="1" link="6" vehicle="1" networkMode="car" relativePosition="1.0" />
    <event time="21713.0" type="PersonLeavesVehicle" person="1" vehicle="1" />
    <event time="21713.0" type="arrival" person="1" link="6" legMode="car" />
    <event time="21713.0" type="actstart" person="1" link="6" actType="w" />
</events>
```

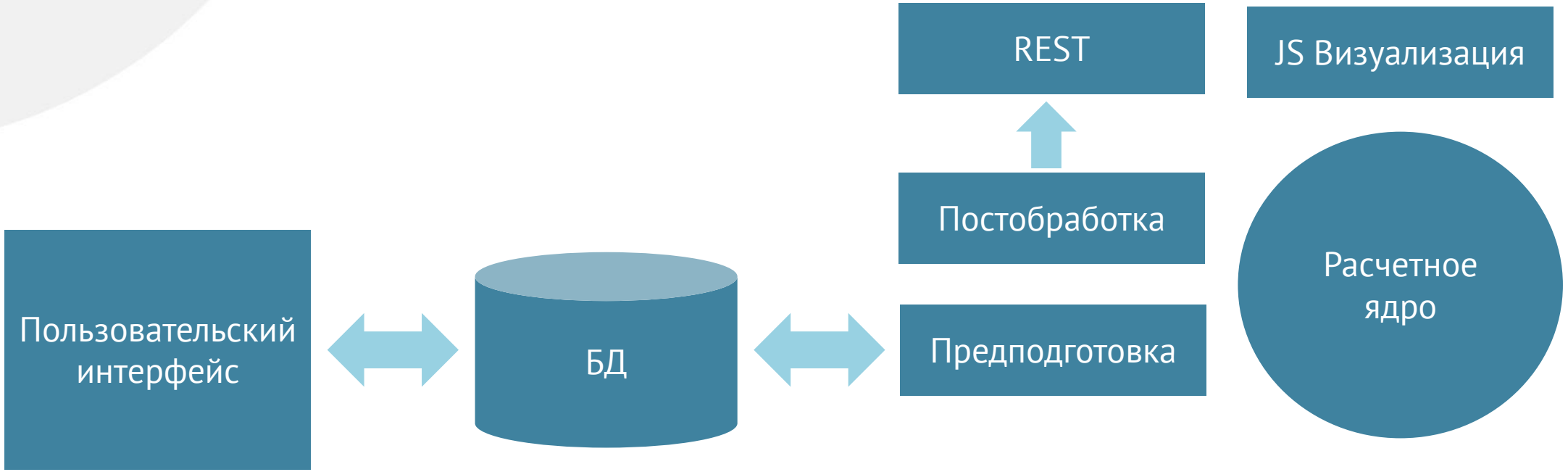
Быстрое решение



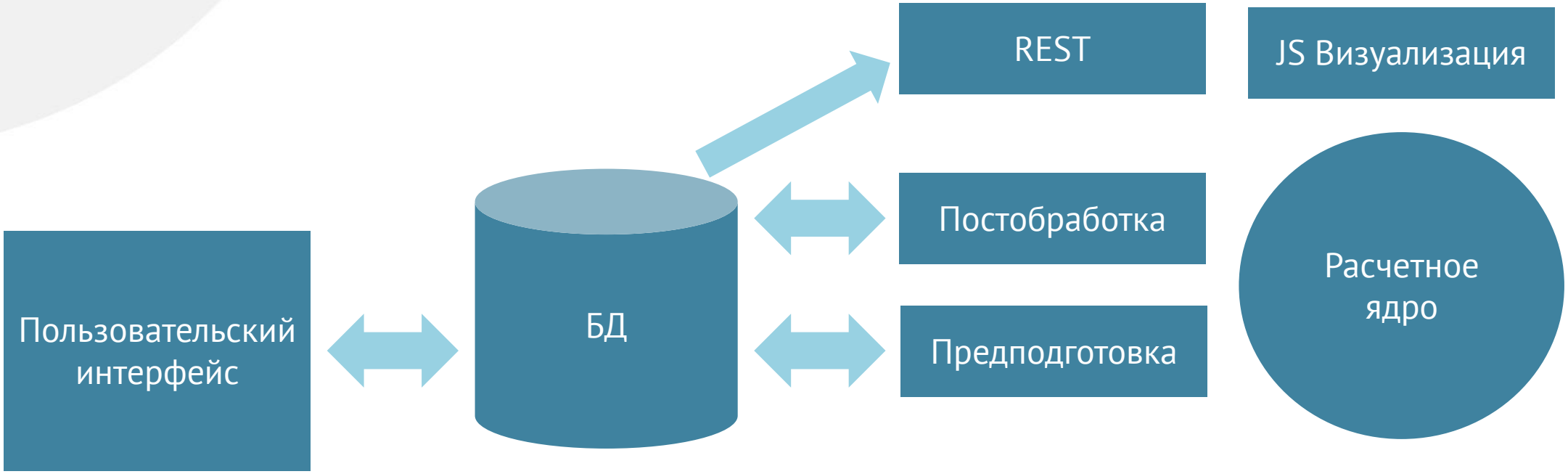
Узкое место

- Вялое ТЗ
- Очень много требований к цвету элементов и толщине линий
- Что можно сделать?
- Ругаться с заказчиком, отказываясь решать непринципиальные вопросы
- Пытаться решить вопросы с GWT и купить JRebel
- Изменить архитектуру, к тому же у вас есть команда JS

Что можно сделать в этой ситуации?



Предподготовка данных персистинг в реляционную БД

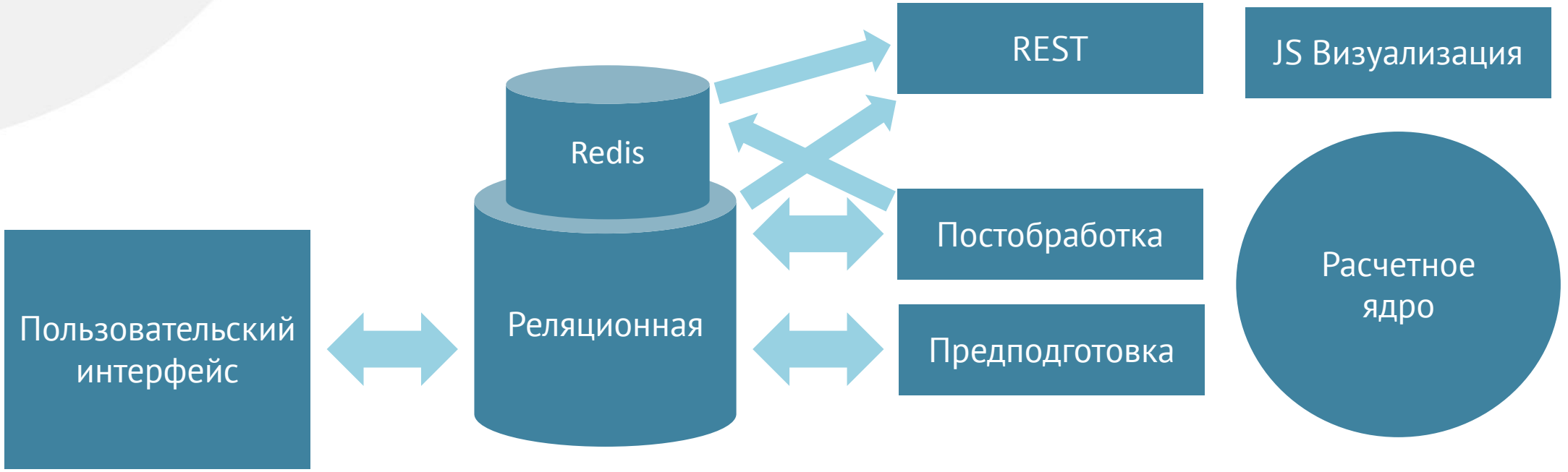


Какие проблемы может вызвать такой подход?

- Долгое сохранение (например, информация по всем рейсам)
- 240 секунд на сохранение данных по всем отправлениям
- Rest долго получает данные и обрабатывает их

Решение по умолчанию – включить кэширование

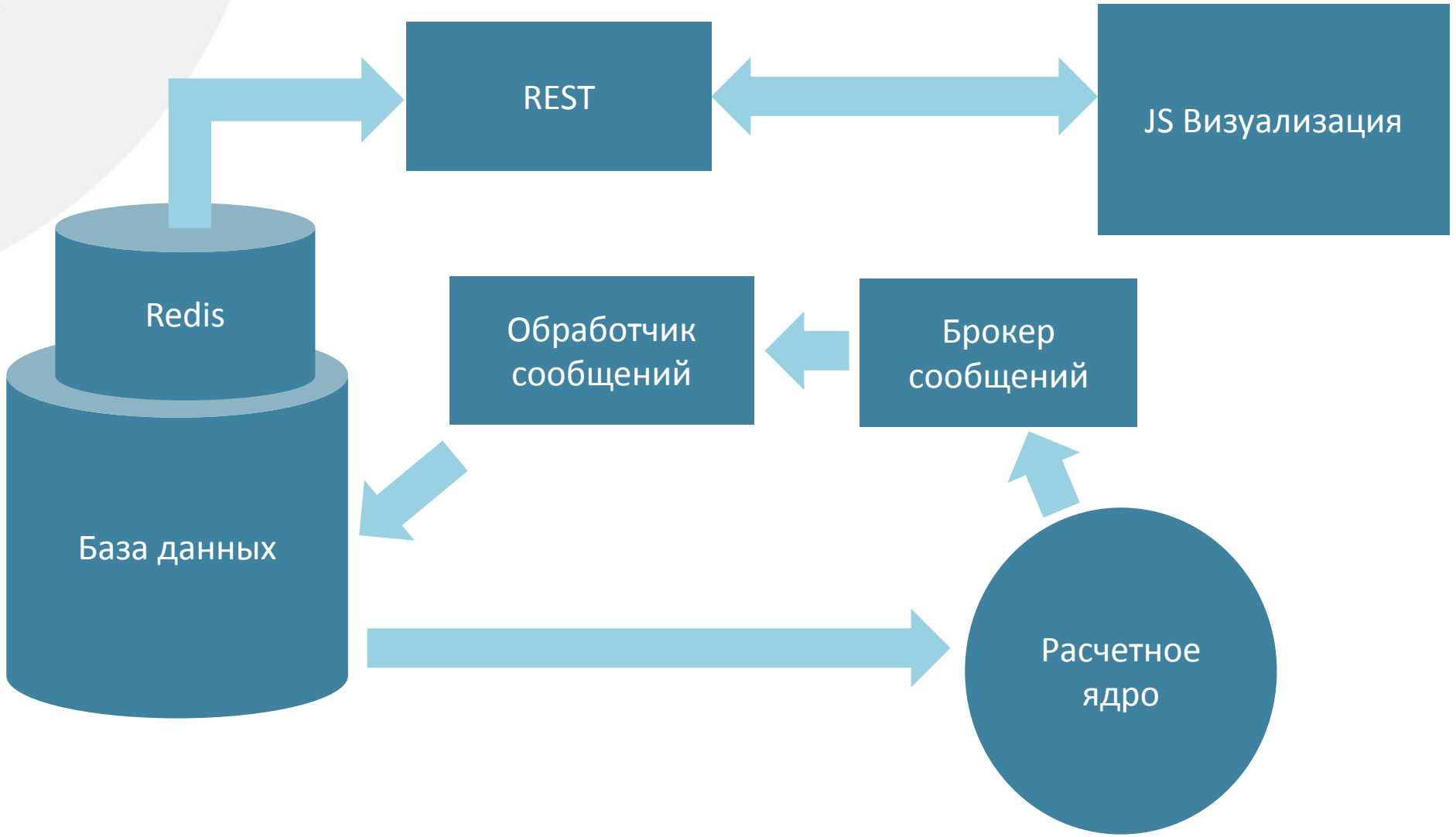
Комбинированный подход Redis + Postgresql



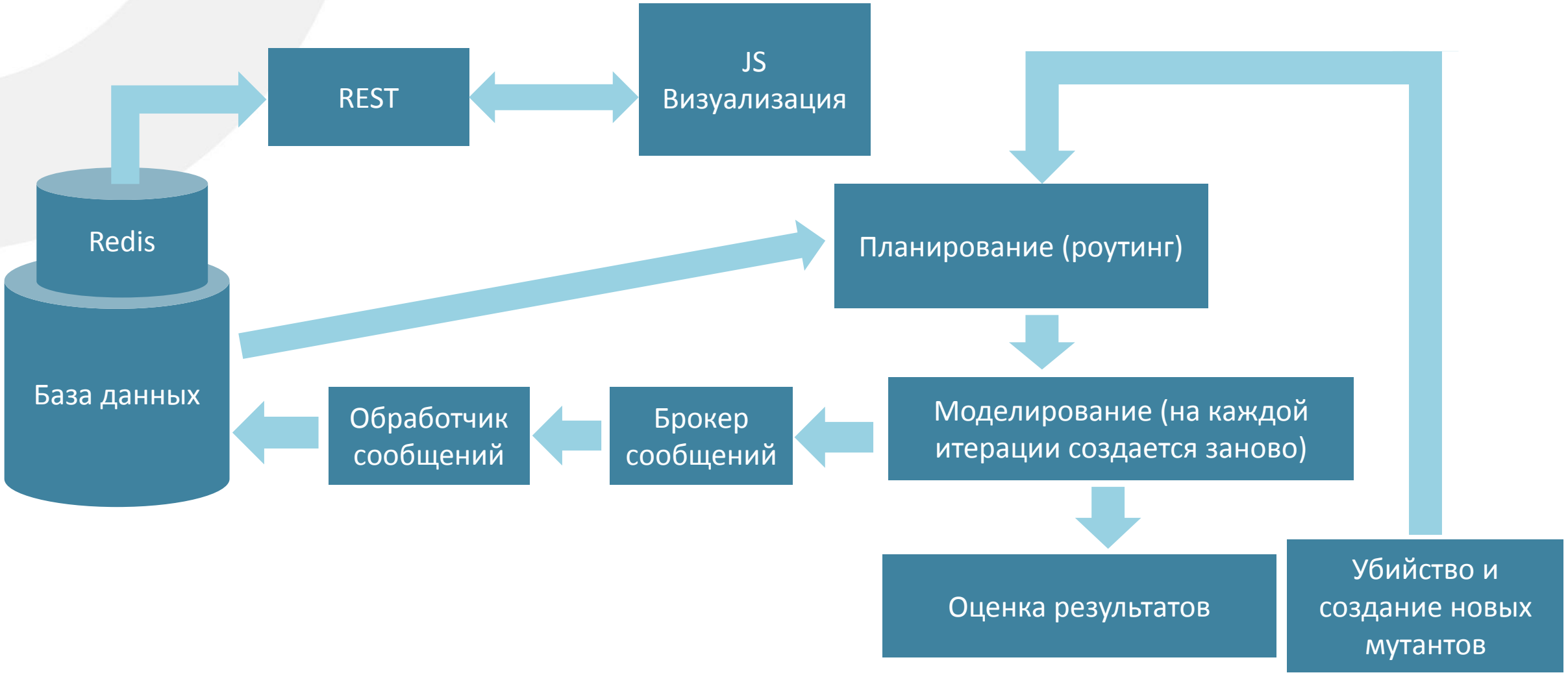
Долго ждать результатов

- Заказчик хочет видеть результаты в реальном времени, чтобы быстрее реагировать на решения
- Постобработка занимает 20 минут, новые данные невозможно переобрабатывать
- Отказаться от постобработки в один присест

Использование сообщений из моделирования



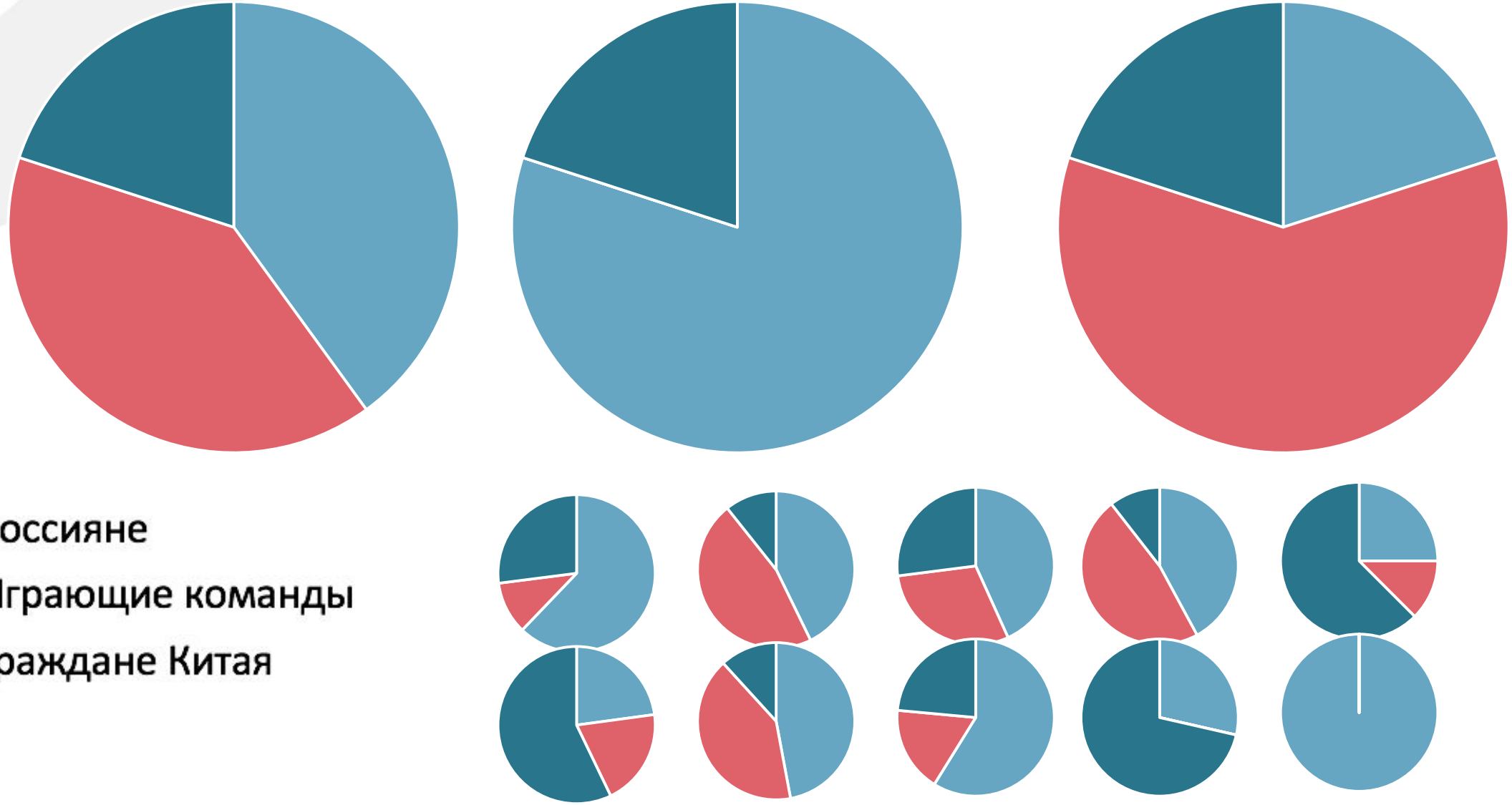
Общая схема


REST
JS
Визуализация
Redis
База данных
Планирование (роутинг)
Моделирование (на каждой итерации создается заново)
Брокер сообщений
Обработчик сообщений
Оценка результатов
Убийство и создание новых мутантов

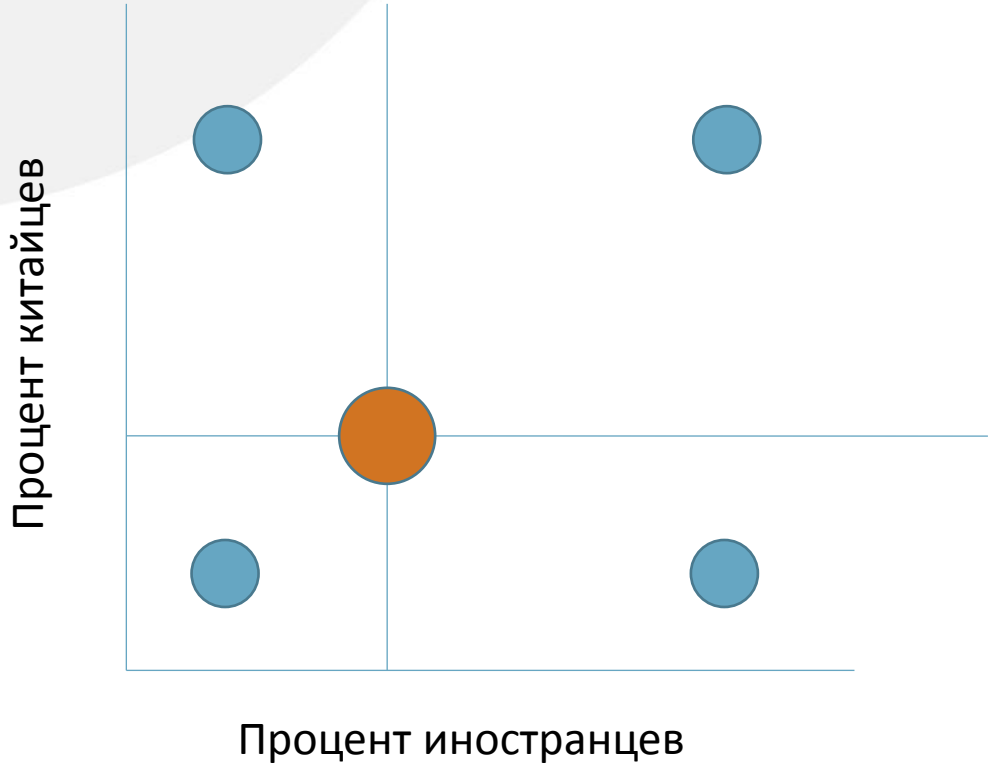
Всегда есть место для глупости

- JS UI работает медленно на медленных каналах
- Что мы забыли?
- Gzip сжатие

Рассмотреть большое количество вариантов



Метамодель



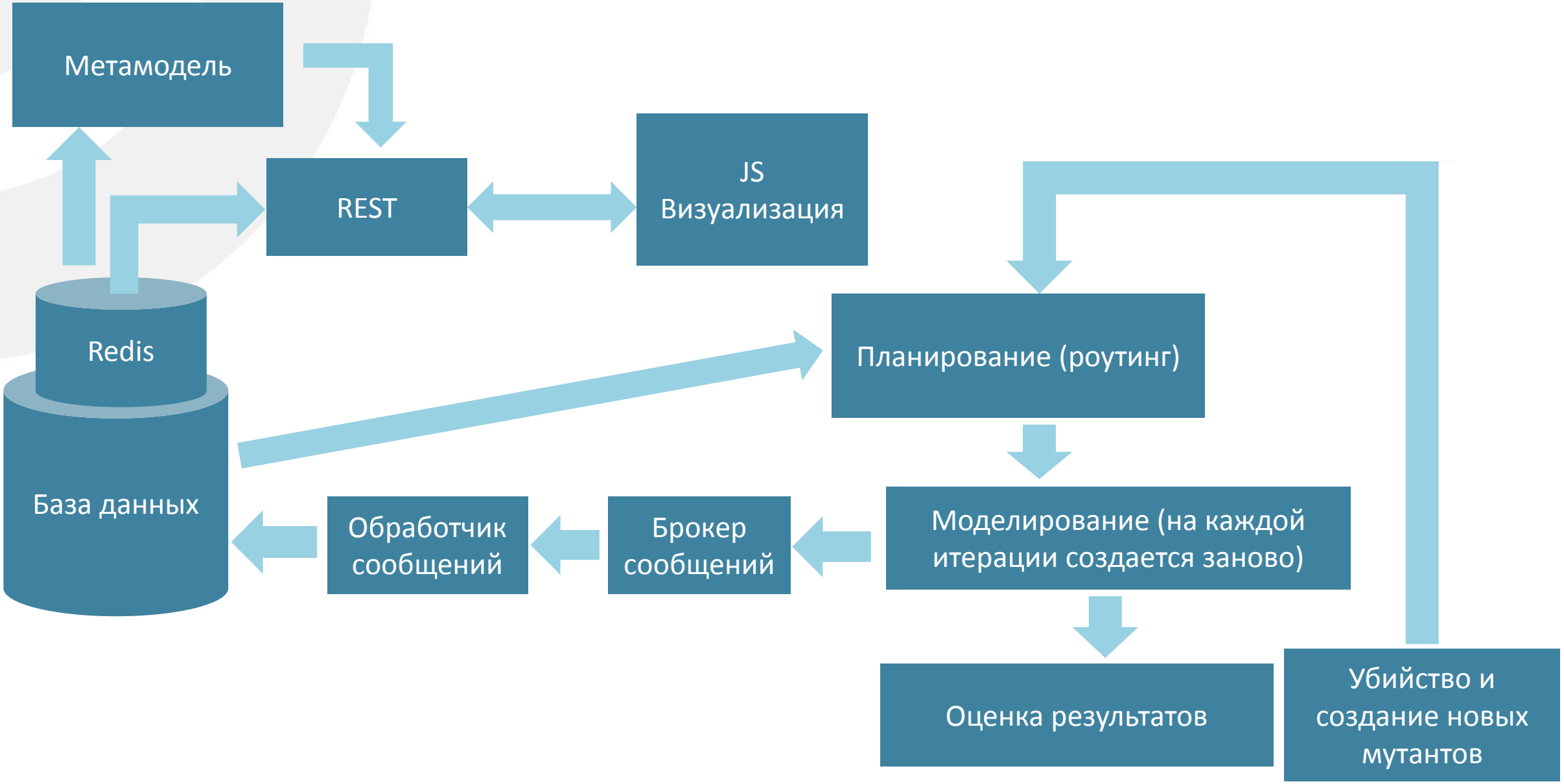
Численные значения:

$$F(x,y) = f(0,0)\,(1-x)(1-y) +$$

$$f(1,0)\,x(1-y) + f(0,1)\,(1-x)y + f(1,1)xy.$$

Для множеств:

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$


Метамодель
REST
JS
Визуализация
Redis
База данных
Планирование (роутинг)
Моделирование (на каждой итерации создается заново)
Брокер сообщений
Обработчик сообщений
Оценка результатов
Убийство и создание новых мутантов

Сдача заказчику

Головоломка

- Аэропорт одного небольшого города
- В нормальной ситуации сюда прилетает 1 самолет в день из Москвы
- В городе практически нет гостиниц, 16 000 зрителей прилетают в день игры
- Из них 7 000 – иностранцы
- Пропускная способность аэропорта на международных рейсах в сутках перед игрой – 6 000 человек, на локальных рейсах – 10 000 человек
- Система показывает, что международный терминал не справится
- Заказчик говорит, что такие результаты он не примет
- Что делать?

Первая проверка. Кубок конфедераций

Что все-таки оказалось не так

- 90% Сходимость по нагрузке
- Некорректное поведение болельщиков, меньшая центрованность в Москве
- В качестве основы взяли количество зрителей. В результате – дисбаланс болельщиков, так как стадионы вмещают разное количество зрителей
- Немцев прилетало меньше чем улетало

Чемпионат мира – 2018

- Поправили поведенческую модель
- Лучший Чемпионат мира в истории
- Пиковый прогноз не состоялся из-за геополитики
- В Саранске всё хорошо

Что дальше?

- Осталось много опыта
- Остались алгоритмы

- ЧЕ – 2024 в Германии?!

Спасибо за внимание.
Вопросы?

Ярослав Смирнов
j.s@optimal-drive.ru
Facebook: Jaroslav.smirnov.9