



# Freedom Intelligent Routing Engine

---

Автоматическая AI-маршрутизация клиентских обращений

Go 1.21

Gemini AI

Django

Streamlit

# Решение: FIRE за 3 шага

**1** GENAI  
+  
GO

**1.1** .csv inputs

**1.2** results.csv

**2** DJANGO  
+  
POSTGRE

**3** STREAMLIT UI  
+  
Star Task  
AI Assistant

Результат: менеджер назначен ДО начала рабочего дня · Среднее время обработки < 4 сек

# Стек: надёжно и без лишнего

 **Go 1.21**

Бэкенд

1420 строк · параллельное геокодирование + кэш

 **Gemini 2.5 Flash**

AI-анализ

1 батч-запрос на 10 тикетов · < 60 сек на 31 тикет

 **Nominatim OSM**

Геокодирование

Haversine-расчёт до 15 офисов · rate limit 1 запрос/сек · LLM-fallback

 **Streamlit**

UI + AI-чат

Дашборд + AI-ассистент (Star Task) · фильтры · цветовая маркировка

 **Keyword Fallback**

Отказоустойчивость

100% покрытие без API · работает offline · детектор дублей

# Geo + AI

```
OfficeCoords = map[string]GeoPoint{
    "Алматы":      {43.2220, 76.8512},
    "Астана":      {51.1801, 71.4598},
    "Шымкент":     {42.3417, 69.5901},
    "Актобе":      {50.2839, 57.1670},
    "Атырау":      {47.1105, 51.9271},
    "Усть-Каменогорск": {49.9490, 82.6285},
    "Актау":       {43.6515, 51.1726},
    "Петропавловск": {54.8656, 69.1521},
    "Кокшетау":    {53.2849, 69.3966},
    "Павлодар":    {52.2873, 76.9674},
    "Тараз":       {42.9000, 71.3667},
    "Семей":       {50.4111, 80.2275},
    "Кызылорда":   {44.8488, 65.5091},
    "Уральск":     {51.2333, 51.3667},
    "Костанай":    {53.2141, 63.6324},
}

func resolveOfficeForTicket(t TicketInput, llmOffice string) (office string, lat, lon float64, method string) {
    isKZ := t.Country == "" ||
        strings.Contains(strings.ToLower(t.Country), "казахстан") ||
        strings.EqualFold(t.Country, "kz") ||
        strings.EqualFold(t.Country, "kazakhstan")

    if !isKZ {
        return "", 0, 0, "foreign"
    }

    // Пробуем Nominatim
    lat, lon, ok := geocodeAddress(t.Country, t.Oblast, t.RawCity, t.Street, t.House)
    if ok {
        fmt.Printf(" 🌐 Nominatim: %.4f, %.4f\n", lat, lon)
        nearestOffice := findNearestOfficeByCoords(lat, lon)
        if nearestOffice != "" {
            return nearestOffice, lat, lon, "nominatim"
        }
    }

    // Fallback: LLM-результат
    if llmOffice != "" {
        fmt.Printf(" 🤖 LLM-геолокация: офис '%s'\n", llmOffice)
        return llmOffice, 0, 0, "llm"
    }

    return "", 0, 0, "unknown"
}
```

<pre>func geocodeAddress(country, oblast, city, street, house string) (float64, float64, bool) {     // Составляем строку запроса из доступных полей     parts := []string{}     if house != "" &amp;&amp; street != "" {         parts = append(parts, house+" "+street)     } else if street != "" {         parts = append(parts, street)     }     if city != "" {         parts = append(parts, city)     } else if oblast != "" {         parts = append(parts, oblast)     }     if country != "" {         parts = append(parts, country)     }      if len(parts) == 0 {         return 0, 0, false     }      query := strings.Join(parts, ", ")     encoded := strings.ReplaceAll(query, " ", "+")     url := "https://nominatim.openstreetmap.org/search?q=" + encoded + "&amp;format=json&amp;limit=1"     response, err := http.Get(url)     if err != nil {         return 0, 0, false     }     defer response.Body.Close()     body, err := io.ReadAll(response.Body)     if err != nil {         return 0, 0, false     }     var results []map[string]string     err = json.Unmarshal(body, &amp;results)     if err != nil {         return 0, 0, false     }     if len(results) == 0 {         return 0, 0, false     }     lat, lon, address := results[0]["lat"], results[0]["lon"], results[0]["address"]     return lat, lon, address != "" }</pre>	<pre>prompt := fmt.Sprintf(`Ты – аналитик клиентских обращений Freedom Broker (Казахстан). Обработай массив тикетов и верни ТОЛЬКО JSON-массив без маркдауна, пояснений и текста вне массива.  ДОСТУПНЫЕ ОФИСЫ (nearest_office СТРОГО только из этого списка, любое другое значение – ОШИБКА): %s  [     {         "id": 1,         "office": "Алматы",         "lat": 43.2220,         "lon": 76.8512,         "method": "nominatim"     },     {         "id": 2,         "office": "Астана",         "lat": 51.1801,         "lon": 71.4598,         "method": "nominatim"     },     {         "id": 3,         "office": "Шымкент",         "lat": 42.3417,         "lon": 69.5901,         "method": "nominatim"     },     {         "id": 4,         "office": "Актобе",         "lat": 50.2839,         "lon": 57.1670,         "method": "nominatim"     },     {         "id": 5,         "office": "Атырау",         "lat": 47.1105,         "lon": 51.9271,         "method": "nominatim"     },     {         "id": 6,         "office": "Усть-Каменогорск",         "lat": 49.9490,         "lon": 82.6285,         "method": "nominatim"     },     {         "id": 7,         "office": "Актау",         "lat": 43.6515,         "lon": 51.1726,         "method": "nominatim"     },     {         "id": 8,         "office": "Петропавловск",         "lat": 54.8656,         "lon": 69.1521,         "method": "nominatim"     },     {         "id": 9,         "office": "Кокшетау",         "lat": 53.2849,         "lon": 69.3966,         "method": "nominatim"     },     {         "id": 10,         "office": "Павлодар",         "lat": 52.2873,         "lon": 76.9674,         "method": "nominatim"     },     {         "id": 11,         "office": "Тараз",         "lat": 42.9000,         "lon": 71.3667,         "method": "nominatim"     },     {         "id": 12,         "office": "Семей",         "lat": 50.4111,         "lon": 80.2275,         "method": "nominatim"     },     {         "id": 13,         "office": "Кызылорда",         "lat": 44.8488,         "lon": 65.5091,         "method": "nominatim"     },     {         "id": 14,         "office": "Уральск",         "lat": 51.2333,         "lon": 51.3667,         "method": "nominatim"     },     {         "id": 15,         "office": "Костанай",         "lat": 53.2141,         "lon": 63.6324,         "method": "nominatim"     },     {         "id": 16,         "office": "Астана (БЦ)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 17,         "office": "Астана (ЮЗ)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 18,         "office": "Астана (СЗ)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 19,         "office": "Астана (ЮВ)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 20,         "office": "Астана (СВ)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 21,         "office": "Астана (Ц)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 22,         "office": "Астана (З)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 23,         "office": "Астана (В)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 24,         "office": "Астана (С)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 25,         "office": "Астана (Ю)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 26,         "office": "Астана (Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 27,         "office": "Астана (Юг)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 28,         "office": "Астана (Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 29,         "office": "Астана (Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 30,         "office": "Астана (Центр)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 31,         "office": "Астана (Север-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 32,         "office": "Астана (Юго-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 33,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 34,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 35,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 36,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 37,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 38,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 39,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 40,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 41,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 42,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 43,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 44,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 45,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 46,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 47,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 48,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 49,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 50,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 51,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 52,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 53,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 54,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 55,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 56,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 57,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 58,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 59,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 60,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 61,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 62,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 63,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 64,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 65,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 66,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 67,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 68,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 69,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 70,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 71,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 72,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 73,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 74,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 75,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 76,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 77,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 78,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 79,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 80,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 81,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 82,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 83,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 84,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 85,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 86,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 87,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 88,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 89,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 90,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 91,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 92,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 93,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 94,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 95,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 96,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 97,         "office": "Астана (Юго-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 98,         "office": "Астана (Северо-Запад)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 99,         "office": "Астана (Северо-Восток)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     },     {         "id": 100,         "office": "Астана (Юго-Север)",         "lat": 51.1801,         "lon": 71.4598,         "method": "llm"     } ]</pre>
<b>ПРАВИЛА КЛАССИФИКАЦИИ ТИПА ОБРАЩЕНИЯ:</b>	
<b>"Жалоба"</b> <ul style="list-style-type: none"><li>– клиент недоволен качеством сервиса, обслуживанием, сроками, но НЕ требует денег и НЕ угрожает судом</li><li>– примеры: "недоволен работой", "ужасный сервис", "уже неделю не решают", "заблокировали без причины"</li><li>– ОТЛИЧИЕ от Претензии: нет требования денег, нет угрозы судом/прокуратурой</li></ul>	
<b>"Смена данных"</b> <ul style="list-style-type: none"><li>– клиент хочет изменить персональные данные: номер телефона, email, паспорт, адрес, ИИН</li><li>– примеры: "хочу изменить номер", "смена телефона", "обновить паспортные данные", "удалить мои данные"</li></ul>	
<b>"Консультация"</b> <ul style="list-style-type: none"><li>– клиент задаёт вопрос, хочет получить информацию, уточнить условия – НЕТ технической проблемы</li><li>– примеры: "как купить акции", "какие комиссии", "можно ли дробно", "что такое ETF", "ИИН компании"</li><li>– ОТЛИЧИЕ от Неработоспособности: клиент может пользоваться системой, просто задаёт вопрос</li></ul>	
<b>"Претензия"</b> <ul style="list-style-type: none"><li>– клиент требует возврата денег/компенсации ИЛИ угрожает судом/прокуратурой/журналистами</li><li>– примеры: "верните деньги", "подам в суд", "обращусь в прокуратуру", "125\$ не пришло верните"</li><li>– ОТЛИЧИЕ от Жалобы: есть конкретное денежное требование ИЛИ угроза правовыми действиями</li></ul>	
<b>"Неработоспособность приложения"</b> <ul style="list-style-type: none"><li>– технические проблемы мешают клиенту ИСПОЛЬЗОВАТЬ сервис: не входит, не приходит SMS, ошибка</li><li>– примеры: "не могу войти", "пароль не принимает", "смс не приходит", "ошибка при входе", "не могу зарегистрироваться"</li><li>– ОТЛИЧИЕ от Консультации: клиент ПЫТАЕТСЯ что-то сделать, но система не даёт</li></ul>	
<b>"Мошеннические действия"</b> <ul style="list-style-type: none"><li>– клиент подозревает мошенничество, несанкционированный доступ, просит проверить легитимность</li><li>– примеры: "не мошенники ли вы", "взломали аккаунт", "Money Advisor мошенники?", "проверьте сертификат"</li></ul>	
<b>"Спам"</b> <ul style="list-style-type: none"><li>– рекламные рассылки, предложения от третьих компаний, не связанные с Freedom Broker</li><li>– примеры: тюльпаны, сварочные агрегаты, ПЕРВОУРАЛЬСКБАНК, ссылки <a href="https://safelinks.protection.outlook.com">safelinks.protection.outlook.com</a></li></ul>	
<b>ПРАВИЛА ТОНАЛЬНОСТИ:</b>	
<b>"Негативный"</b> – явное недовольство, угрозы, требования, обвинения, срочность с давлением	
<b>"Позитивный"</b> – благодарность, похвала, удовлетворённость	
<b>"Нейтральный"</b> – нейтральный вопрос, запрос информации, техническая проблема без эмоций, спам	
<b>ПРАВИЛА ПРИОРИТЕТА (1–10):</b>	



# Fallback

```
r := AIResult{
    Type:      "Консультация",
    Sentiment: "Нейтральный",
    Language:  "RU",
    Priority:   "5",
    Summary:   "Keyword-анализ. Требуется проверка менеджером.",
    NearestOffice: "",
    Source:    "Fallback",
}

// — Определение языка —
kazWords := []string{"сіз", "өтінемін", "қате", "көмек", "рахмет", "жоқ", "болады",
    "саламатсыздарма", "менде", "бұйрық", "неге", "алуға"}
engWords := []string{"please", "help", "error", "account", "transfer", "unable",
    "issue", "hello", "dear", "regards", "blocked", "verify", "validation"}

kazCount, engCount := 0, 0
for _, w := range kazWords {
    if strings.Contains(lower, w) {
        kazCount++
    }
}
for _, w := range engWords {
    if strings.Contains(lower, w) {
        engCount++
    }
}
if kazCount >= 2 {
    r.Language = "KZ"
} else if engCount >= 2 {
    r.Language = "ENG"
}
```

```
// — Классификация по ключевым словам —
switch {
case containsAny(text, "суд", "прокуратура", "адвокат", "иск", "court", "lawyer",
    "правоохранительные органы", "заявление в", "следственный"):
    r.Type = "Претензия"
    r.Sentiment = "Негативный"
    r.Priority = "10"
    r.Summary = "Клиент угрожает обращением в правоохранительные органы или суд. Немедленная эскалация Главному сп"

case containsAny(text, "мошенник", "украли", "взлом", "несанкционированн", "fraud",
    "scam", "мошеннические", "финансовые махинации"):
    r.Type = "Мошеннические действия"
    r.Sentiment = "Негативный"
    r.Priority = "9"
    r.Summary = "Подозрение на мошенничество или несанкционированные действия. Срочно в отдел безопасности."

case containsAny(text, "верните", "возврат", "компенсация", "возместите", "refund",
    "не пришло", "не на моем счету", "списали"):
    r.Type = "Претензия"
    r.Sentiment = "Негативный"
    r.Priority = "8"
    r.Summary = "Требование возврата средств. Запросить детали транзакции и подтверждающие документы."

case containsAny(text, "смена номера", "изменить данные", "паспорт", "реквизиты",
    "смена данных", "изменить номер", "персональные данные", "удалить мои данные"):
    r.Type = "Смена данных"
    r.Sentiment = "Нейтральный"
    r.Priority = "6"
    r.Summary = "Запрос на изменение персональных данных. Запросить документы для верификации."

case containsAny(text, "не могу войти", "не работает", "вылетает", "зависает",
    "ошибка", "crash", "error", "blocked", "заблокирован", "блокирован",
    "пароль не принимает", "смс не приходит", "код не приходит"):
    r.Type = "Неработоспособность приложения"
    r.Sentiment = "Негативный"
    r.Priority = "6"
    r.Summary = "Технический сбой при входе или работе с приложением. Запросить ОС, версию приложения и скриншоты."
```

⚙️ Управление

Запускает Go-движок: анализ  
тикетов через Gemini + роутинг.

▶ Запустить анализ

🔄 Обновить данные

🗄️ БД: `fire_db` на `localhost:5433`

📊 Тикетов в БД: 155

🔥

# FIRE — Freedom Intelligent Routing Engine

Система автоматического распределения обращений клиентов | Freedom Broker

📊

## Оперативная сводка

Всего тикетов

155

VIP + Priority

59

🚫 Спам

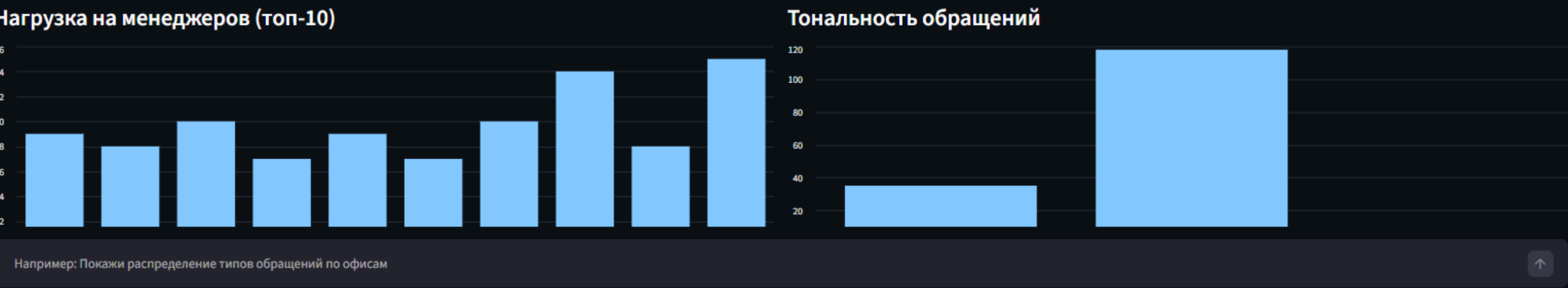
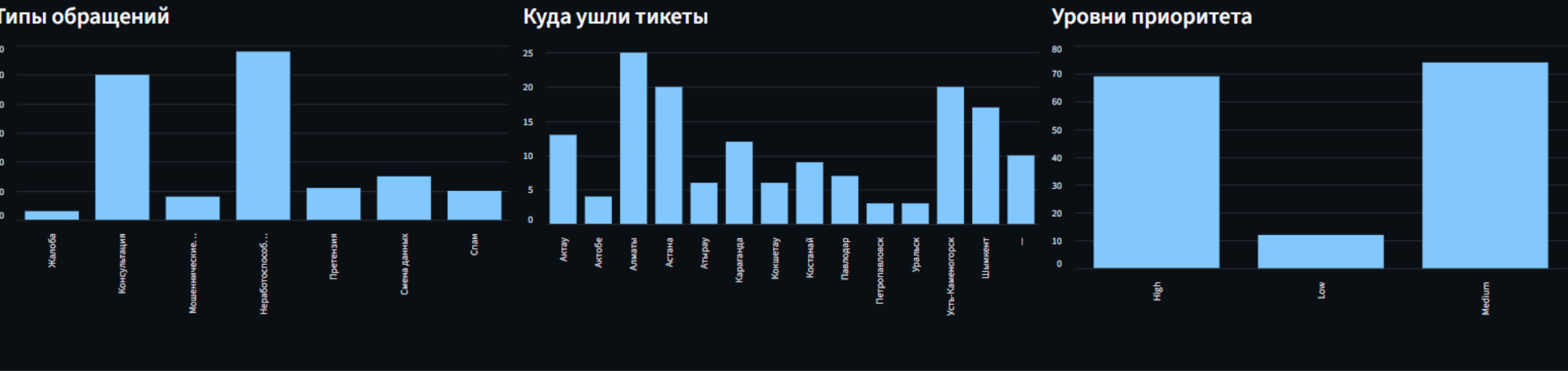
10

👤 Претензии / Фрод

19

🚩 Эскалировано в ГО

4



⚙️ Управление

Запускает Go-движок: анализ тикетов через Gemini + роутинг.

▶ Запустить анализ

🔄 Обновить данные

📁 БД: `fire_db` на `localhost:5433`

📄 Тикетов в БД: 155

📄 Детализация распределения

🔖 Тип обращения

Choose options

🔥 Приоритет

Choose options

👤 Сегмент

Choose options

🏢 Офис

Choose options

	Сегмент	Тип	Тональность	Язык	Приоритет	Приоритет_уровень	Рекомендации менеджеру	Назначенный Менеджер	Должность	Офис Назначения	Эскалирован
64	Mass	Консультация	Нейтральный	RU	5	Medium	Клиент запрашивает выписку по брокерскому счету за период с 01.01.2025 по 01.0	Менеджер 21	Специалист	Усть-Каменогорск	Нет
65	VIP	Неработоспособность приложения	Негативный	RU	10	High	Клиент не может войти в аккаунт, так как система не принимает пароль, а коды д	Менеджер 18	Главный специалист	Алматы	Нет
66	Mass	Спам	Нейтральный	KZ	1	Low	Бұл Freedom Broker-ге IQAS интеллектуалдық лигасын ашуға және квиз кештерін ү	—	—	—	Нет
67	Mass	Неработоспособность приложения	Негативный	KZ	6	Medium	Клиент жеке куәлігін PDF форматында жібергенде верификациядан өтпей жатқа	Менеджер 8	Ведущий специалист	Усть-Каменогорск	Нет
68	VIP	Неработоспособность приложения	Негативный	RU	10	High	Клиент не может войти в аккаунт Tradernet, система выдает ошибку неверного па	Менеджер 31	Главный специалист	Костанай	Нет
69	Priority	Консультация	Нейтральный	RU	10	High	Клиент просит завершить все деловые отношения с Freedom Broker и предостави	Менеджер 44	Главный специалист	Шымкент	Нет
70	Mass	Неработоспособность приложения	Негативный	RU	6	Medium	Клиент не может зарегистрироваться в приложении Freedom Broker, получая оши	Менеджер 17	Ведущий специалист	Караганда	Нет
71	Mass	Мошеннические действия	Негативный	RU	9	High	Клиент получает множество SMS-кодов подтверждения и подозревает взлом акка	Менеджер 24	Ведущий специалист	Алматы	Нет
72	Mass	Неработоспособность приложения	Нейтральный	KZ	6	Medium	Клиенттің жеке куәлігі қабылданбай жатыр. Техникалық мәселені тексеріп, клиен	Менеджер 39	Ведущий специалист	Астана	Нет
73	VIP	Консультация	Нейтральный	RU	10	High	Клиент интересуется процедурой перевода средств с карты Freedom Broker на ка	Менеджер 42	Главный специалист	Усть-Каменогорск	Нет
74	Priority	Консультация	Нейтральный	RU	10	High	Клиент интересуется сроками закрытия счета после подачи поручения, так как сч	Менеджер 19	Ведущий специалист	Кокшетау	Нет
75	VIP	Неработоспособность приложения	Нейтральный	RU	10	High	Клиент не может перевести деньги между своими счетами, так как приложение и	Менеджер 22	Главный специалист	Астана	Нет

Показано 155 из 155 тикетов

> 📄 Эскалированные тикеты (4 шт) — нажмите для просмотра

📎 Вложения к тикетам

Выберите тикет с вложением:

7 — irrelevant.png



Например: Покажи распределение типов обращений по офисам

↑



Управление

Запускает Go-движок: анализ  
тикетов через Gemini + роутинг.

▶ Запустить анализ

Обновить данные

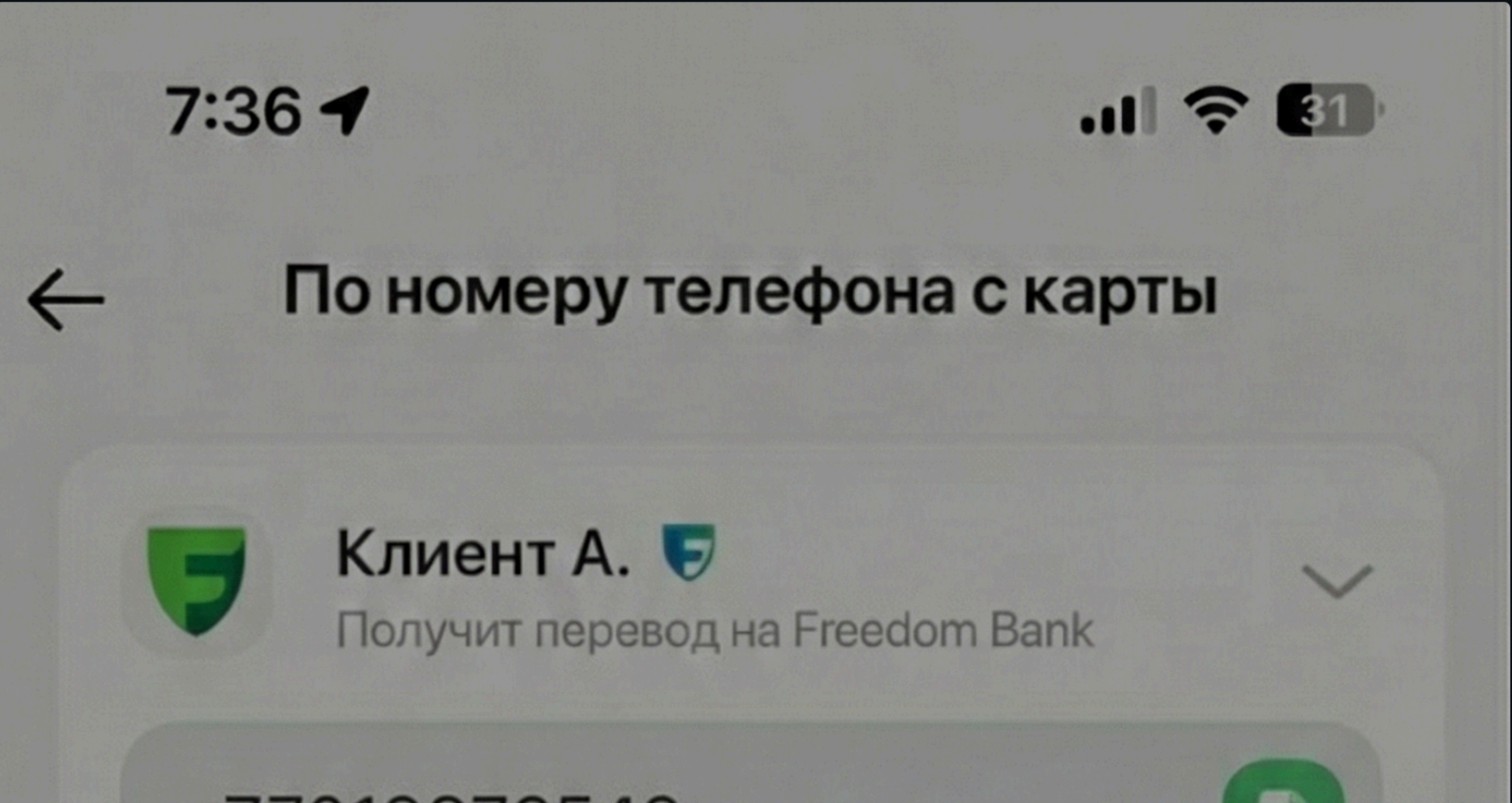
БД: fire\_db на localhost:5433

Тикетов в БД: 155

Вложения к тикетам

Выберите тикет с вложением:

7 — irrelevant.png



Например: Покажи распределение типов обращений по офисам





⚙️ Управление

Запускает Go-движок: анализ тикетов через Gemini + роутинг.

▶ Запустить анализ

🔄 Обновить данные

🗄️ БД: fire\_db на localhost:5433

📊 Тикетов в БД: 155

Май

Неф

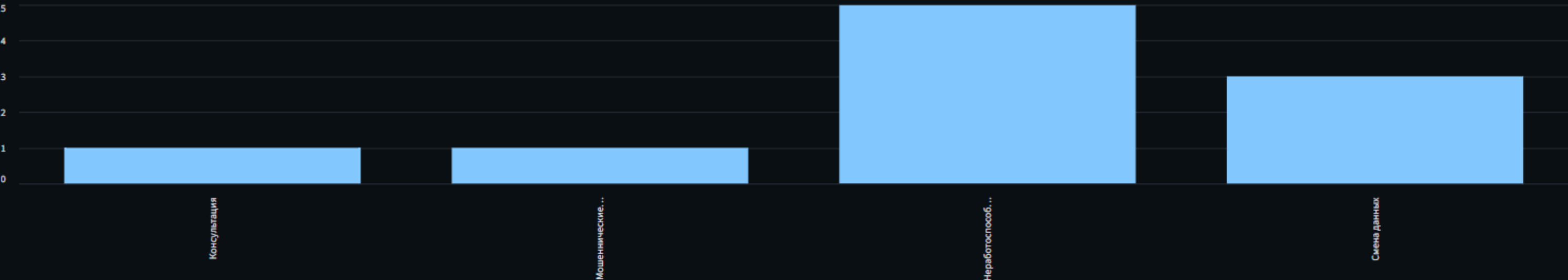


Нарисуй 2д график распределения типов обращений в 3 городах, офисы которых имеют наименьшее количество тикетов



График покажет распределение типов обращений для офисов Петропавловска, Уральска и Актобе, которые имеют наименьшее количество тикетов.

Распределение типов обращений в Петропавловске, Уральске и Актобе



💡 Примеры вопросов к ассистенту

Графики и диаграммы:

- Покажи распределение типов обращений по офисам
- Построй график нагрузки на менеджеров
- Покажи тональность обращений по сегментам
- Построй диаграмму языков обращений
- Покажи топ-5 самых загруженных офисов

Аналитика:

- Сколько VIP-клиентов было эскалировано в главный офис?
- Какой процент тикетов получил приоритет High?
- Какой менеджер ведёт больше всего тикетов?
- Сколько тикетов помечены как спам и в каких офисах они сконцентрированы?
- Есть ли связь между сегментом клиента и тональностью обращения?
- Какие типы обращений чаще всего эскалируются?
- Покажи статистику по эскалированным тикетам: по сегменту и типу

Например: Покажи распределение типов обращений по офисам



# Road to Production

## Datasaur'26 MVP → Enterprise Production

- CSV & Batch → Kafka Message Broker (ACID-транзакционность)
- In-Memory Script → Event-Driven Daemon + Redis
- One-way Routing → Closed-Loop Feedback

# FIRE MVP ГОТОВ

- ✓ Полное соответствие ТЗ: все фильтры, Round Robin, эскалация
- ✓ AI на трёх языках: рус / каз / eng · summary на языке клиента
- ✓ Геолокация по 15 офисам · Nominatim + Haversine + LLM-fallback
- ✓ Дашборд + AI-ассистент · Star Task выполнен
- ✓ Keyword Fallback · 100% покрытие без API · детектор дублей



# Ждем ваши вопросы!

Github  
repo:



TG:  
@askhn  
@asssstnuar