**Телекоммуникационная компания**

1. **Определение требований**
   1. Объем данных: 1 ПБ в год, рост 40% ежегодно
   2. Скорость получения: до 10000 событий в секунду
   3. Типы данных: 40% структурированные, 50% полуструктурированные, 10% неструктурированные
   4. Требования к обработке: анализ качества связи, прогнозирование нагрузки на сеть.
   5. Доступность: 99.99%, время отклика <10 секунд
   6. Безопасность:

- шифрование;

- соответствие 152-ФЗ и отраслевым стандартам.

**2. Выбор модели хранилища данных**

Для структурированных данных будет использоваться объектно-реляционная база данных - Microsoft SQL Server.

Для полуструктурированных будет использоваться графовую базу данных - Neo4j, так как она эффективна для анализа отношений данных, исходя из чего можно получить ценную информацию.

Для неструктурированных будет использоваться хранилище данных в виде ключ-значение - Redis

Для телекоммуникационной компании с 1 ПБ данных в год (рост 40% ежегодно), высокой скоростью поступления (10 000 событий/сек) и требованиями к аналитике в реальном времени (<10 сек) оптимальной моделью хранения будет "Звезда", потому что:

* Высокая скорость запросов:
  + В телекоммуникационных компаниях критично время отклика <10 сек (анализ QoS, прогнозирование нагрузки).
  + Star Schema минимизирует JOIN за счет денормализации (факты + измерения в одной таблице).
* Оптимизация для потоковой обработки:
  + Данные CDR и метрик удобно хранить в виде фактовых таблиц (например, fact\_calls с полями call\_duration, signal\_strength).
  + Измерения (dimensions) – dim\_time, dim\_customer, dim\_location – связаны напрямую.
* Меньше сложностей с ETL:

Прием 10 000 событий/сек требует быстрой загрузки. В Snowflake пришлось бы разбивать данные на множество нормализованных таблиц.

* Масштабируемость:

Совместима с Apache Druid / ClickHouse (OLAP для телекома), где Star Schema работает эффективнее.

На рисунке 1 изображена модель хранения «Звезда»

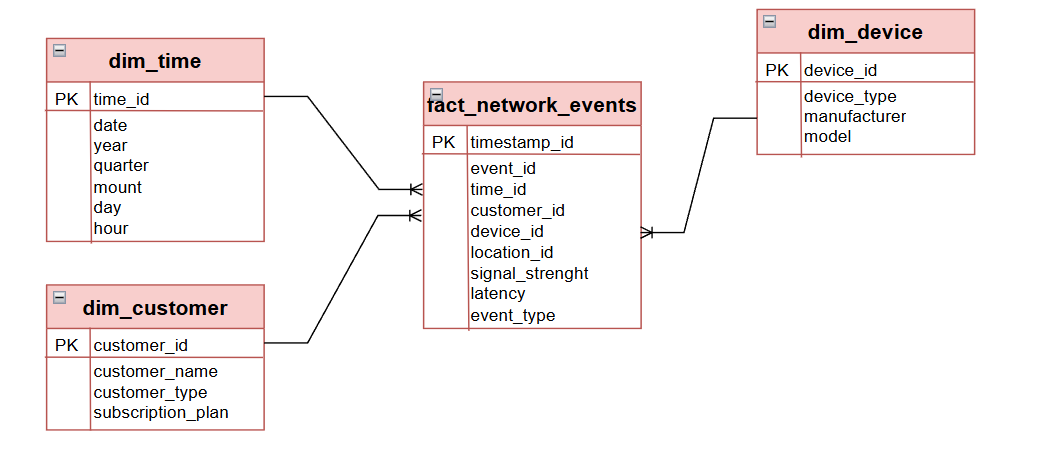


Рисунок 1. Модель хранения «Звезда»

**3. Архитектура хранилища больших данных**

Слой источника данных:

* CDR (Call Detail Records) – данные о звонках и сессиях
* Пользовательские метаданные (тарифы, абонентские данные)
* Социальные сети и отзывы (неструктурированные данные)

Слой сбора данных:

* Logstash для преобразования данных в желаемый формат (горизонтально масштабируемый конвейер обработки данных с поддержкой Elasticsearch и Kibana)
* Apache Kafka система обработки потоков данных для обработки высокоскоростных потоков данных.

Слой хранения данных

* NoSQL-база данных MongoDB, для хранения полуструктурированных данных, таких как данные о звонках и сессиях
* Реляционная база данных PostgreSQL, для хранения структурированных данных, таких как пользовательские метаданные
* Объектное хранилище HDFS (Hadoop Distributed File System) — это распределённая файловая система, предназначенная для хранения больших массивов данных в рамках кластера из нескольких узлов.

Слой обработки данных

* Apache Spark для обработки и анализа данных в реальном времени.
* Apache Flink для обработки высокоскоростных потоков данных.

Слой аналитики и машинного обучения

* Python для анализа данных и построения моделей в Jupyter Notebook.
* Power BI для представления результатов анализа.

Слой управления данными

* Apache Atlas для управления метаданными и обеспечения их качества.

Слой оркестрации и мониторинга

* Prometheus для обеспечения мониторинга и оповещения о проблемах.
* Grafana для представления результатов мониторинга.

На рисунке 2 изображена архитектура

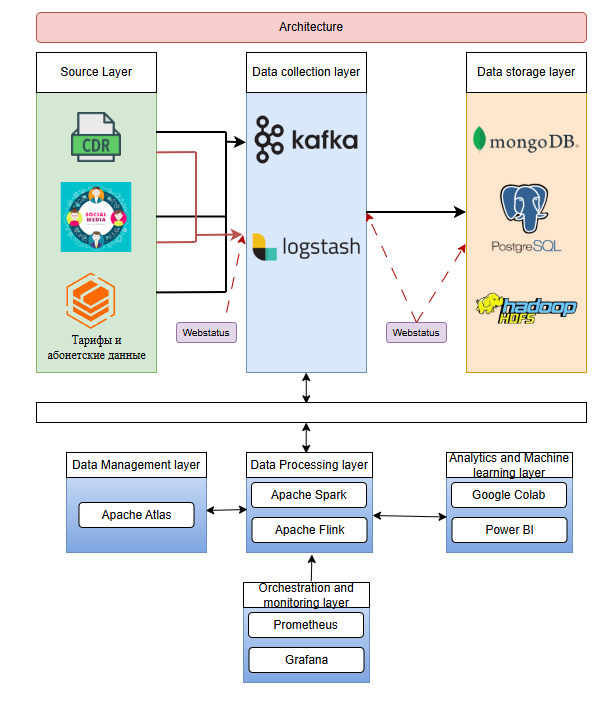


Рисунок 2. Архитектура

**4. Процесс обработки данных**

− Данные собираются из различных источников через слой сбора данных.

− Сырые данные сохраняются в HDFS для долгосрочного хранения.

− Потоковые данные обрабатываются в реальном времени с помощью Apache Spark для быстрой аналитики.

− Аналитики используют Jupyter Notebooks и Power BI для исследования данных и создания отчетов.

**5. Масштабирование и отказоустойчивость**

* Использование HDFS для легкого масштабирования по горизонтали. Если объём данных или нагрузка увеличиваются, то можно просто добавить больше серверов в вычислительный кластер.
* Репликация данных в HDFS и PostgreSQL для обеспечения отказоустойчивости. Каждый блок данных в HDFS дублируется на несколько узлов. Если один узел выходит из строя, информация может быть восстановлена из других.
* Применение Prometheus для обеспечения мониторинга и оповещения о проблемах и Grafana для представления результатов мониторинга.

**6. Безопасность**

1. Двухфакторная аутентификация для доступа к данным.
2. Регулярное резервное копирование и план аварийного восстановления

**Выводы:**

Согласно изначальным требованиям телекоммуникационной компании, были выполнены все поставленные задачи, и в качестве результата получена масштабируемая и отказоустойчивая архитектура хранения данных, которая позволяет обработать большие объемы данных в реальном времени, что необходимо для мониторинга телекоммуникаций

Не смотря на гарантированную надежность благодаря транзитивности схемы «Снежинка», выбор был сделан в сторону модели «Звезда», так как она больше подходит для быстрого выполнения запросов.