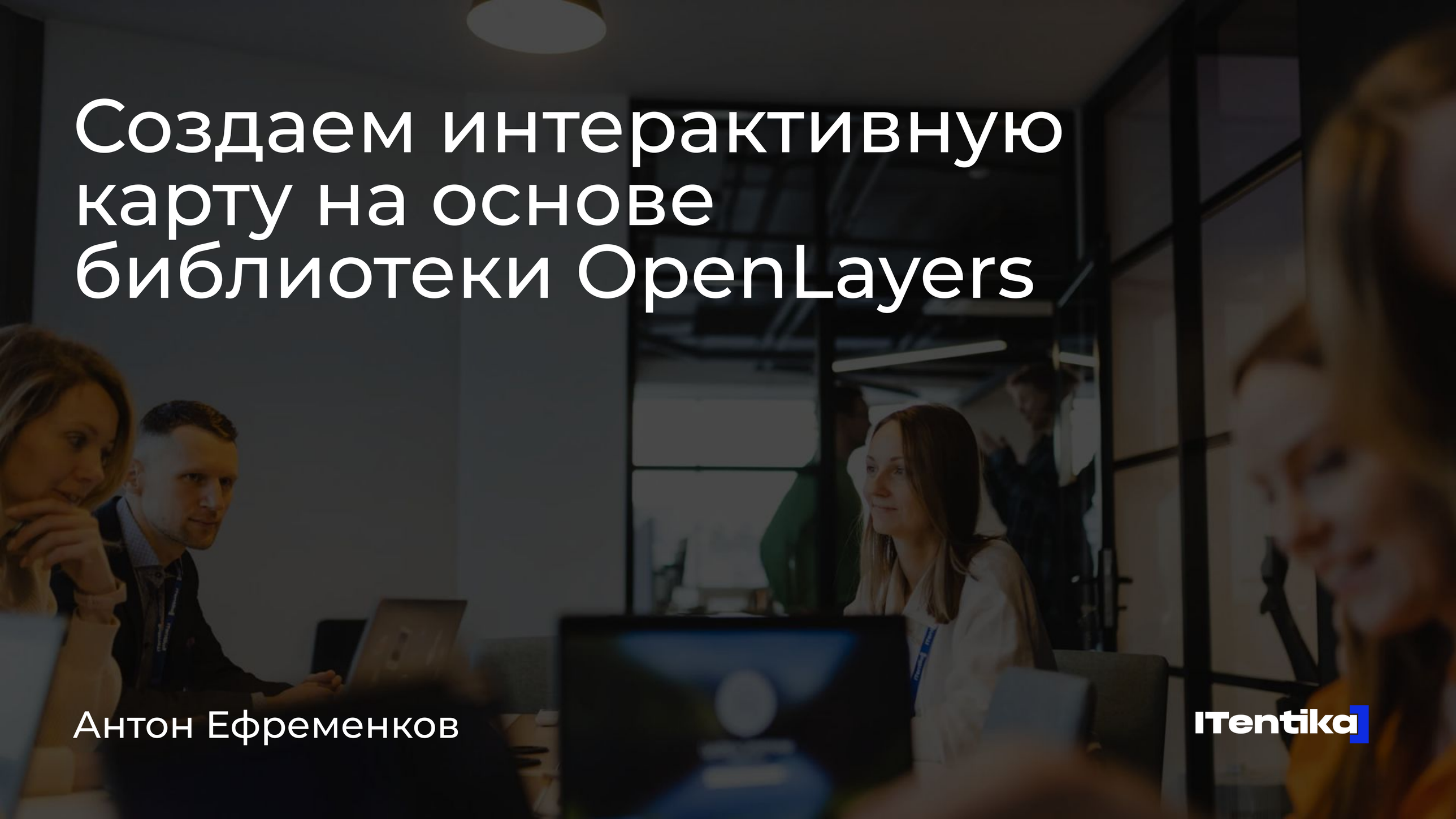
Создаем интерактивную карту на основе библиотеки OpenLayers

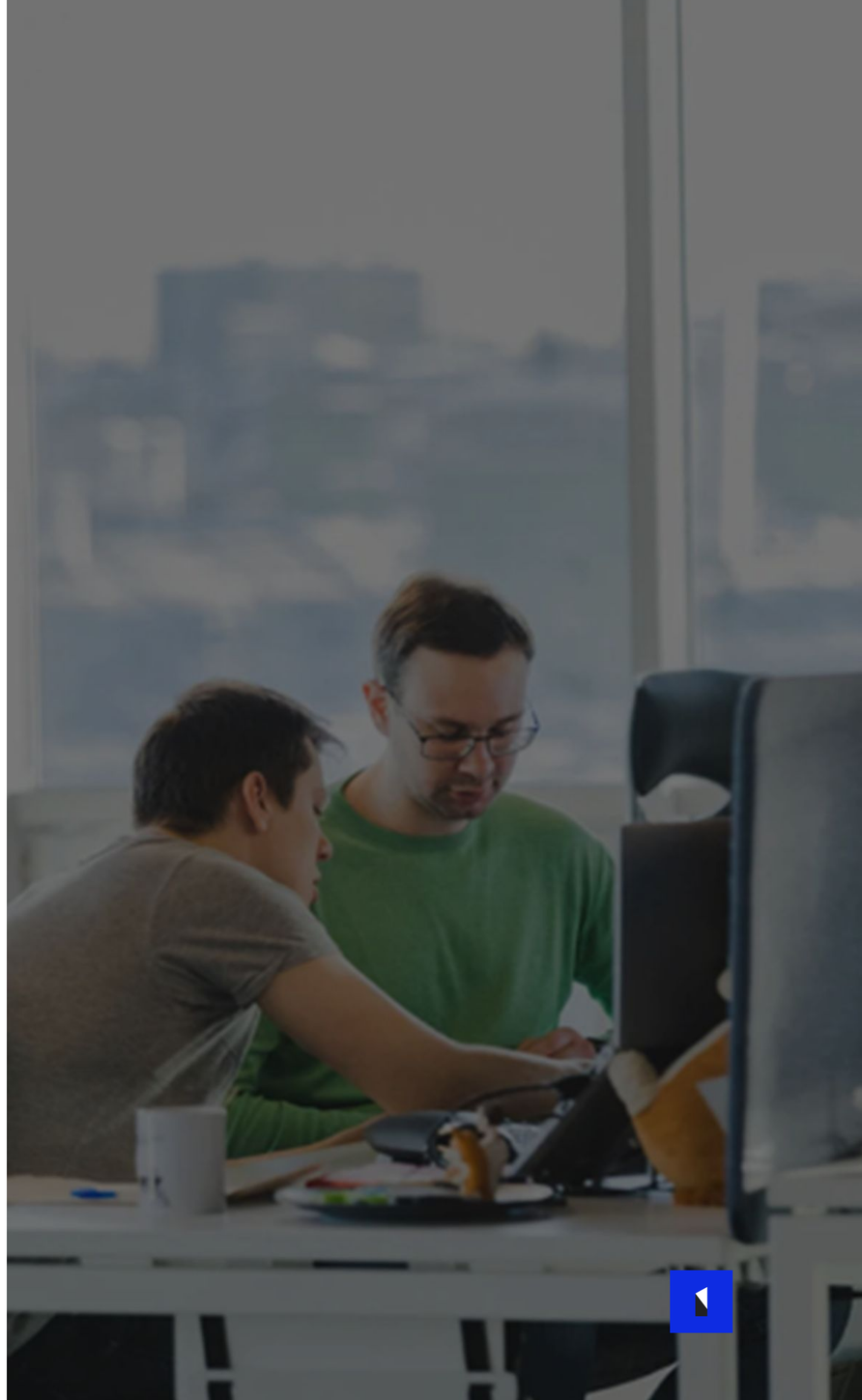


Антон Ефременков

ITentika

Наш кейс

Необходимо разработать систему для управления транспортной сетью города (автобусы, троллейбусы, трамваи, более экзотический транспорт).
Энтерпрайз-проект, команда более 30 человек.

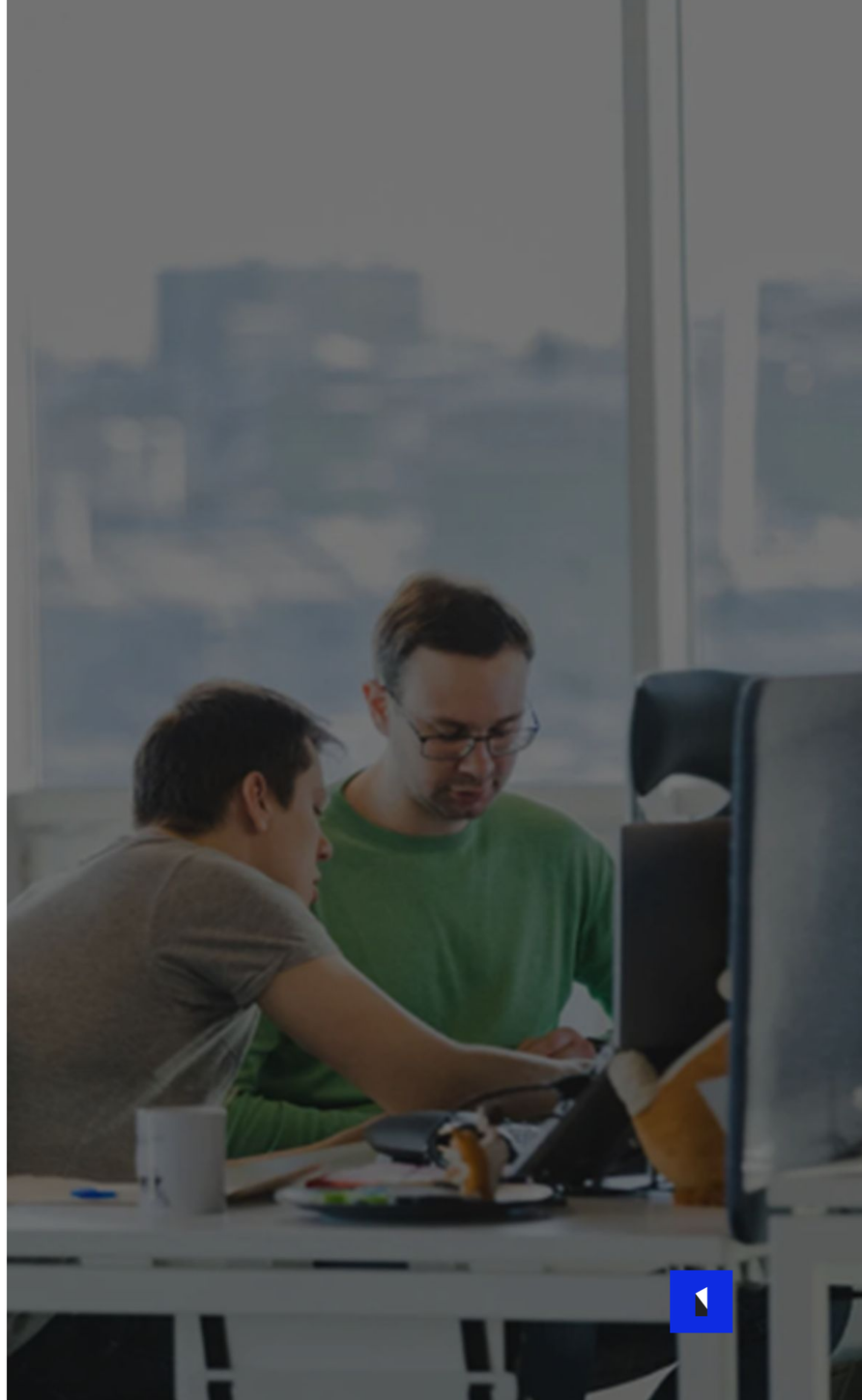


Наш кейс

Необходимо разработать систему для управления транспортной сетью города (автобусы, троллейбусы, трамваи, более экзотический транспорт).
Энтерпрайз-проект, команда более 30 человек.

Основные задачи:

- отобразить карту города

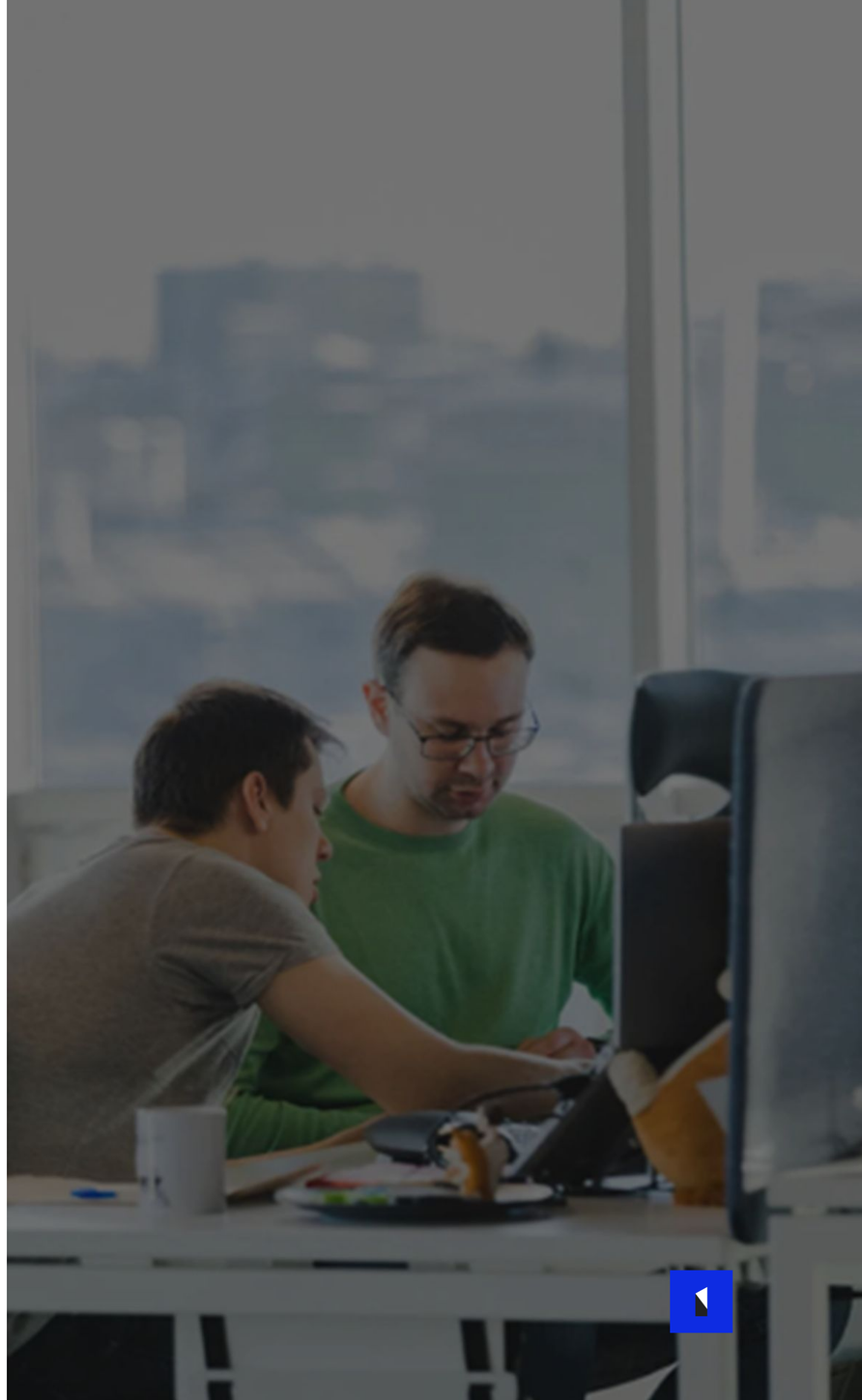


Наш кейс

Необходимо разработать систему для управления транспортной сетью города (автобусы, троллейбусы, трамваи, более экзотический транспорт).
Энтерпрайз-проект, команда более 30 человек.

Основные задачи:

- отобразить карту города
- добавлять остановки

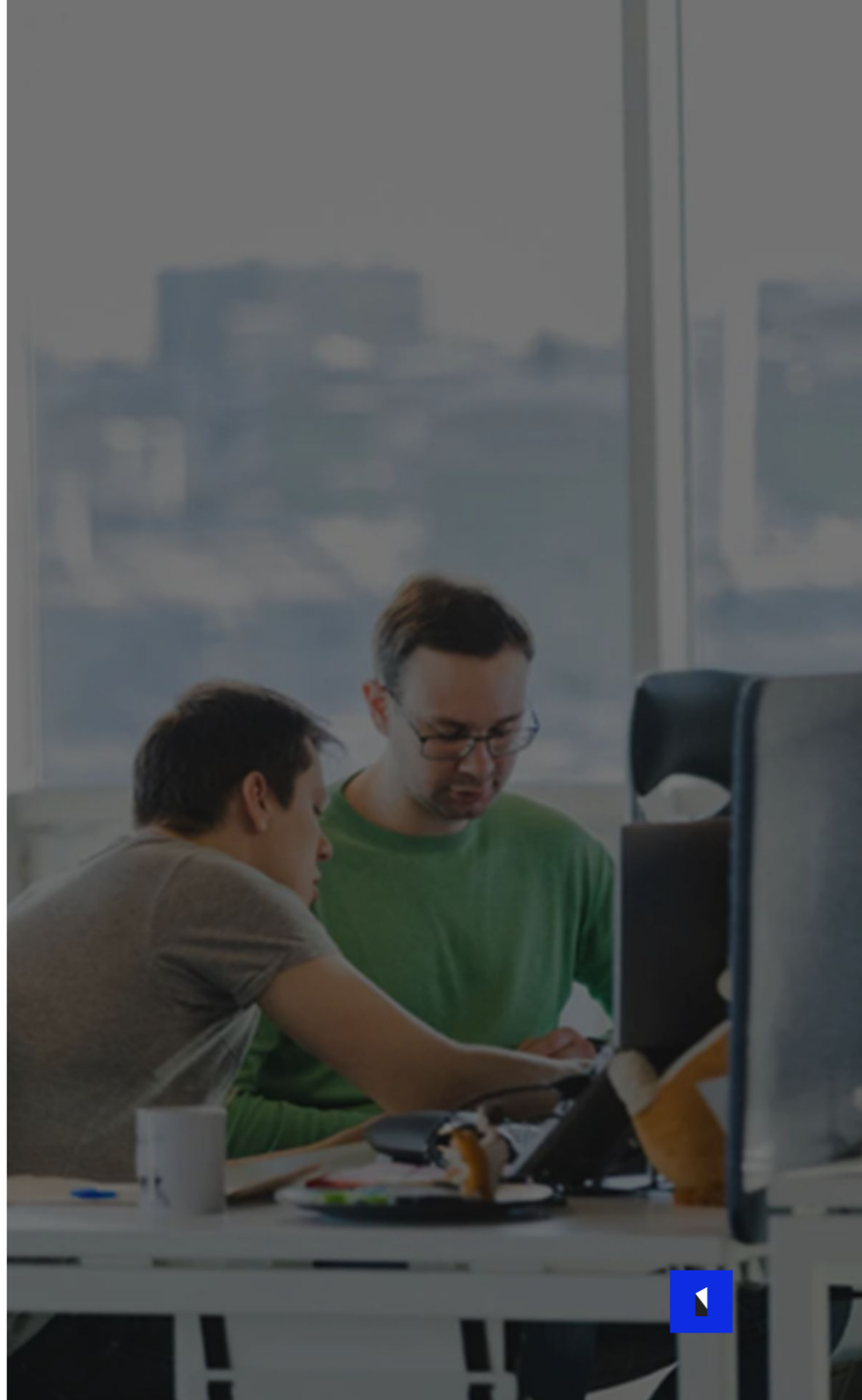


Наш кейс

Необходимо разработать систему для управления транспортной сетью города (автобусы, троллейбусы, трамваи, более экзотический транспорт).
Энтерпрайз-проект, команда более 30 человек.

Основные задачи:

- отобразить карту города
- добавлять остановки
- отображать маркеры ТС

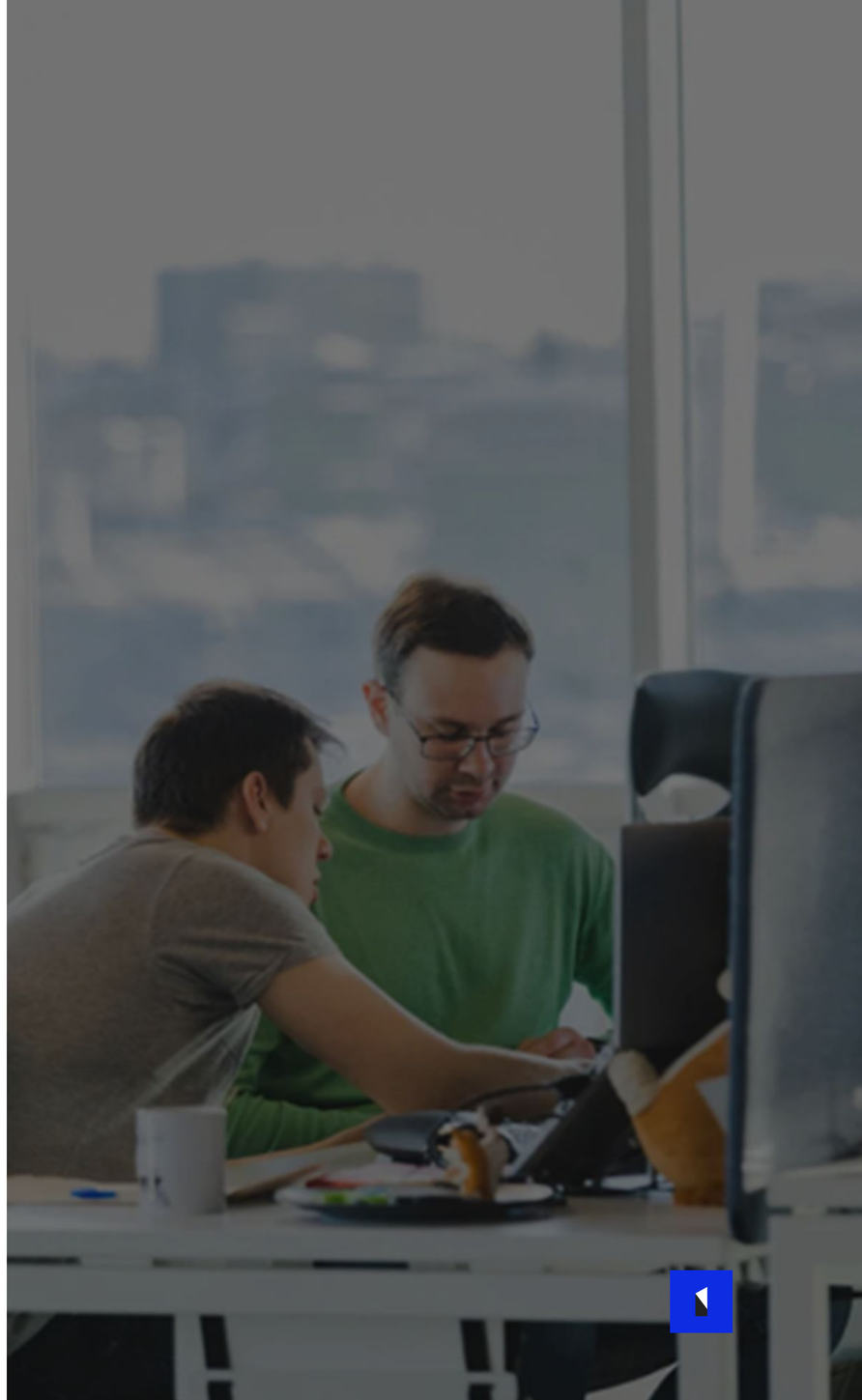


Наш кейс

Необходимо разработать систему для управления транспортной сетью города (автобусы, троллейбусы, трамваи, более экзотический транспорт).
Энтерпрайз-проект, команда более 30 человек.

Основные задачи:

- отобразить карту города
- добавлять остановки
- отображать маркеры ТС
- прокладывать маршруты

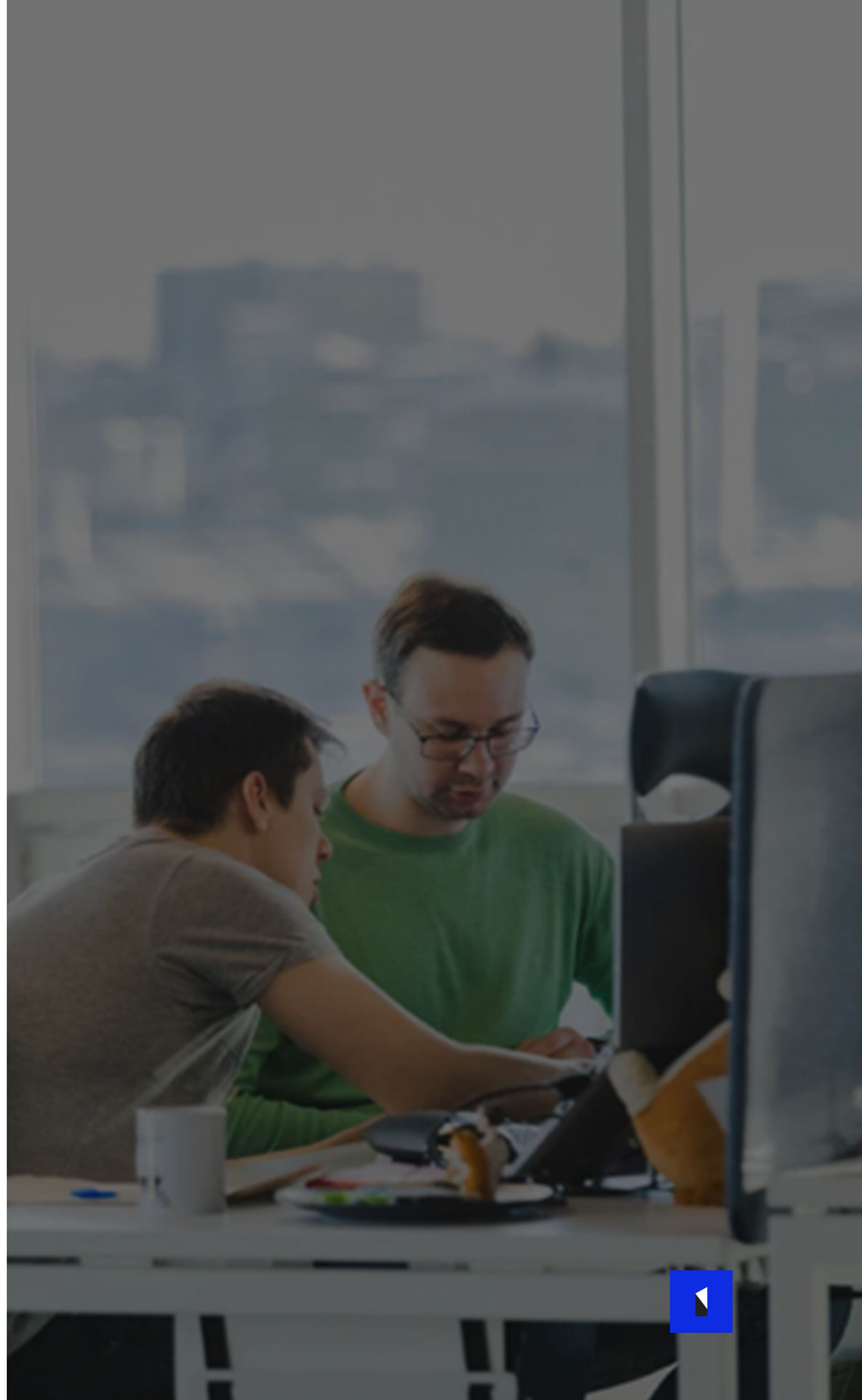


Наш кейс

Необходимо разработать систему для управления транспортной сетью города (автобусы, троллейбусы, трамваи, более экзотический транспорт).
Энтерпрайз-проект, команда более 30 человек.

Основные задачи:

- отобразить карту города
- добавлять остановки
- отображать маркеры ТС
- прокладывать маршруты
- рисовать геометрические фигуры

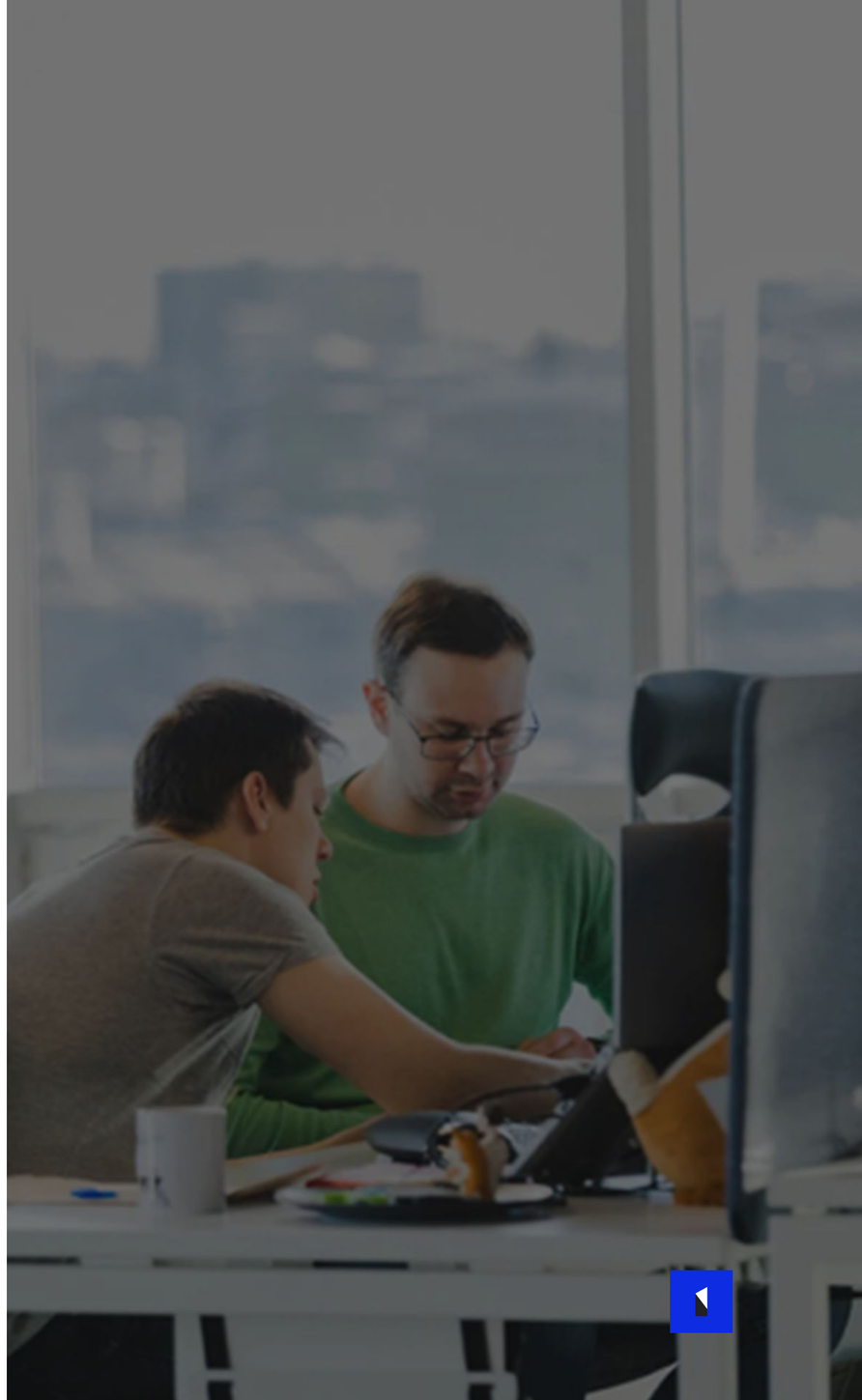


Наш кейс

Необходимо разработать систему для управления транспортной сетью города (автобусы, троллейбусы, трамваи, более экзотический транспорт).
Энтерпрайз-проект, команда более 30 человек.

Основные задачи:

- отобразить карту города
- добавлять остановки
- отображать маркеры ТС
- прокладывать маршруты
- рисовать геометрические фигуры
- обрабатывать клики по разным объектам



Отлично, у нас ещё один интересный проект!

Отлично, у нас ещё один интересный проект!

Но...

Отлично, у нас ещё один интересный проект! Но..

Есть экспертиза с Leaflet

Leaflet — одна из наиболее популярных JS-библиотек с открытым исходным кодом, предназначена для отображения интерактивных карт. Leaflet разработан с учетом простоты, производительности и удобства использования. Он эффективно работает на всех основных десктоп- и мобильных платформах.

Отлично, у нас ещё один интересный проект! Но..

Есть экспертиза с Leaflet

Leaflet — одна из наиболее популярных JS-библиотек с открытым исходным кодом, предназначена для отображения интерактивных карт. Leaflet разработан с учетом простоты, производительности и удобства использования. Он эффективно работает на всех основных десктоп- и мобильных платформах.

Заказчик - гос. компания

К сожалению, эти детали разглашать не можем =(

Отлично, у нас ещё один интересный проект! Но..

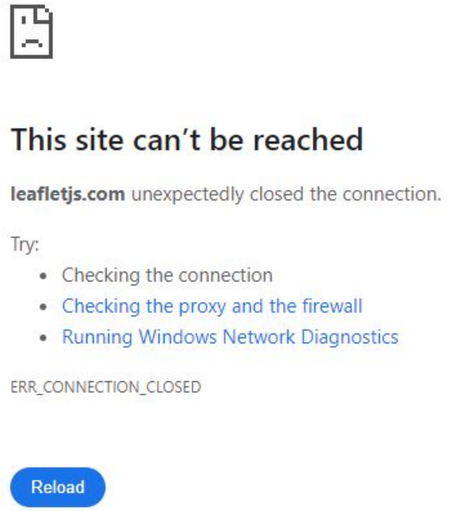
Есть экспертиза с Leaflet

Leaflet — одна из наиболее популярных JS-библиотек с открытым исходным кодом, предназначена для отображения интерактивных карт. Leaflet разработан с учетом простоты, производительности и удобства использования. Он эффективно работает на всех основных десктоп- и мобильных платформах.

Заказчик - гос. компания

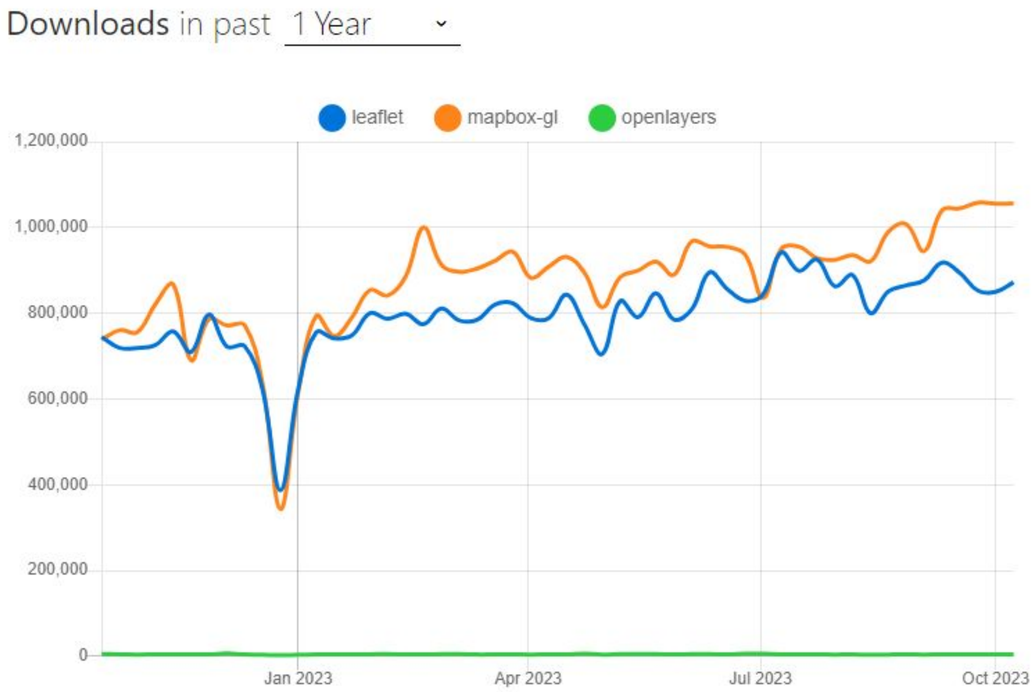
К сожалению, эти детали разглашать не можем =(

Оф. сайт Leaflet недоступен



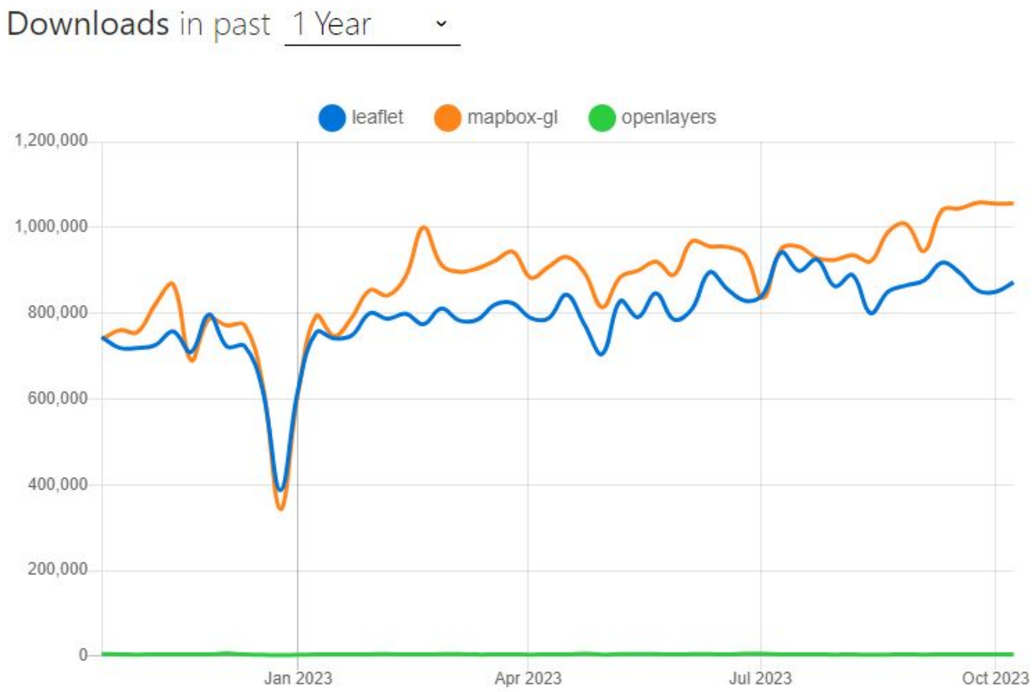
Нужна альтернатива!

В поисках альтернативы

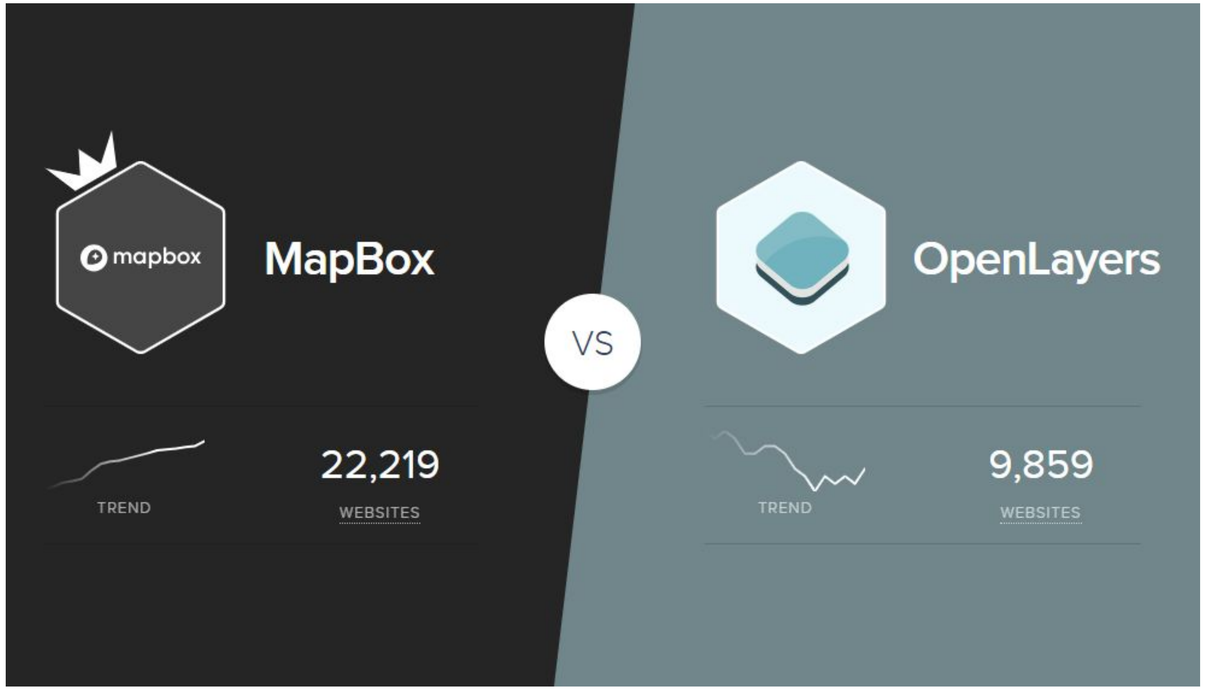


Npm trends

В поисках альтернативы



Npm trends



MapBox vs OpenLayers

Написали PoC

Отображаем карту

```
const view = new View({
    center: options.mapCenter,
    zoom: options.initialZoomLevel,
});

useGeographic();
const scaleControl = new ScaleLine({ units: 'metric' });

const tileLayer = new TileLayer({
    source: new TileWMS({
        url: environment.mapServerUrl,
        params: {LAYERS: 'default', FORMAT: 'image/png'}
    }),
});

return new Map({
    interactions: defaultInteractions()
                      .extend(options.interactions || []),
    view: view,
    layers: ([tileLayer] as Array<Layer>)
                .concat(options.vectorLayers || []),
    controls: [scaleControl],
    target: 'ol-map',
});
```


Написали PoC

Начальное
позиционирование карты

```
const view = new View({
    center: options.mapCenter,
    zoom: options.initialZoomLevel,
});

useGeographic();
const scaleControl = new ScaleLine({ units: 'metric' });

const tileLayer = new TileLayer({
    source: new TileWMS({
        url: environment.mapServerUrl,
        params: {LAYERS: 'default', FORMAT: 'image/png'}
    }),
});

return new Map({
    interactions: defaultInteractions()
                      .extend(options.interactions || []),
    view: view,
    layers: ([tileLayer] as Array<Layer>)
                .concat(options.vectorLayers || []),
    controls: [scaleControl],
    target: 'ol-map',
});
```


Написали PoC

Географические координаты

Отображение масштаба карты

```
const view = new View({
    center: options.mapCenter,
    zoom: options.initialZoomLevel,
});

useGeographic();
const scaleControl = new ScaleLine({ units: 'metric' });

const tileLayer = new TileLayer({
    source: new TileWMS({
        url: environment.mapServerUrl,
        params: {LAYERS: 'default', FORMAT: 'image/png'}
    }),
});

return new Map({
    interactions: defaultInteractions()
                      .extend(options.interactions || []),
    view: view,
    layers: ([tileLayer] as Array<Layer>)
                .concat(options.vectorLayers || []),
    controls: [scaleControl],
    target: 'ol-map',
});
```

Написали PoC

Растровый слой-«подложка»
с самой картой

```
const view = new View({
    center: options.mapCenter,
    zoom: options.initialZoomLevel,
});

useGeographic();
const scaleControl = new ScaleLine({ units: 'metric' });

const tileLayer = new TileLayer({
    source: new TileWMS({
        url: environment.mapServerUrl,
        params: {LAYERS: 'default', FORMAT: 'image/png'}
    }),
});

return new Map({
    interactions: defaultInteractions()
                      .extend(options.interactions || []),
    view: view,
    layers: ([tileLayer] as Array<Layer>)
                .concat(options.vectorLayers || []),
    controls: [scaleControl],
    target: 'ol-map',
});
```

Написали PoC

Добавление элементов интерактивности

```
const view = new View({
    center: options.mapCenter,
    zoom: options.initialZoomLevel,
});

useGeographic();
const scaleControl = new ScaleLine({ units: 'metric' });

const tileLayer = new TileLayer({
    source: new TileWMS({
        url: environment.mapServerUrl,
        params: {LAYERS: 'default', FORMAT: 'image/png'}
    }),
});

return new Map({
   interactions: defaultInteractions()
                      .extend(options.interactions || []),
    view: view,
    layers: ([tileLayer] as Array<Layer>)
                .concat(options.vectorLayers || []),
    controls: [scaleControl],
    target: 'ol-map',
});
```


Написали PoC

Добавление необходимых слоёв

```
const view = new View({
    center: options.mapCenter,
    zoom: options.initialZoomLevel,
});

useGeographic();
const scaleControl = new ScaleLine({ units: 'metric' });

const tileLayer = new TileLayer({
    source: new TileWMS({
        url: environment.mapServerUrl,
        params: {LAYERS: 'default', FORMAT: 'image/png'}
    }),
});

return new Map({
    interactions: defaultInteractions()
                      .extend(options.interactions || []),
    view: view,
    layers: ([tileLayer] as Array<Layer>)
                .concat(options.vectorLayers || []),
    controls: [scaleControl],
    target: 'ol-map',
});
```


Написали PoC

Добавляем векторные слои

```
this.labelVectorSource =
    new VectorSource<Point>({ wrapX: false });
const labelLayer =
    new VectorLayer<VectorSource<Point>>({
        source: this.labelVectorSource,
        style: this.labelStyle,
    });


this.roadVectorSource =
    new VectorSource<MultiLineString>({ wrapX: false });
const roadLayer =
    new VectorLayer<VectorSource<MultiLineString>>({
        source: this.roadVectorSource,
    });


this.drawVectorSource = new VectorSource({ wrapX: false });
const drawLayer = new VectorLayer({
    source: this.drawVectorSource,
});
```


Написали PoC

Добавляем векторные слои

```
this.labelVectorSource =
    new VectorSource<Point>({ wrapX: false });
const labelLayer =
    new VectorLayer<VectorSource<Point>>({
        source: this.labelVectorSource,
        style: this.labelStyle,
    });
```

```
this.roadVectorSource =
    new VectorSource<MultiLineString>({ wrapX: false });
const roadLayer =
    new VectorLayer<VectorSource<MultiLineString>>({
        source: this.roadVectorSource,
    });
```

```
this.drawVectorSource = new VectorSource({ wrapX: false });
const drawLayer = new VectorLayer({
    source: this.drawVectorSource,
});
```


Написали PoC

Добавляем векторные слои

```
this.labelVectorSource =
    new VectorSource<Point>({ wrapX: false });
const labelLayer =
    new VectorLayer<VectorSource<Point>>({
        source: this.labelVectorSource,
        style: this.labelStyle,
    });
```

```
this.roadVectorSource =
    new VectorSource<MultiLineString>({ wrapX: false });
const roadLayer =
    new VectorLayer<VectorSource<MultiLineString>>({
        source: this.roadVectorSource,
    });
```

```
this.drawVectorSource = new VectorSource({ wrapX: false });
const drawLayer = new VectorLayer({
    source: this.drawVectorSource,
});
```


Написали PoC

Добавляем векторные слои

```
this.labelVectorSource =
    new VectorSource<Point>({ wrapX: false });
const labelLayer =
        new VectorLayer<VectorSource<Point>>({
        source: this.labelVectorSource,
        style: this.labelStyle,
    });


this.roadVectorSource =
    new VectorSource<MultiLineString>({ wrapX: false });
const roadLayer =
    new VectorLayer<VectorSource<MultiLineString>>({
        source: this.roadVectorSource,
    });


this.drawVectorSource = new VectorSource({ wrapX: false });
const drawLayer = new VectorLayer({
    source: this.drawVectorSource,
});
```


Написали PoC

Стилизуем маркеры

```
new Style({
    image: new Icon({
        height: isBig ? 32 : 20,
        src: `/assets/markers_stop.svg`,
    }),
    zIndex: 5,
});
```

```
new Style({
    fill: new Fill({
        color: [64, 169, 255, 0.35],
    }),

    stroke: new Stroke({
        color: 'green',
        width: 2,
        lineDash: [8, 12],
    }),
});
```


Написали PoC

Стилизуем маркеры

```
new Style({
    image: new Icon({
        height: isBig ? 32 : 20,
        src: `/assets/markers_stop.svg`,
    }),
    zIndex: 5,
});
```

```
new Style({
    fill: new Fill({
        color: [64, 169, 255, 0.35],
    }),

    stroke: new Stroke({
        color: 'green',
        width: 2,
        lineDash: [8, 12],
    }),
});
```


Написали PoC

Стилизуем маркеры

```
new Style({
    image: new Icon({
        height: isBig ? 32 : 20,
        src: `/assets/markers_stop.svg`,
    }),
    zIndex: 5,
});
```

```
new Style({
    fill: new Fill({
        color: [64, 169, 255, 0.35],
    }),

    stroke: new Stroke({
        color: 'green',
        width: 2,
        lineDash: [8, 12],
    }),
});
```


Написали PoC

Рисуем фигуры

```
this.draw = new Draw({
    source: this.drawVectorSource,
    type: 'LineString',
});
```

```
this.map.addInteraction(this.draw);
```

```
this.draw = new Draw({
    source: this.drawVectorSource,
    type: 'Polygon',
    geometryFunction: createRegularPolygon(6),
});
```


Написали PoC

Рисуем фигуры

```
this.draw = new Draw({
    source: this.drawVectorSource,
    type: 'LineString',
});
```

```
this.map.addInteraction(this.draw);
```

```
this.draw = new Draw({
    source: this.drawVectorSource,
    type: 'Polygon',
    geometryFunction: createRegularPolygon(6),
});
```


Написали PoC

Рисуем фигуры

```
this.draw = new Draw({
    source: this.drawVectorSource,
    type: 'LineString',
});
```

```
this.map.addInteraction(this.draw);
```

```
this.draw = new Draw({
    source: this.drawVectorSource,
    type: 'Polygon',
    geometryFunction: createRegularPolygon(6),
});
```


Написали PoC

Рисуем фигуры

```
this.draw = new Draw({
    source: this.drawVectorSource,
    type: 'LineString',
});


this.map.addInteraction(this.draw);


this.draw = new Draw({
    source: this.drawVectorSource,
    type: 'Polygon',
    geometryFunction: createRegularPolygon(6),
});
```


Написали PoC

Модификация фигуры
буквально в 2 строки

```
this.modifyShapeInteraction = new Modify({
    source: this.drawVectorSource,
    pixelTolerance: 10,
});
```

```
return new Map({
    interactions:
        defaultInteractions()
            .extend([this.modifyShapeInteraction]),
    view: view,
    layers: ([...]),
    controls: [scaleControl],
    target: 'ol-map',
});
```


Написали PoC

Модификация фигуры
буквально в 2 строки

```
this.modifyShapeInteraction = new Modify({
    source: this.drawVectorSource,
    pixelTolerance: 10,
});

return new Map({
    interactions:
        defaultInteractions()
            .extend([this.modifyShapeInteraction]),
    view: view,
    layers: ([...]),
    controls: [scaleControl],
    target: 'ol-map',
});
```


Написали PoC

Модификация фигуры
буквально в 2 строки

```
this.modifyShapeInteraction = new Modify({
    source: this.drawVectorSource,
    pixelTolerance: 10,
});

return new Map({
    interactions:
        defaultInteractions()
            .extend([this.modifyShapeInteraction]),
    view: view,
    layers: ([...]),
    controls: [scaleControl],
    target: 'ol-map',
});
```

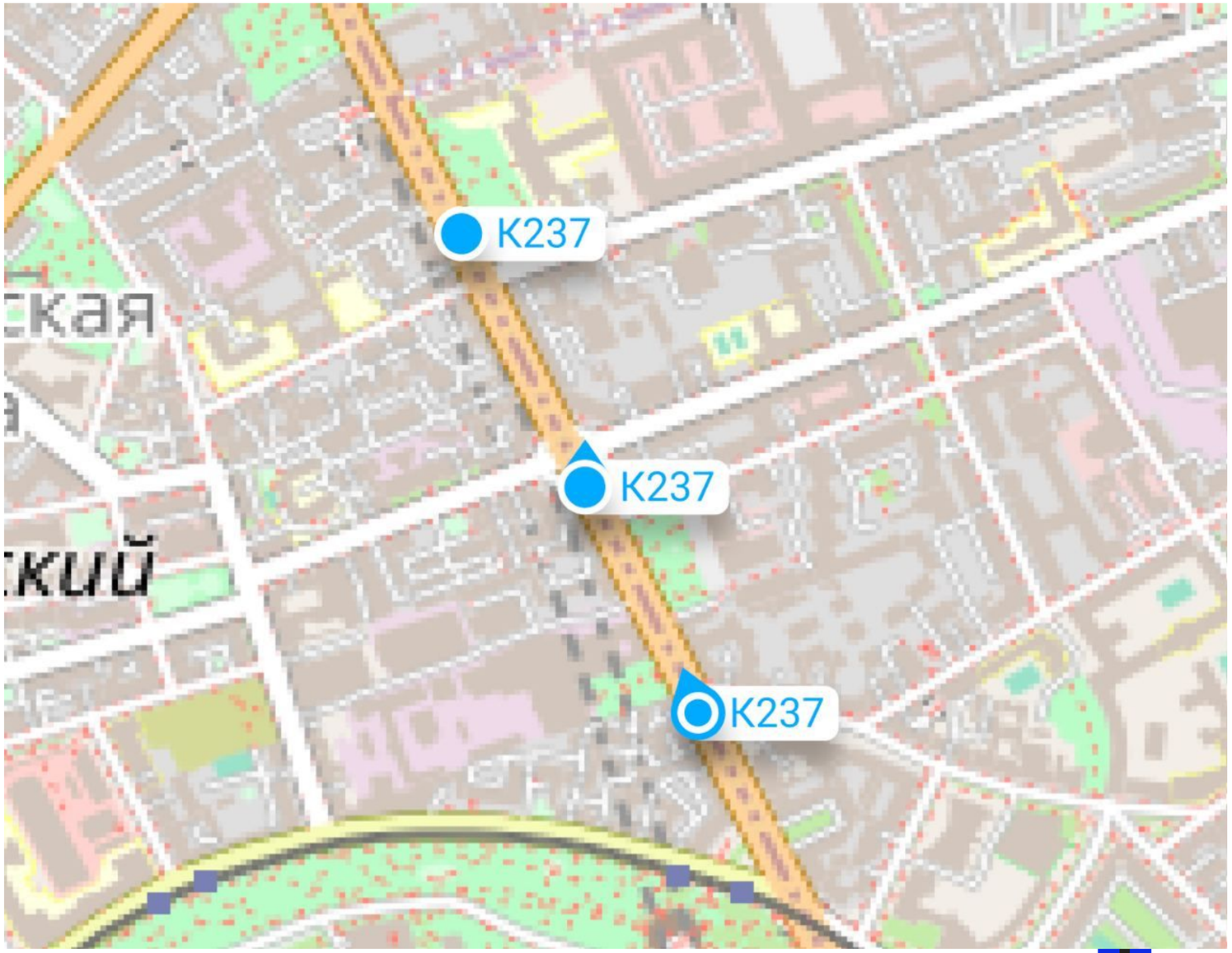

Написали PoC

Первые результаты

Восторг!!!

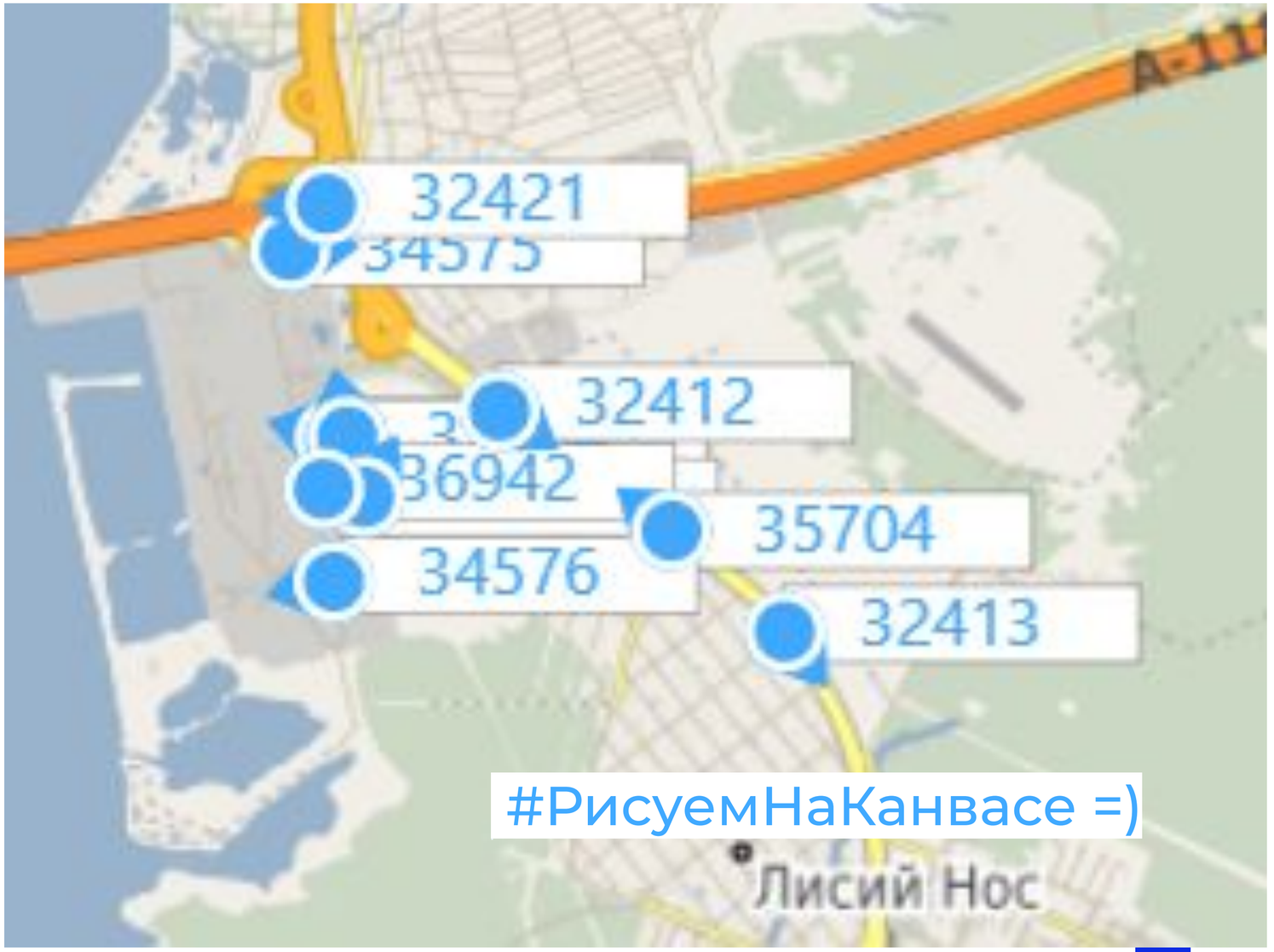
Проблемы лейблов

Заказчик хотел это



Проблемы лейблов

Результат после первой итерации




SO CLOSE

Проблемы лейблов

Как именно библиотека позволяет стилизовать маркеры на карте:

Проблемы лейблов

Как именно библиотека позволяет стилизовать маркеры на карте:

Fill и Stroke

Простая заливка цветом и отрисовка границы элемента

Проблемы лейблов

Как именно библиотека позволяет стилизовать маркеры на карте:

Fill и Stroke

Простая заливка цветом и отрисовка границы элемента

Image

Нужен для отображения картинки/иконки в качестве самого маркера

Проблемы лейблов

Как именно библиотека позволяет стилизовать маркеры на карте:

Fill и Stroke

Простая заливка цветом и отрисовка границы элемента

Image

Нужен для отображения картинки/иконки в качестве самого маркера

Text

Подпись в виде текста, которому можно навесить fill и stroke

– Нет скругления углов

– Нет скругления углов
– Нет возможности добавить тень

– Нет скругления углов
– Нет возможности добавить тень
– Нет большинства других CSS-свойств

А может, не надо?..

А может, не надо?..

Надо, ребята. Надо...

Просто добавь слоёв!

```
const textStyle = new Style({
    text: new Text({
        text: `${coordinates.boardNumber}`,
        textAlign: 'left',
        fill: new Fill({ color: '#40A9FF' }),
        font: labelsFont,
        padding: [6, 20, 0, 24],
        offsetX: 24,
        offsetY: 1,
    }),
    zIndex: coordinates.vehicleId,
});
```

Просто добавь слоёв!

Плюсы:

- оно работает!
- понятен порядок слоёв

```
const textStyle = new Style({
    text: new Text({
        text: `${coordinates.boardNumber}`,
        textAlign: 'left',
        fill: new Fill({ color: '#40A9FF' }),
        font: labelsFont,
        padding: [6, 20, 0, 24],
        offsetX: 24,
        offsetY: 1,
    }),
    zIndex: coo
});
```

```
const vehicleMarkerStyle = new Style({
    image: new Icon({
        height: 46,
        width: 46,
        src: '/assets/marker.svg',
        anchor: [0.5, 0.5],
        rotation: coordinates.direction *
Math.PI / 180,
    }),
    zIndex: coordinates.vehicleId,
});
```

Просто добавь слоёв!

Плюсы:

- оно работает!
- понятен порядок слоёв

Минусы:

- сложная логика
- хрупкая структура
- на большом количестве ТС подтормаживает

```
const textStyle = new Style({
    text: new Text({
        text: `${coordinates.boardNumber}`,
        textAlign: 'left'
```

```
const backWidth = textWidth + shadowGap + radiusGap + leftPadding;
const backStyle = new Style({
    image: new Icon({
        height: 56,
        width: backWidth,
        src: '/assets/vehicle-label-back.svg',
        anchor: [0.1, 0.35],
    }),
    zIndex: coordinates.vehicleId,
});
```

```
        anchor: [0.5, 0.5],
        rotation: coordinates.direction *
Math.PI / 180,
    }),
    zIndex: coordinates.vehicleId,
});
```


Что могло пойти не так?..

Что могло пойти не так?..

Его Величество Редизайн*

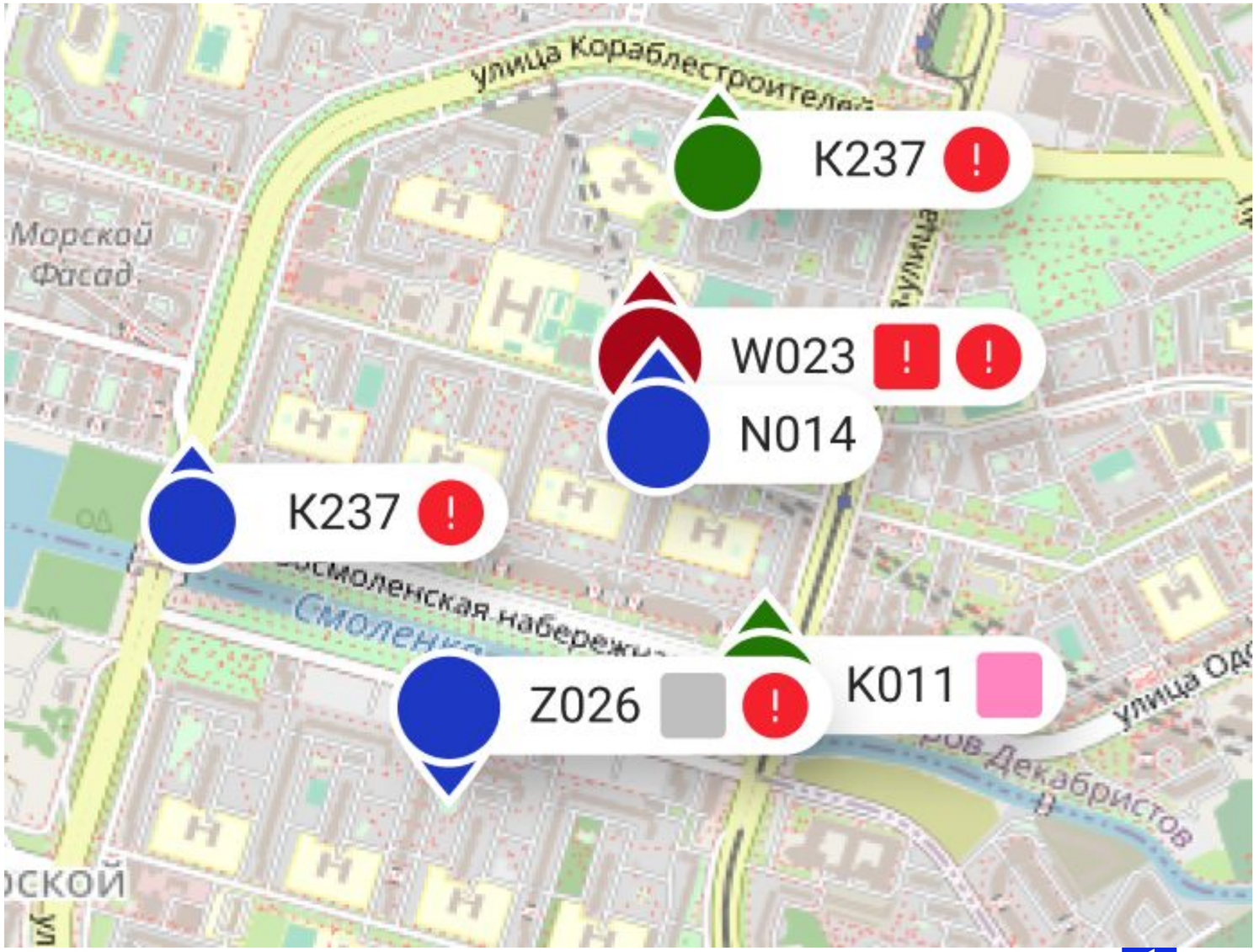
Что могло пойти не так?..

Его Величество Редизайн*

* да, даже на этой стадии проекта

Проблемы лейблов

Теперь заказчик хочет это



Просто добавь слоёв!

```
const shadowGap = 28;
const radiusGap = 8;
const leftPadding = 24;
const bnsoGap = coords.hasEquipFault ? 22 : 0;
const emergencyGap =
  coords.vehicleStatus === VehicleStatus.Emergency ? 20 : 0;
const extraGap = bnsoGap && emergencyGap ? 6 : 0;

const backStyle = new Style({
  image: new Icon({
    height: 56,
    width: textWidth + shadowGap + radiusGap + leftPadding
        + bnsoGap + emergencyGap + extraGap,
    src: '/assets/vehicle-label-back.svg',
    anchor: [0.1, 0.35],
  }),
  zIndex: coords.vehicleId,
});

const styles = [backStyle, textStyle, vehicleMarkerStyle,
    vehicleIconStyle, bnsoStyle, emergencyStyle, statusStyle];
```


Просто добавь слоёв!

```
const shadowGap = 28;
const radiusGap = 8;
const leftPadding = 24;
const bnsoGap = coords.hasEquipFault ? 22 : 0;
const emergencyGap =
  coords.vehicleStatus === VehicleStatus.Emergency ? 20 : 0;
const extraGap = bnsoGap && emergencyGap ? 6 : 0;

const backStyle = new Style({
  image: new Icon({
    height: 56,
    width: textWidth + shadowGap + radiusGap + leftPadding
        + bnsoGap + emergencyGap + extraGap,
    src: '/assets/vehicle-label-back.svg',
    anchor: [0.1, 0.35],
  }),
  zIndex: coords.vehicleId,
});

const styles = [backStyle, textStyle, vehicleMarkerStyle,
    vehicleIconStyle, bnsoStyle, emergencyStyle, statusStyle];
```


Просто добавь слоёв!

```
const shadowGap = 28;
const radiusGap = 8;
const leftPadding = 24;
const bnsoGap = coords.hasEquipFault ? 22 : 0;
const emergencyGap =
  coords.vehicleStatus === VehicleStatus.Emergency ? 20 : 0;
const extraGap = bnsoGap && emergencyGap ? 6 : 0;

const backStyle = new Style({
  image: new Icon({
    height: 56,
    width: textWidth + shadowGap + radiusGap + leftPadding
        + bnsoGap + emergencyGap + extraGap,
    src: '/assets/vehicle-label-back.svg',
    anchor: [0.1, 0.35],
  }),
  zIndex: coords.vehicleId,
});

const styles = [backStyle, textStyle, vehicleMarkerStyle,
    vehicleIconStyle, bnsoStyle, emergencyStyle, statusStyle];
```

Альтернативы?

- Делать более простые лейблы
- Выносить больше информации в попапы
- Расширять фильтрацию

Этап пройден

Как ездят машинки


```
{
    "vehicleId": 23238,
    "vehicleType": "Автобус",
    "boardNumber": "34569",
    "longitude": 30.255013,
    "latitude": 59.990498,
    "direction": 86,
    "hasEquipFault": true,
    "vehicleStatus": 4,
    "bnsoStatus": 3
},
{
    "vehicleId": 25196,
    "vehicleType": "Троллейбус",
    "boardNumber": "32517",
    "longitude": 29.724106,
    "latitude": 60.016846,
    "direction": 45,
    "hasEquipFault": false,
    "vehicleStatus": 2,
    "bnsoStatus": 1
},
```

Как ездят машинки

```
const line = new LineString([
    [prevCoordinates.longitude, prevCoordinates.latitude],
    [newCoordinates.longitude, newCoordinates.latitude],
]);

let step = 0;
const key = setInterval(() => {
    if (step < 100) {
        step++;
        feature.setGeometry(
            new Point(line.getCoordinateAt(step/100))
        );
    } else {
        clearInterval(key);
    }
}, 50);
```


Как ездят машинки

```
const line = new LineString([
    [prevCoordinates.longitude, prevCoordinates.latitude],
    [newCoordinates.longitude, newCoordinates.latitude],
]);

let step = 0;
const key = setInterval(() => {
    if (step < 100) {
        step++;
        feature.setGeometry(
            new Point(line.getCoordinateAt(step/100))
        );
    } else {
        clearInterval(key);
    }
}, 50);
```


Как ездят машинки

```
const line = new LineString([
    [prevCoordinates.longitude, prevCoordinates.latitude],
    [newCoordinates.longitude, newCoordinates.latitude],
]);

let step = 0;
const key = setInterval(() => {
    if (step < 100) {
        step++;
        feature.setGeometry(
            new Point(line.getCoordinateAt(step/100))
        );
    } else {
        clearInterval(key);
    }
}, 50);
```

Как ездят машинки

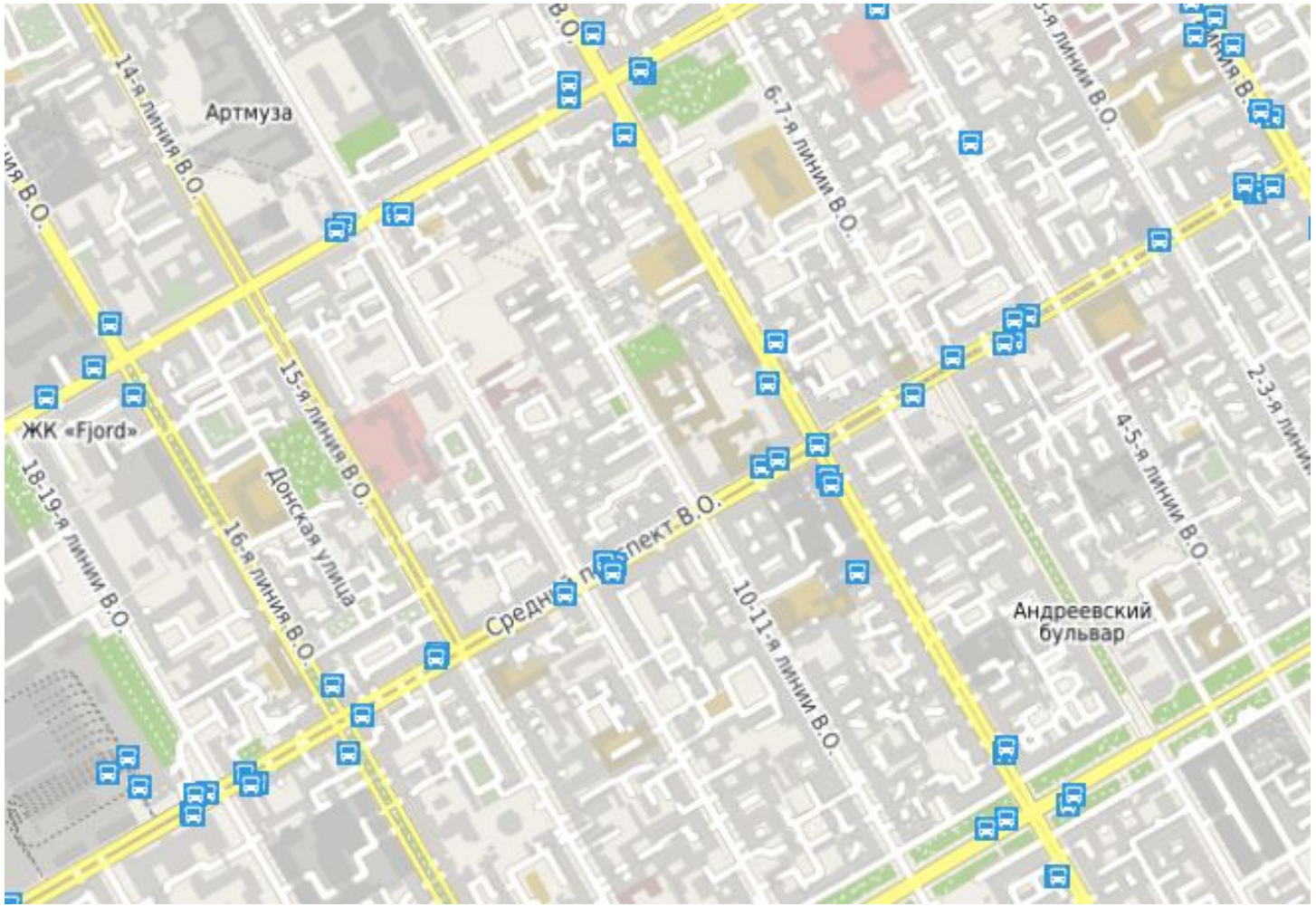
```
const line = new LineString([
    [prevCoordinates.longitude, prevCoordinates.latitude],
    [newCoordinates.longitude, newCoordinates.latitude],
]);

let step = 0;
const key = setInterval(() => {
    if (step < 100) {
        step++;
        feature.setGeometry(
            new Point(line.getCoordinateAt(step/100))
        );
    } else {
        clearInterval(key);
    }
}, 50);
```


Работаем с объектами на карте

Снова проблемы?..

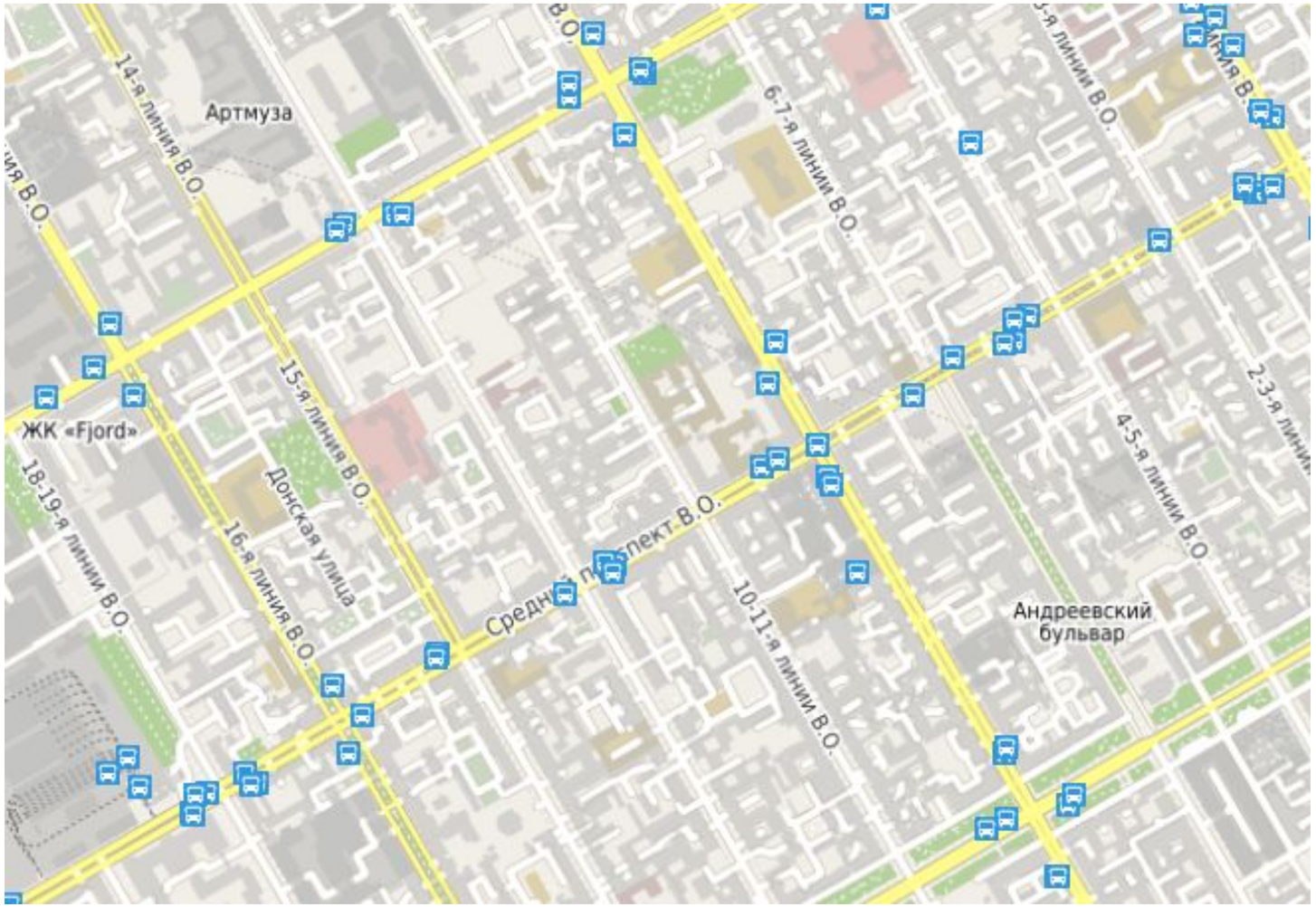
Остановки «на минималках»

- снова маркеры



Остановки «на минималках»

- снова маркеры
- иконка зависит от типа остановки



Остановки в действии

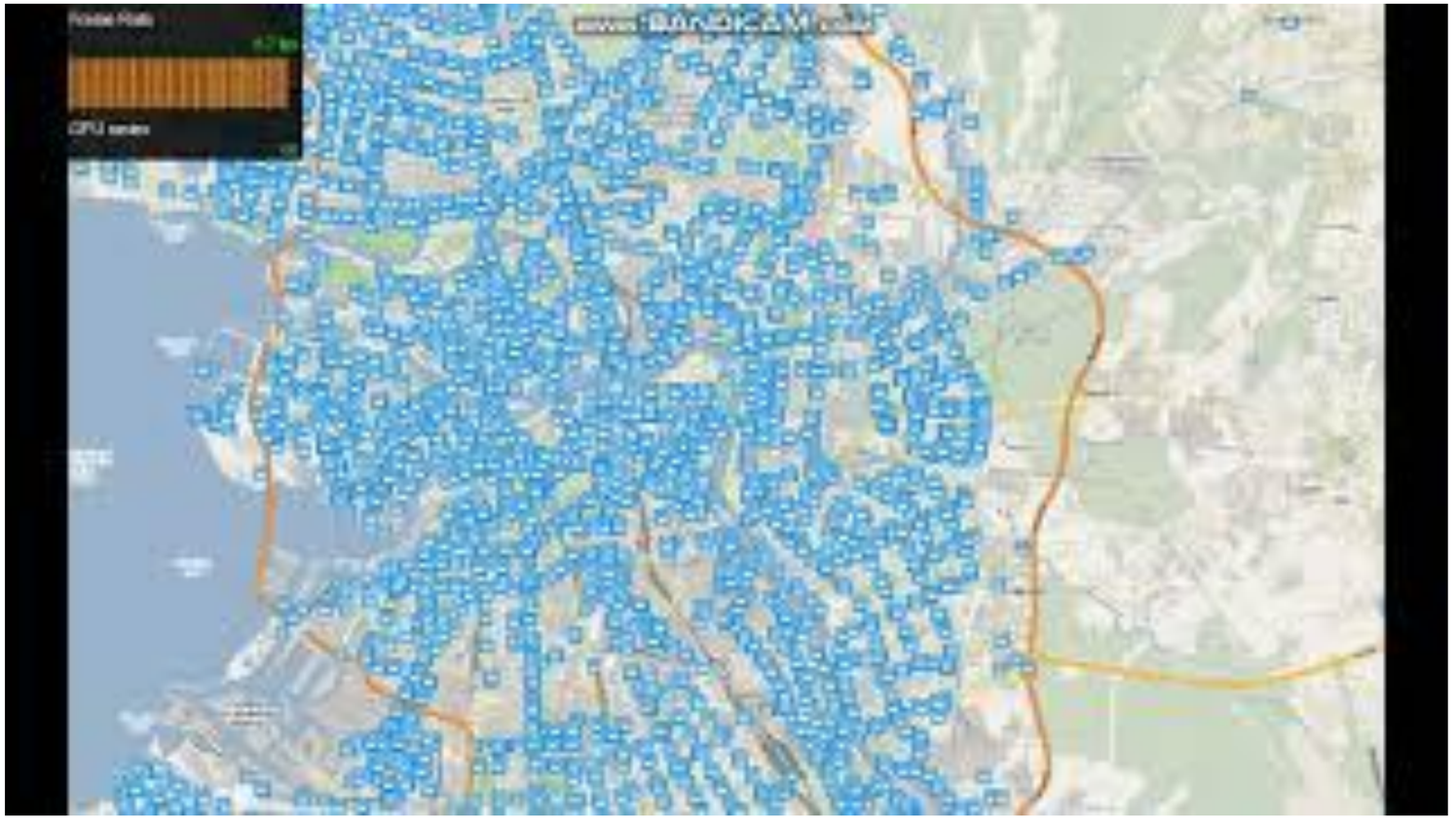
А давай покажем всё?

Остановки в действии

Почему???

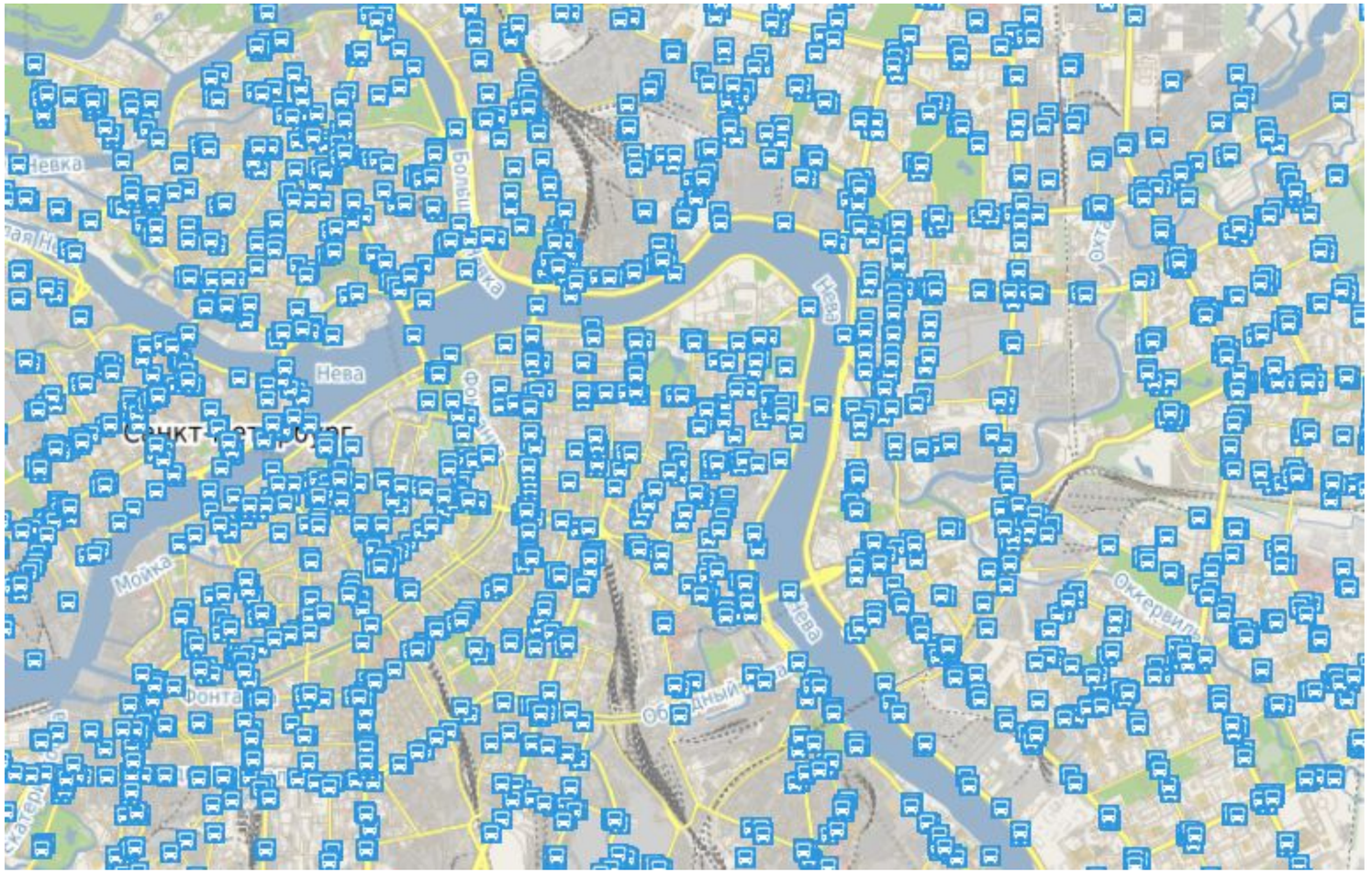
Проверка гипотез

- грешим на лишние рендеры



Проверка гипотез

- грешим на лишние рендеры
- а если сделать кластеризацию?



Проверка гипотез

- грешим на лишние рендеры
- а если сделать кластеризацию?
- ещё варианты?

ol/layer/VectorImage-VectorImageLayer

```
import VectorImageLayer from 'ol/layer/VectorImage.js';
```

Vector data is rendered client-side, to an image. This layer type provides great performance during panning and zooming, but point symbols and texts are always rotated with the view and pixels are scaled during zoom animations. For more accurate rendering of vector data, use `VectorLayer` instead.

Проверка гипотез

- ~~грешим на лишние рендеры~~
- ~~а если сделать кластеризацию?~~
- ~~ещё варианты?~~
- виртуальный скролл
- ограничение для зума

Проверка гипотез

- ~~грешим на лишние рендеры~~
- ~~а если сделать кластеризацию?~~
- ~~ещё варианты?~~
- виртуальный скролл
- ограничение для зума

```
getFeaturesInExtent(extent, projection) {
  if (this.featuresRtree_) {
    const multiWorld =
    projection && projection.canWrapX() && this.getWrapX();

    if (!multiWorld) {
      return this.featuresRtree_.getInExtent(extent);
    }

    const extents = wrapAndSliceX(extent, projection);
    return [].concat(
      ...extents.map((anExtent) =>
        this.featuresRtree_.getInExtent(anExtent))
    );
  }
  if (this.featuresCollection_) {
    return this.featuresCollection_.getArray().slice(0);
  }
  return [];
}
```


Проверка гипотез

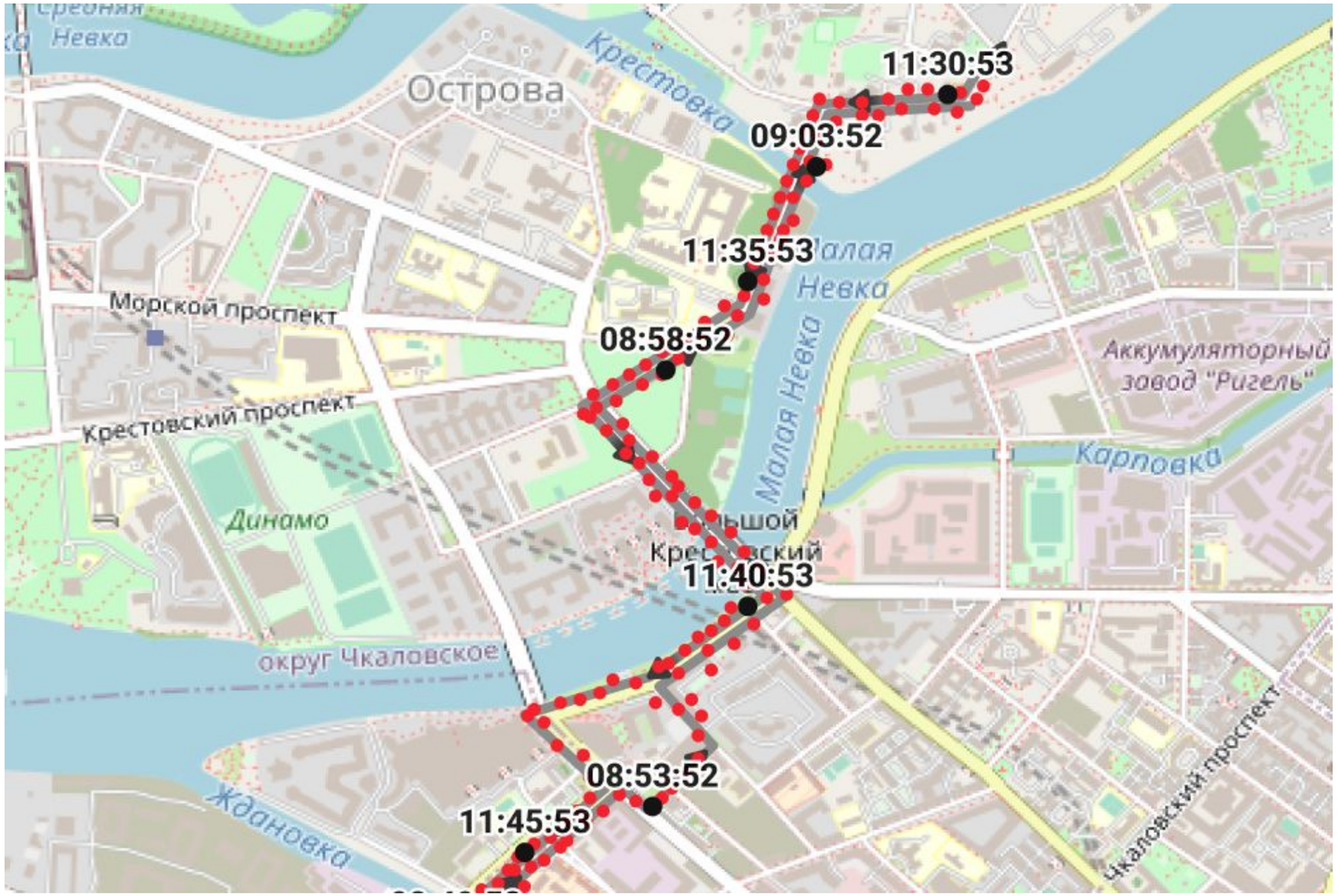
- ~~грешим на лишние рендеры~~
- ~~а если сделать кластеризацию?~~
- ~~ещё варианты?~~
- виртуальный скролл
- ограничение для зума

```
getFeaturesInExtent(extent, projection) {
  if (this.featuresRtree_) {
    const multiWorld =
    projection && projection.canWrapX() && this.getWrapX();

    if (!multiWorld) {
      return this.featuresRtree_.getInExtent(extent);
    }

    const extents = wrapAndSliceX(extent, projection);
    return [].concat(
      ...extents.map((anExtent) =>
        this.featuresRtree_.getInExtent(anExtent))
    );
  }
  if (this.featuresCollection_) {
    return this.featuresCollection_.getArray().slice(0);
  }
  return [];
}
```


Трек движения ТС
и 86к координат
на карте

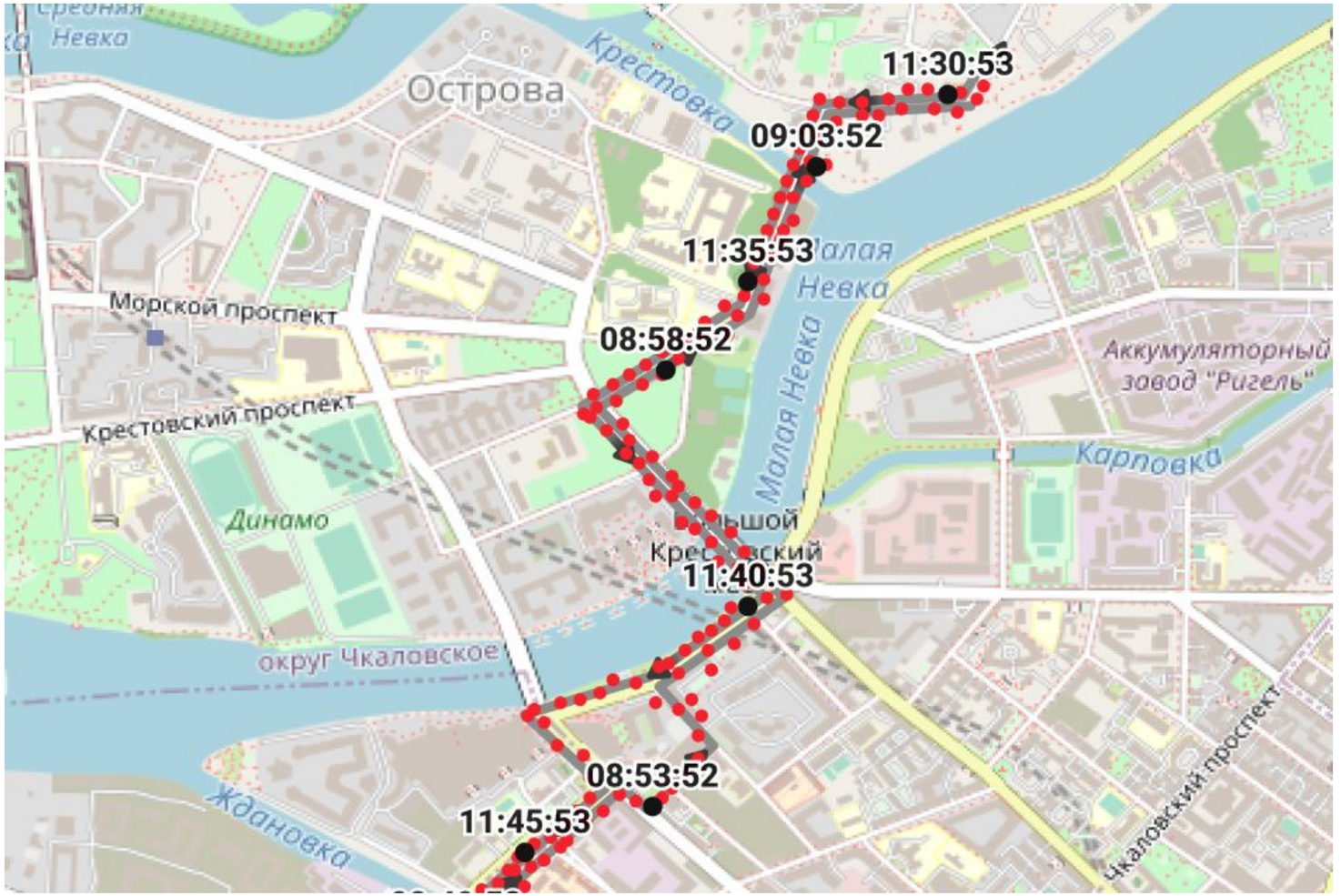
Трек движения ТС

- показываем 11к точек



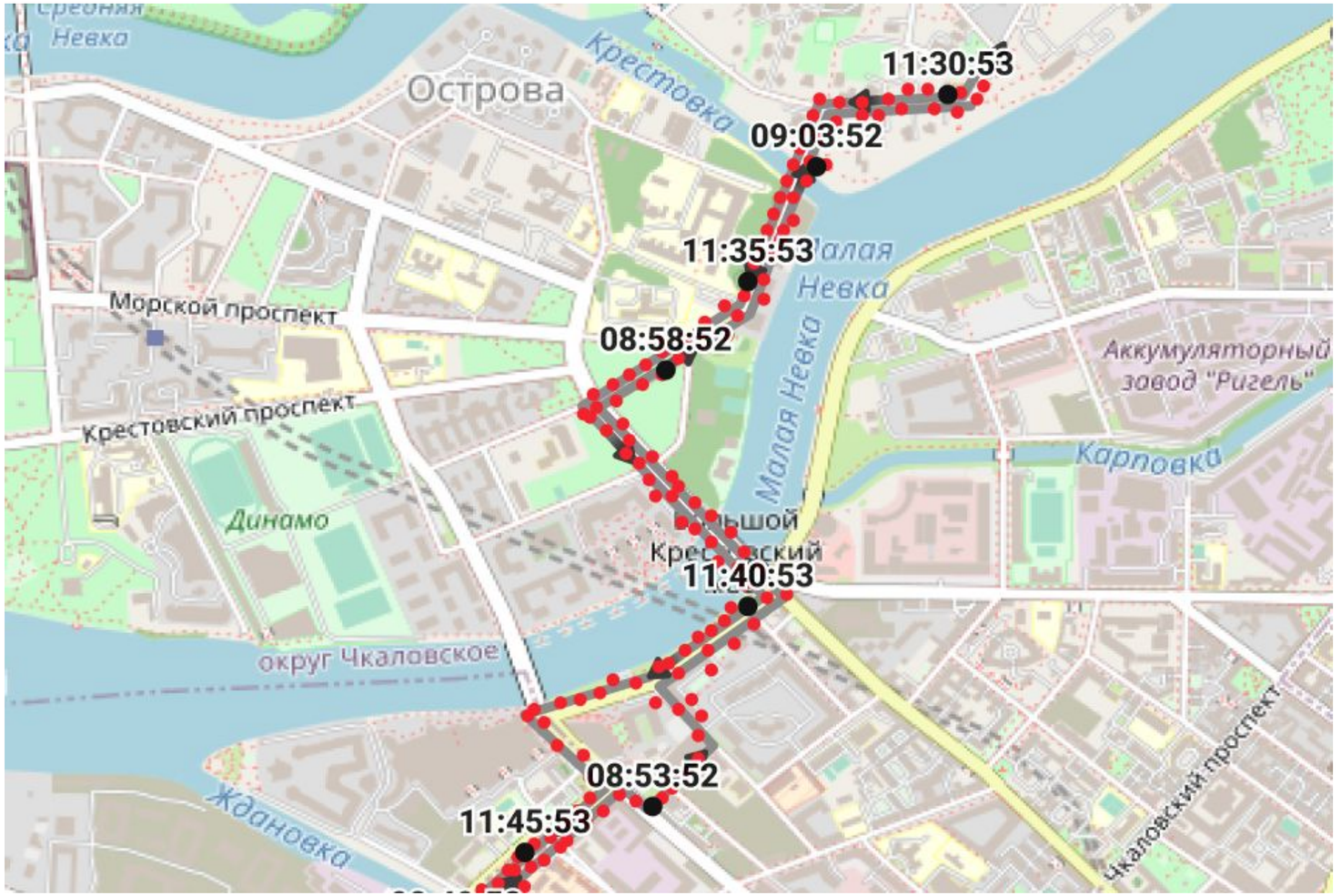
Трек движения ТС

- показываем 11к точек
- 11к => 86к



Трек движения ТС

- показываем 11к точек
- 11к => 86к
- помните слайды про перформанс? =)



Трек движения ТС

- отказываемся от соединения точек на фронте
- получаем подготовленные бэком координаты для трека
- отображаем “сырые” точки

```
const lineString = new LineString(allCoords);
const trackFeature = new Feature({
    type: 'historyTrack',
    geometry: lineString,
    vehicleId: track[0].transportId,
});

this.vehicleTrackVectorSource.addFeature(trackFeature);


const coordinate =
    <Coordinate>[trackPoint.longitude, trackPoint.latitude];
const rawPoint = new Feature({
    type: 'rawPoint',
    geometry:  new Point(coordinate),
    vehicleId: rawPoints[0].transportId,
});

this.vehicleTrackRawPointVectorSource.addFeature(rawPoint);
```


Трек движения ТС

- отказываемся от соединения точек на фронте
- получаем подготовленные бэком координаты для трека
- отображаем “сырые” точки

```
const lineString = new LineString(allCoords);
const trackFeature = new Feature({
    type: 'historyTrack',
    geometry: lineString,
    vehicleId: track[0].transportId,
});

this.vehicleTrackVectorSource.addFeature(trackFeature);


const coordinate =
    <Coordinate>[trackPoint.longitude, trackPoint.latitude];
const rawPoint = new Feature({
    type: 'rawPoint',
    geometry:  new Point(coordinate),
    vehicleId: rawPoints[0].transportId,
});

this.vehicleTrackRawPointVectorSource.addFeature(rawPoint);
```

Трек движения ТС

- отказываемся от соединения точек на фронте
- получаем подготовленные бэком координаты для трека
- отображаем “сырые” точки

```
const lineString = new LineString(allCoords);
const trackFeature = new Feature({
    type: 'historyTrack',
    geometry: lineString,
    vehicleId: track[0].transportId,
});

this.vehicleTrackVectorSource.addFeature(trackFeature);
```

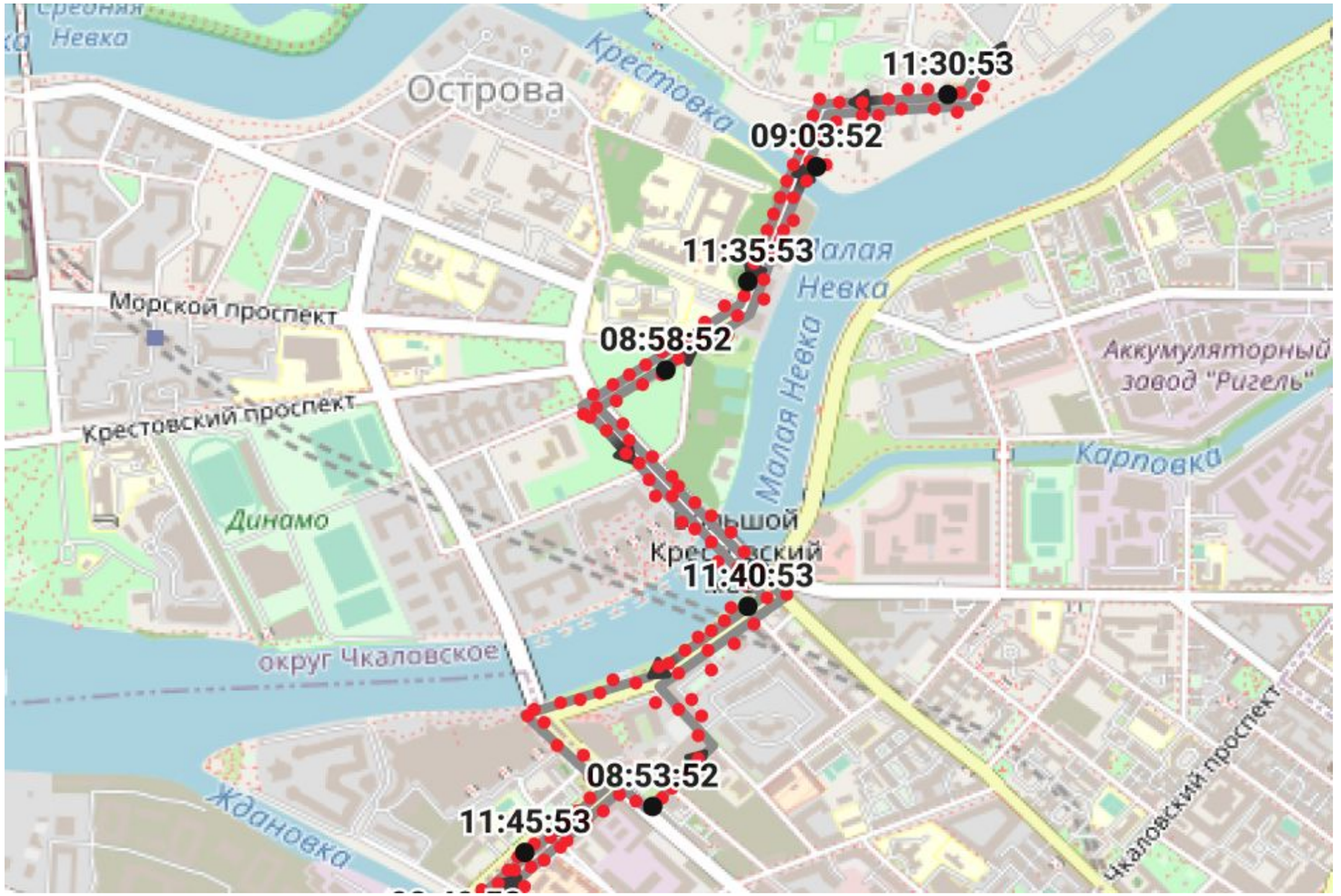
```
const coordinate =
    <Coordinate>[trackPoint.longitude, trackPoint.latitude];
const rawPoint = new Feature({
    type: 'rawPoint',
    geometry:  new Point(coordinate),
    vehicleId: rawPoints[0].transportId,
});

this.vehicleTrackRawPointVectorSource.addFeature(rawPoint);
```


Трек движения ТС

- да, мы сильнее нагрузили бэк
- да, бэку пришлось частично пересесть с pgRouting на osrm

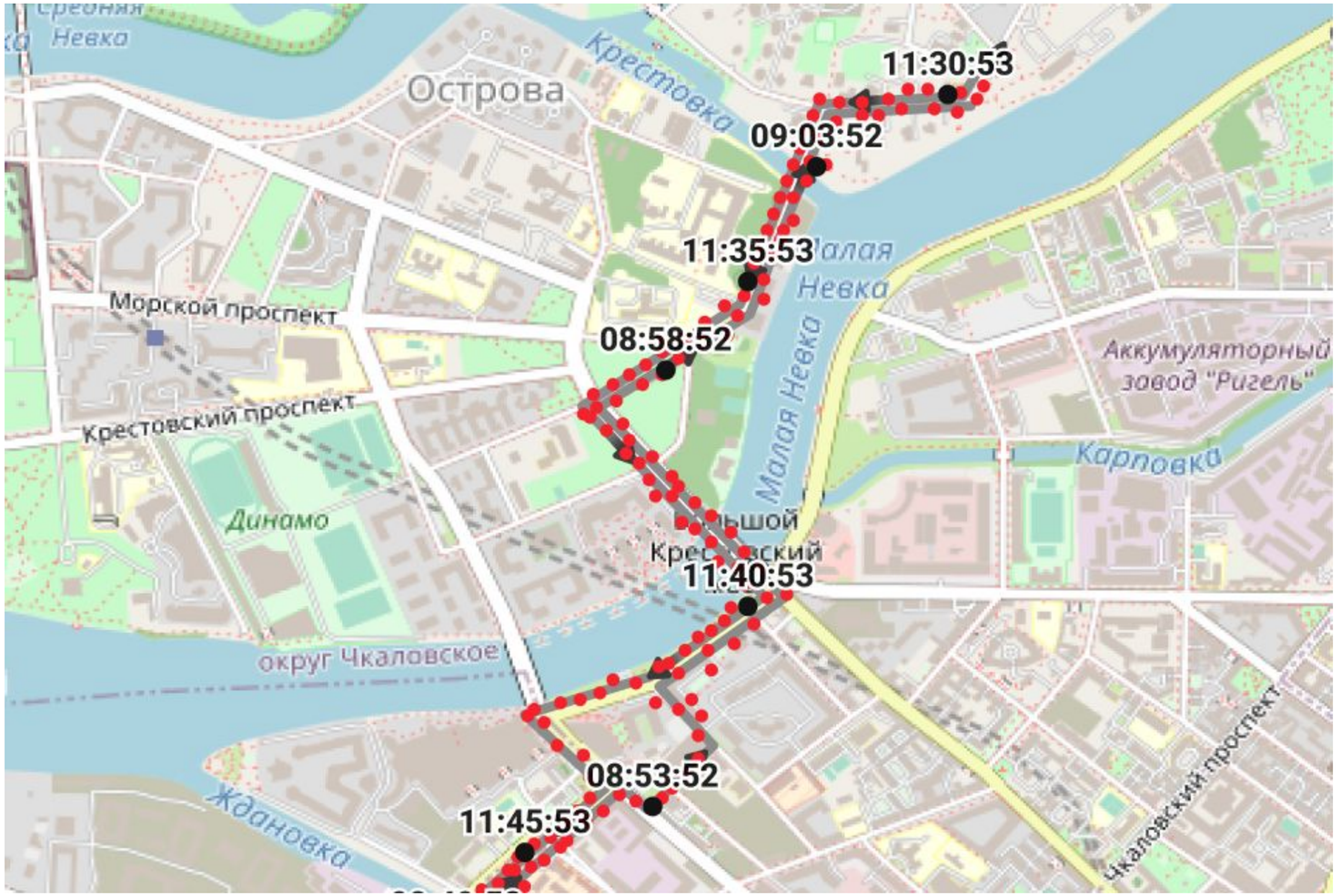
Трек движения ТС

- да, мы сильнее нагрузили бэк
- да, бэку пришлось частично пересесть с pgRouting на osrm
- это не наши проблемы :D



Трек движения ТС

- да, мы сильнее нагрузили бэк
- да, бэку пришлось частично пересесть с pgRouting на osrm
- это не наши проблемы :D
- карта подлагивает, но терпимо
- далее проблем не было



Итоги

- приёмка прошла успешно

Итоги

- приёмка прошла успешно
- проект в продакшне уже полгода

Итоги

- приёмка прошла успешно
- проект в продакшне уже полгода
- критических багов, связанных с картой, нет

А что, если?..

Проблема лэйблов. Leaflet Edition

```
const map = L.map('map').setView(L.latLng(coordinates), 12);
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

for (const vehicle of vehicles) {
  new L.marker(L.latLng([vehicle.latitude,
vehicle.longitude]),
  {
    icon: new L.DivIcon({
      className: 'vehicle-marker',
      html: getMarkerHtml(vehicle),
    })
  }).addTo(map);
}
```


Проблема лэйблов. Leaflet Edition

Создание карты

```
const map = L.map('map').setView(L.latLng(coordinates), 12);
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

for (const vehicle of vehicles) {
  new L.marker(L.latLng([vehicle.latitude,
vehicle.longitude]),
  {
    icon: new L.DivIcon({
      className: 'vehicle-marker',
      html: getMarkerHtml(vehicle),
    })
  }).addTo(map);
}
```


Проблема лэйблов. Leaflet Edition

Добавление маркера в DOM

```
const map = L.map('map').setView(L.latLng(coordinates), 12);
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

for (const vehicle of vehicles) {
  new L.marker(L.latLng([vehicle.latitude,
vehicle.longitude]),
  {
    icon: new L.DivIcon({
      className: 'vehicle-marker',
      html: getMarkerHtml(vehicle),
    })
  }).addTo(map);
}
```


Проблема лэйблов. Leaflet Edition

```
function getMarkerHtml(vehicle) {
  return `
    <div class="vehicle-label">
      <span>${vehicle.boardNumber}</span>
      ${getStatusPart(vehicle.vehicleStatus)}
      ${vehicle.hasEquipFault ?
          '<img src="/assets/alert.svg"/>' : ''
      }
    </div>
    <img
      class="vehicle-image" src="/assets/vehicle-icon.svg"
      style="transform: rotateZ(${vehicle.direction}deg);"
    />`;
}

function getStatusPart(status) {
  if (status < 4) return '';

  const cssClass = `vh-status ${getStatusClass(status)}`;
  return `<div class="${cssClass}"></div>`;
}
```

Проблема лэйблов. Leaflet Edition

Разметка лэйбла

```
function getMarkerHtml(vehicle) {
  return `
    <div class="vehicle-label">
      <span>${vehicle.boardNumber}</span>
      ${getStatusPart(vehicle.vehicleStatus)}
      ${vehicle.hasEquipFault ?
          '<img src="/assets/alert.svg"/>' : ''
      }
    </div>
    <img
      class="vehicle-image" src="/assets/vehicle-icon.svg"
      style="transform: rotateZ(${vehicle.direction}deg);"
    />`;
}

function getStatusPart(status) {
  if (status < 4) return '';

  const cssClass = `vh-status ${getStatusClass(status)}`;
  return `<div class="${cssClass}"></div>`;
}
```

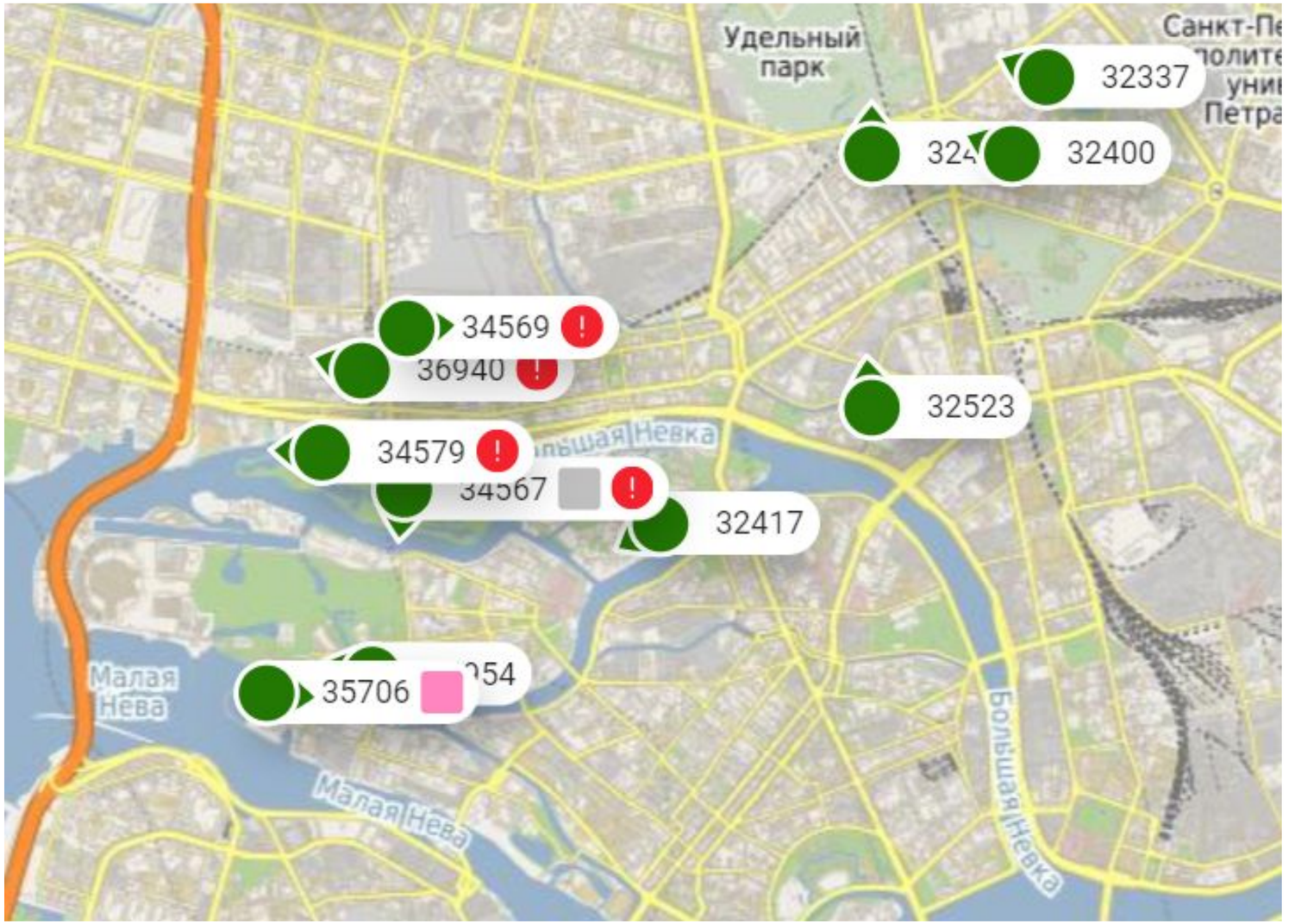
Проблема лэйблов. Leaflet Edition

Разметка для статуса ТС

```
function getMarkerHtml(vehicle) {
  return `
    <div class="vehicle-label">
      <span>${vehicle.boardNumber}</span>
      ${getStatusPart(vehicle.vehicleStatus)}
      ${vehicle.hasEquipFault ?
          '<img src="/assets/alert.svg"/>' : ''
      }
    </div>
    <img
      class="vehicle-image" src="/assets/vehicle-icon.svg"
      style="transform: rotateZ(${vehicle.direction}deg);"
    />`;
}

function getStatusPart(status) {
  if (status < 4) return '';

  const cssClass = `vh-status ${getStatusClass(status)}`;
  return `<div class="${cssClass}"></div>`;
}
```

Проблема лэйблов. Leaflet Edition

Лэйблы на карте



Остановки в действии. Leaflet Edition

```
const icon = L.icon({
  iconUrl: './assets/marker_stop.svg',
  iconSize: [12, 12],
  iconAnchor: [8, 8]
});

const map = L.map('map').setView(L.latLng(coordinates), 12);
const markersCanvas = new L.MarkersCanvas();
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

let markers = [];
for (const point of points) {
  markers.push(createMarker(point));
}

markersCanvas.addMarkers(markers);
markersCanvas.addTo(map);

function createMarker(point) {
  return new L.marker(L.latLng([point.y, point.x]), {
    icon: icon,
  });
}
```

Остановки в действии. Leaflet Edition

Создание иконки остановки

```
const icon = L.icon({
  iconUrl: './assets/marker_stop.svg',
  iconSize: [12, 12],
  iconAnchor: [8, 8]
});

const map = L.map('map').setView(L.latLng(coordinates), 12);
const markersCanvas = new L.MarkersCanvas();
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

let markers = [];
for (const point of points) {
  markers.push(createMarker(point));
}

markersCanvas.addMarkers(markers);
markersCanvas.addTo(map);

function createMarker(point) {
  return new L.marker(L.latLng([point.y, point.x]), {
    icon: icon,
  });
}
```

Остановки в действии. Leaflet Edition

Инициализация карты

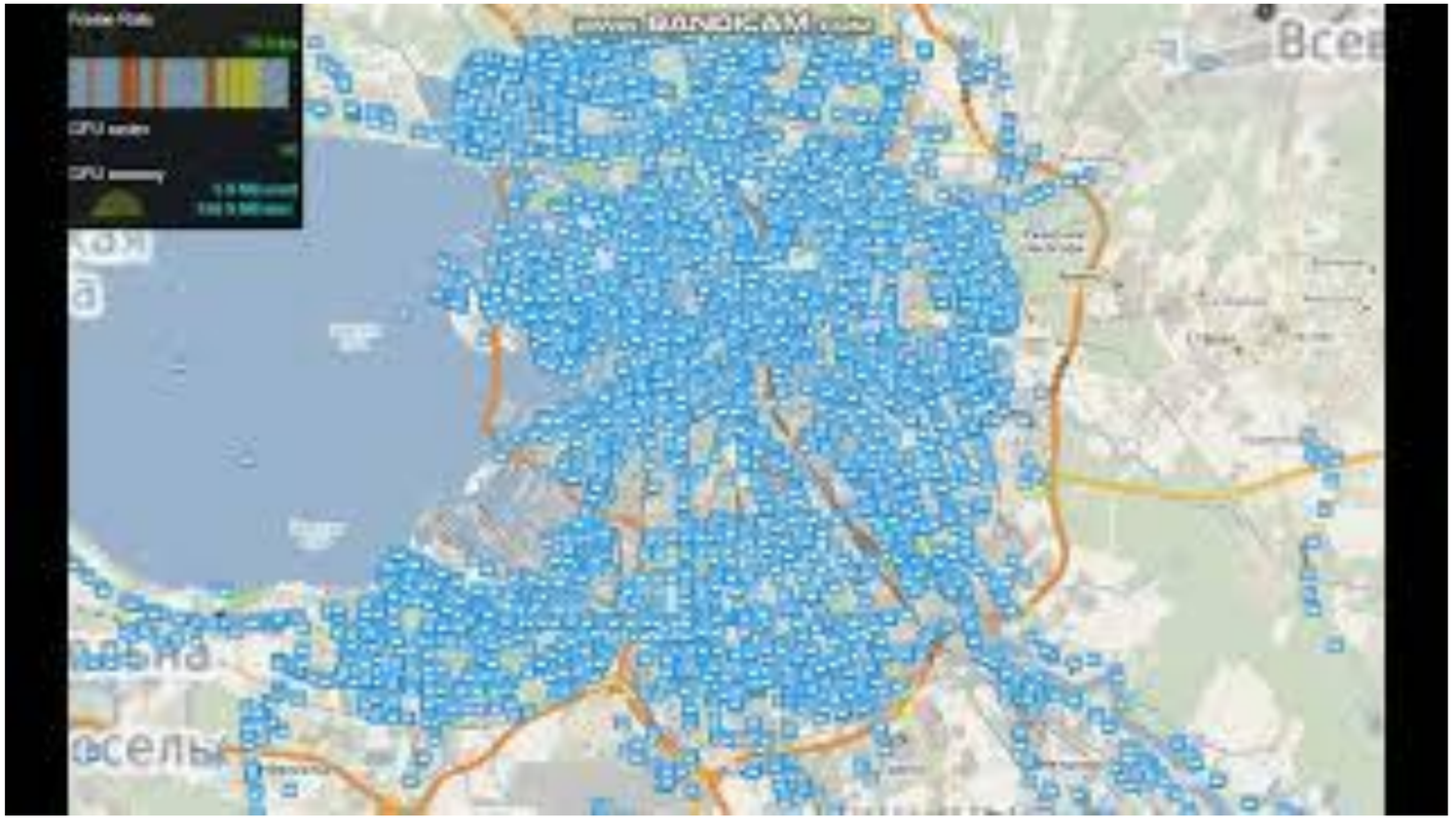
```
const icon = L.icon({
  iconUrl: './assets/marker_stop.svg',
  iconSize: [12, 12],
  iconAnchor: [8, 8]
});

const map = L.map('map').setView(L.latLng(coordinates), 12);
const markersCanvas = new L.MarkersCanvas();
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

let markers = [];
for (const point of points) {
  markers.push(createMarker(point));
}

markersCanvas.addMarkers(markers);
markersCanvas.addTo(map);

function createMarker(point) {
  return new L.marker(L.latLng([point.y, point.x]), {
    icon: icon,
  });
}
```

Остановки в действии. Leaflet Edition

Добавление маркеров

```
const icon = L.icon({
  iconUrl: './assets/marker_stop.svg',
  iconSize: [12, 12],
  iconAnchor: [8, 8]
});

const map = L.map('map').setView(L.latLng(coordinates), 12);
const markersCanvas = new L.MarkersCanvas();
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

let markers = [];
for (const point of points) {
  markers.push(createMarker(point));
}

markersCanvas.addMarkers(markers);
markersCanvas.addTo(map);

function createMarker(point) {
  return new L.marker(L.latLng([point.y, point.x]), {
    icon: icon,
  });
}
```

Остановки в действии. Leaflet Edition



Трек движения. Leaflet Edition

```
const map = L.map('map').setView(L.latLng(coordinates), 12);
const markersCanvas = new L.MarkersCanvas();
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

let markers = [];
for (const point of trackData.rawPoints) {
  markers.push(createMarker(point));
}
markersCanvas.addTo(map);
markersCanvas.addMarkers(markers);


const coords = tracks.flatMap(t => JSON.parse(t.track));
const polyline = new L.Polyline(coords, {
  color: '#8C8C8C',
  weight: 3,
  opacity: 1,
  smoothFactor: 1
}).addTo(map);
```


Трек движения. Leaflet Edition

Подготовка

```
const map = L.map('map').setView(L.latLng(coordinates), 12);
const markersCanvas = new L.MarkersCanvas();
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

let markers = [];
for (const point of trackData.rawPoints) {
  markers.push(createMarker(point));
}
markersCanvas.addTo(map);
markersCanvas.addMarkers(markers);


const coords = tracks.flatMap(t => JSON.parse(t.track));
const polyline = new L.Polyline(coords, {
  color: '#8C8C8C',
  weight: 3,
  opacity: 1,
  smoothFactor: 1
}).addTo(map);
```


Трек движения. Leaflet Edition

«Сырые» точки

```
const map = L.map('map').setView(L.latLng(coordinates), 12);
const markersCanvas = new L.MarkersCanvas();
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

let markers = [];
for (const point of trackData.rawPoints) {
  markers.push(createMarker(point));
}
markersCanvas.addTo(map);
markersCanvas.addMarkers(markers);

const coords = tracks.flatMap(t => JSON.parse(t.track));
const polyline = new L.Polyline(coords, {
  color: '#8C8C8C',
  weight: 3,
  opacity: 1,
  smoothFactor: 1
}).addTo(map);
```


Трек движения. Leaflet Edition

Трек движения

```
const map = L.map('map').setView(L.latLng(coordinates), 12);
const markersCanvas = new L.MarkersCanvas();
const wmsLayer =
  L.tileLayer.wms(url, { layers: 'default' }).addTo(map);

let markers = [];
for (const point of trackData.rawPoints) {
  markers.push(createMarker(point));
}
markersCanvas.addTo(map);
markersCanvas.addMarkers(markers);

const coords = tracks.flatMap(t => JSON.parse(t.track));
const polyline = new L.Polyline(coords, {
  color: '#8C8C8C',
  weight: 3,
  opacity: 1,
  smoothFactor: 1
}).addTo(map);
```


Трек движения. Leaflet Edition

Спасибо за внимание!

Вы супер!