**Вариант 3.4. Логистическая компания**

**Шаг 1: Определение требований**

* 1. **Объем данных**
     1. Ожидаемый объем 50 ТБ в год
     2. Ожидаемый рост: 25% ежегодно
  2. **Скорость получения данных:**

до 500 событий в секунду.

* 1. **Типы данных:**
     1. Структурированные: о заказах и накладных, о клиентах и контрагентах, о транспорте и ресурсах, транзакции, справочники. – 70%
     2. Полуструктурированные: данные от картографических сервисов, лог-файлы, данные от водителей и курьеров. – 20%
     3. Неструктурированные: сканы документов, фото груза, видео с камер наблюдения. – 10%
  2. **Требования к обработке:** 
     1. Оптимизация маршрутов в реальном времени.
     2. Прогнозирование спроса на услуги.
  3. **Доступность данных:** 
     1. Время отклика: <30 секунд.
     2. Доступность системы: 99.9%,
  4. **Безопасность данных:** 
     1. Базовое шифрование.
     2. Соответствие 152-ФЗ.

**Шаг 2: Выбор модели хранилища данных:**

Выбор схемы «Звезда», т. к. имеет простую структуру. Анализ в логистике часто требует срезов по времени (день/неделя/месяц), географии (город/регион/страна), типу груза, клиенту, транспорту, водителю. Денормализованные измерения в "Звезде" содержат все необходимые для этих срезов атрибуты в одной таблице, что ускоряет фильтрацию и группировку. Прогнозирование спроса и анализ эффективности маршрутов основаны на агрегированных показателях (количество заказов, сумма, вес, время доставки, пробег), которые эффективно извлекаются из таблицы фактов с минимумом JOIN'ов к измерениям.

**2.1. Источники данных**

⎯ Система управления перевозками (TMS).

⎯ Система управления складом (WMS).

⎯ Финансовые системы (ERP).

⎯ Устройства IoT.

⎯ Внешние сервисы (карты, погода, таможенные системы).

⎯ Камеры видеонаблюдения.

**2.2. Слой сбора данных**

⎯ Apache Kafka – для сбора потоковых данных, горизонтальное масштабирование.

⎯ Fluentd – для сбора логов (обработка десятков тысяч событий в секунду, имеет плагин kafka2 для прямой отправки в Kafka, шифрование, открытое решение).

**2.3. Слой хранения данных**

⎯ PostgreSQL для структурированных данных. — надежная реляционная СУБД, для транзакционных данных и справочников

⎯ ClickHouse для аналитических запросов с высокой скоростью, хранения логов.

⎯ MinIO для неструктурированных, легко масштабируется (сканы документов, фото грузов, видео с камер).

⎯ MongoDB – для полуструктурированных данных, лучше PostgreSQL работает с геопространственными запросами, лог-файлы, комментарии водителей, гибкая схема документов.

⎯ Redis – кэширование, хранения актуального состояния (екущие позиции транспорта, активные заказы, кэш оптимальных маршрутов для участков).

**2.4. Слой обработки данных**

⎯ Apache Flink – для распределенной потоковой обработки,

⎯ Apache Spark – для обработки в пакетном режиме (сложные преобразования, ML).

⎯ OpenCV (анализ фото до/после загрузки, анализ видео, контроль погрузки/разгрузки).

**2.5. Слой аналитики и машинного обучения**

⎯ Jupiter Notebook – для аналитик и ML. Стандарт для аналитиков и дата-сайентистов, поддерживает Python, ML-библиотеки (scikit-learn, TensorFlow, PyTorch).

⎯ Яндекс Даталенс – для визуализации и дашбордов. Бизнес-аналитика.

⎯ Grafana. Визуализация метрик из Prometheus (и других источников) в графиках/дашбордах. Мониторинг инфраструктуры / performance / SLA.

**2.6. Слой управления данными**

⎯ Arenadata Platform Security – для централизованного управления политиками безопасности (авторизация, маскировка данных, аудит).

**2.7. Слой оркестрации и мониторинга**

⎯ Apache Airflow – для оркестрации, объединения данных из разных СУБД.

⎯ Prometheus – для мониторинга и алертинга (система сбора метрик).

**Безопасность**

1. Шифрование: упомянуто на всех уровнях.
2. Авторизация.
3. Маскирование/Анонимизация: при загрузке в аналитическое хранилище (ClickHouse) чувствительные данные (ПДн из 152-ФЗ: ФИО клиентов/водителей, паспорта, номера телефонов) должны маскироваться или токенизироваться в ETL/ELT (Spark/Flink).
4. Аудит.

**Отказоустойчивость:**

Tier II ЦОД: репликация (синхронная/асинхронная) в СУБД (PostgreSQL, ClickHouse, MongoDB) и объектном хранилище (MinIO/S3) гарантирует сохранность и доступность при отказе узла/AZ, Kafka с реплицированными топиками, балансировщики нагрузки.

**Потянет рост данных**

Год 1: 50 ТБ

Год 2: 62.5 ТБ (+25%)

Год 3: 77.6 ТБ (+25%)

PostgreSQL : шардинг (Citus/Patroni) легко поддерживает до 100+ ТБ. Рекомендация: автоматический партиционирование по времени (например, помесячно).

ClickHouse: обрабатывает петабайты. TTL-правила автоматически удалят устаревшие данные.

MinIO: масштабируется до эксабайтов.

Kafka: 1 брокер выдерживает 10–15K событий/сек. Кластер из 3 нод с репликацией — избыточен.

Flink: обрабатывает 500 событий/сек на 2–4 воркерах. Автомасштабирование в Kubernetes нивелирует пики.

Redis: кэширует ключевые данные (позиции ТС, маршруты). Объём оперативки: ≤128 ГБ даже для 100K активных заказов.

**Выводы:**

1. Объем данных (0 ТБ/год, рост 25% годовых): MinIO: оптимален для хранения и масштабирования под растущие объемы неструктурированных данных (сканы, фото, видео - 10%). Шардинг PostgreSQL: ключевая стратегия для горизонтального масштабирования структурированных данных (70%) и справочников по мере роста. Автомасштабирование: необходимо для Kafka (слой приема), Flink/Spark (слой обработки), MinIO и виртуальных машин/контейнеров в облаке/на платформе для адаптации к растущей нагрузке. ClickHouse/Arenadata DB: специализированные аналитические СУБД обеспечивают эффективное хранение и сверхбыстрое выполнение запросов к большим историческим данным (логи, агрегированные факты).
2. Скорость (до 500 событий/сек): Apache Kafka: легко покрывает требуемую скорость приема (500 событий/сек) с большим запасом (способен обрабатывать десятки/сотни тысяч сообщ./сек). Обеспечивает гарантированную доставку и буферизацию для потоковых процессоров. Apache Flink: обеспечивает обработку событий с низкой задержкой (миллисекунды - единицы секунд), критичную для оптимизации маршрутов в реальном времени.
3. Оптимизация маршрутов (РВ): Apache Flink + Kafka + Redis идеальны для обработки потоковых данных (GPS, трафик, новые заказы) и расчета/обновления оптимальных маршрутов с минимальной задержкой. Redis хранит актуальное состояние (позиции, заказы, кэш маршрутов).
4. Прогнозирование спроса (Оценка объемов): ClickHouse + Apache Spark + Jupyter Notebook обеспечивают мощную платформу для хранения исторических данных, выполнения сложных пакетных преобразований, построения и обучения ML-моделей прогнозирования. Поддержка Python и ML-библиотек в Jupyter - стандарт для Data Science.
5. Доступность данных (<30с отклик, 99.9%): PostgreSQL (с индексами, кэшем) + Redis (кэш) обеспечивают быстрый доступ к текущим транзакциям и справочникам. ClickHouse гарантирует высокую скорость аналитических запросов даже на больших данных.
6. Безопасность данных (Шифрование, 152-ФЗ): шифрование применяется на всех уровнях (передача TLS/SSL, хранение). Маскирование/Токенизация чувствительных данных (ФИО, паспорта, телефоны) реализуется в ETL/ELT (Spark/Flink) при загрузке в аналитические хранилища (ClickHouse). Arenadata Platform Security для централизованного управления авторизацией и ролями.