Company Profile

“Medical Data Processing” Company is based out of San Francisco, a city in California of the United States of America. The company was founded in 2007 and has grown significantly since. Company specializes in processing various types of EMR (Electronic Medical Records) and provides real-time insights to various medical facilities. The customers of Medical Data Processing companies run multiple medical facilities providing various kinds of patient care such as Urgent Care (UC), hospitals, nursing homes, emergency rooms, critical care units etc. The company specializes in providing data insight solutions. These data insights are used by their customers to stay compliant with laws, track patient health metrics, admit discharge records, bed availability. The company has about 1100 customers and 370 employees. Their solution is used by about 8000 individual medical care facilities.

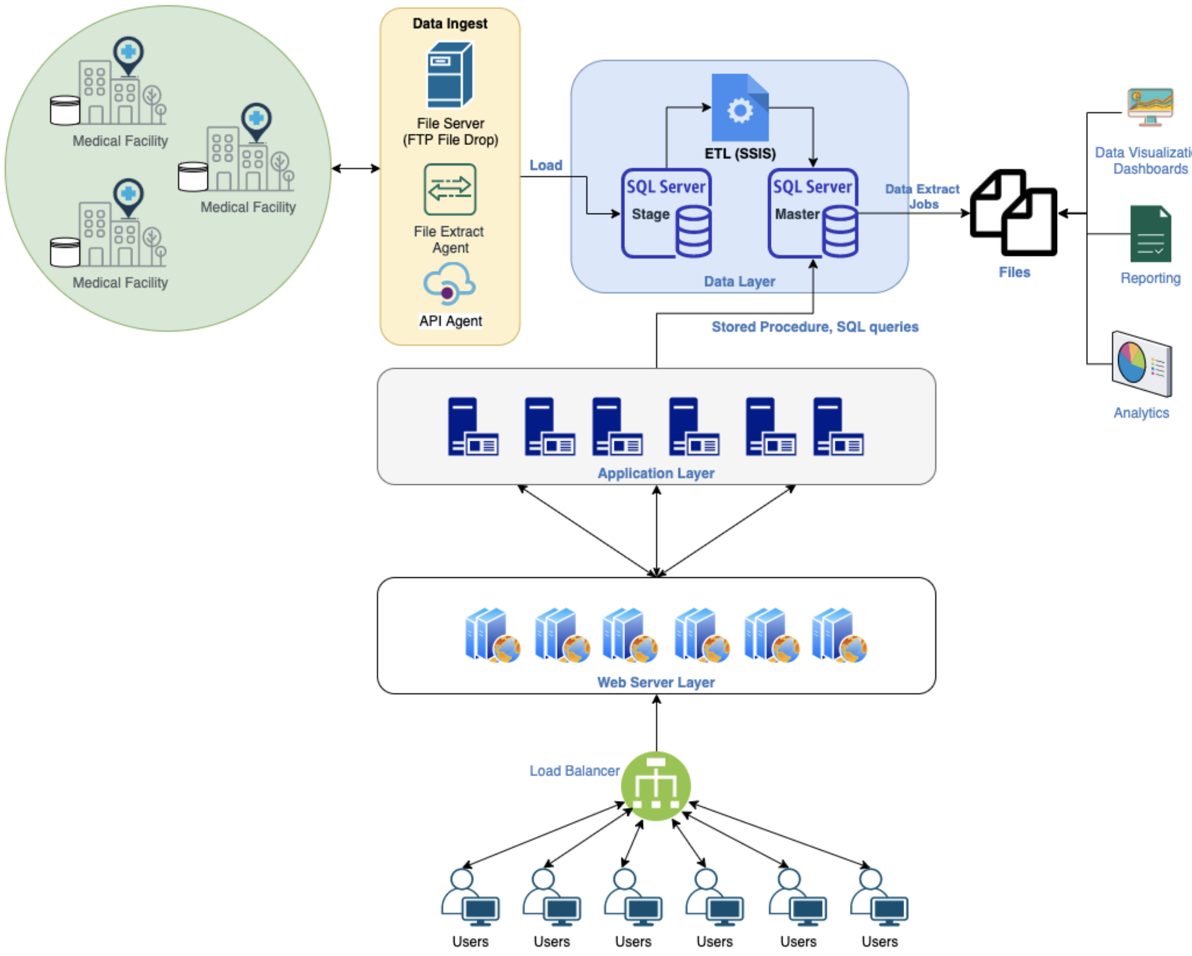
Current Architecture

Since the beginning of the company in 2008, Medical Data Processing leveraged traditional monolithic 3-tier application architecture backed by proprietary SQL based databases and warehouses. The following picture shows a high-level diagram of their existing architecture. Today, Medical Data Processing Systems stores all of their existing data in SQL Server.

Over 8000 individual facilities produce the data in different data formats such as XML, TXT, CSV. Today, the data is retrieved using multiple techniques such as calling customer API, pulling the data from customer’s FTP servers. Company also hosts its own FTP server where customers can push data as shown in the “Data Ingest” box in the diagram below. The incoming data arrives in a staging area. All files are compressed using “gzip” format and password protected.

Once the data is available in the staging area (file system on a server), this data is loaded into a “stage” SQL server database instance. From there, proprietary technologies and scripts (such as SSIS) are used to perform the data transformation, cleaning, deduplication. This transformed data is stored in a “master” SQL server database server. Once the data is loaded into the master, existing data from the stage server is purged.

A web-based portal is designed for customers. Customers can log into these web servers and they can access the data insights, visualize the data, generate custom reports. All of the requests from users are processed by application servers. Application servers host a set of APIs which in turn query the master database. For repetitive tasks (such as standardized reports) and frequently used tables, the company leverages stored procedures, and indexes in SQL database. The company’s existing solution is shown in the picture below.



The Problem

Company currently hosts over 8TB of data in SQL Server today. Company has experienced hyper growth over the past 3 years. However, as the volume of data continues to grow, the existing single node SQL Server is not able to scale. Currently the company can only process the data nightly due to the compute capacity limitations. ETL processes and SQL reporting queries are running slow due to increased data volumes. Company has optimized the database by creating indexes and upgraded the server hardware to maximum CPU, RAM and storage configurations that the server can support. However, it has not helped significantly in terms of performance and scale.

Last week, there was a surge in data, during the nightly ETL process, the database server crashed and the whole system was offline for several hours. Engineering team has recommended purging some of the older data to reduce load on the server. Also, there is no rapid back-up and recovery plan. Hundreds of custom designed scripts (e.g SSIS packages) run nightly to complete the ETL process. Each script is custom designed to process each type of file. However, overall, the scripts perform the same kind of data transformations. Database backups are taken on a nightly basis. However, in the event of a failure restoring the backups to a new database takes hours and systems would be offline during the restore process, leading to risk and poor customer experience. SQL Server has become a single point of failure, hosting critical customer data.

Today, since the SQL database does not have enough capacity, doing analytics and reporting requires the company to export the required data on a nightly basis to seperate servers. Then this exported copy of the data is used for analytics purposes. This leads to multiple problems such as data duplication and wasted storage space. On top of this, exporting data on a nightly basis requires maintaining multiple versions. Keeping track of the most up to date data location for hundreds of tables is complex, time consuming and error prone. Discovering the latest copy of the data across the multiple departments of the company is not possible leading to data silos within the company. CTO would like to build additional capabilities with the historical data that company has such as building Machine Learning models, and near-real time dashboards containing patient data for each facility without the need to move the data from one system to another.

Existing Technical Environment

- 1 Master SQL DB Server
- 1 Stage SQL DB Server
    - 64 core vCPU
    - 512 GB RAM
    - 12 TB disk space (70% full, ~8.4 TB)
    - 70+ ETL jobs running to manage over 100 tables
- 3 other smaller servers for Data Ingestion (FTP Server, data and API extract agents)
- Series of web and application servers (32 GB RAM Each, 16 core vCPU)

Current Data Volume

- Data coming from over 8K facilities
- 99% zip files size ranges from 20 KB to 1.5 MB
- Edge cases - some large zip files are as large as 40 MB
- Each zip files when unzipped will provide either CSV, TXT, XML records
- In case of XML zip files, each zip file can contain anywhere from 20-300 individual XML files, each XML file with one record
- **Average zip files per day:** 77,000
- **Average data files per day:** 15,000,000
- **Average zip files per hour:** 3500
- **Average data files per hour:** 700,000
- **Data Volume Growth rate:** 15-20% YoY

Business Requirements

- Improve uptime of overall system
- Reduce latency of SQL queries and reports
- System should be reliable and fault tolerant
- Architecture should scale as data volume and velocity increases
- Improve business agility and speed of innovation through automation and ability to experiment with new frameworks
- Embrace open source tools, avoid proprietary solutions which can lead to vendor lock-in
- Metadata driven design - a set of common scripts should be used to process different types of incoming data sets rather than building custom scripts to process each type of data source.

  Centrally store all of the enterprise data and enable easy access

Technical Requirements

- Ability to process incoming files on the fly (instead of nightly batch loads today)
- Separate the metadata, data and compute/processing layers
- Ability to keep unlimited historical data
- Ability to scale up processing speed with increase in data volume
- System should sustain small number of individual node failures without any downtime
- Ability to perform change data capture (CDC), UPSERT support on a certain number of tables
- Ability to drive multiple use cases from same dataset, without the need to move the data or extract the data
    - Ability to integrate with different ML frameworks such as TensorFlow
    - Ability to create dashboards using tools such as PowerBI, Tableau, or Microstrategy
    - Generate daily, weekly, nightly reports using scripts or SQL
- Ad-hoc data analytics, interactive querying capability using SQL