Business Problem

ERGO Group AG is an insurance group with a presence in more than 20 countries especially Europe and Asia. The current project is a collaboration with IBM GBS and ERGO Life Insurance (LI). ERGO offers services covering accident and health insurance, group insurance, annuities as well as investment management wings. Ergo mainly required a tool for analysis of contract portfolios with ad-hoc reporting. Also, evaluations of benefits/claims, sales collections, risk management, complaint management. IBM wants to build a platform with standard processes in the cloud on which ERGO will be the first customer to manage their life contracts to offer the same to other insurers. So, the aim is to build a solution not ERGO specific but for the entire insurance industry.

To resolve this business problem, a big data solution has to be developed which meets the following functional and non-functional requirements

Functional Requirements:

- Data to be captured as a single version of truth for easier analysis.
- Migrate data from IBM MQ, SAP, DB2 data sources
- Develop an application to do batch streaming from DB2, SAP data sources
- Develop an application to do near real-time streaming from IBM MQ
- Process unstructured JSON / XML format of input data coming from IBM MQ
- Setup tool that triggers the ETL scripts for scheduled batch processing
- Install Cloudera with HIVE as Datawarehouse and Postgres as external HIVE Metastore Server
- Install services like HDFS, Spark, Yarn, Zookeeper, Hue, Oozie, Kafka, Cloudera Manager, Activity Monitor, Report Manager, Cloudera Navigator
- Provide SSO login to Cognos technical user / end-user to run reports / analyse data

Non-Functional Requirements:

- Incorrect data records to be logged for further analysis
- End users with restricted access to specific tables/databases/views
- Data privacy rules to be adhered
- Implement security features with Kerberos and Unix user roles/groups
- If job chain is interrupted, the subsequent job chain or dependant jobs must not be started.
- At least 99.5% of system availability and system scalability capabilities are required
- Define backup/restore, disaster recovery strategies.
- Automate deployment pipeline for release and test management
- To ensure high quality, run smoke and performance tests
- Technical description must be available for all attributes of a data model to a user via UI
- Customisation of Cognos UI to ERGO template and support for German language

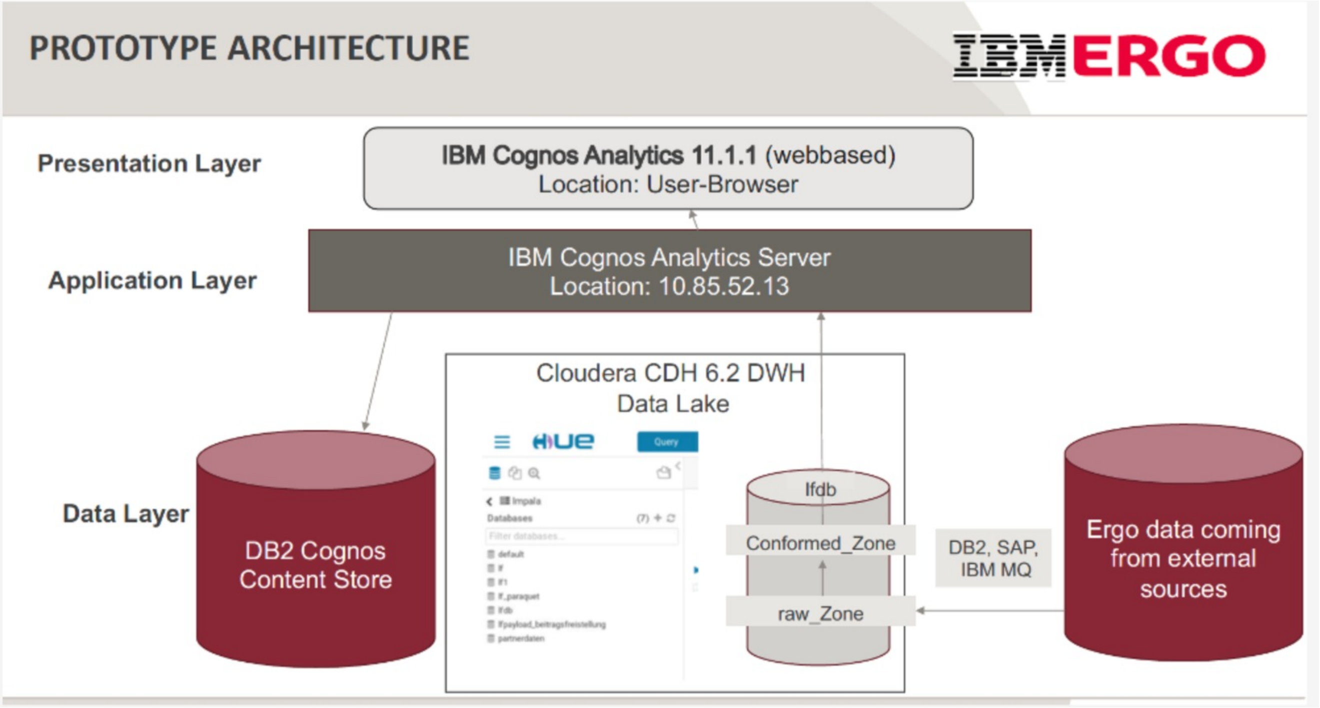
The project was divided into multiple releases. My scope in the project is for Release 0 and Release 1.

- During Release 0, I ensured ETL scripts were developed according to the defined data architecture. Here, we utilized the benefits of Spark for faster data processing. In particular, I led the offshore team in developing ETL scripts using Python-spark library for batch processing and Scala-spark library for near real-time streaming. To meet nonfunctional requirements, I have designed performance testing scenarios (Behavior-driven tests, smoke tests, unit tests) and created a platform that captures the results of each scenario to check if the solution meets the expected SLA. I setup review calls when needed so that selected tables are loaded to DWH with ETL scripts by the development team without blockers

- During Release 1, I ensured reporting is delivered with enhanced features. Here, we employed a Cognos BI tool for making reporting easier to business users. I collaborated with BI Analysts in setting up Cognos, establishing a connection to HIVE Datawarehouse, generating reports with selected tables. I also ensured the team customize the solution to ERGO UI design specifications to be able to integrate with ERGO Software. For security reasons, I made sure that we restricted access to end users by enabling the Cognos SSO feature. By the end of release 1, further features like the authentication of HDFS data using Kerberos, backup and CI/CD deployment strategies were developed.

Solution

Creating a big data solution involves validating the ETL architecture with the client and internal teams. The ETL architecture followed the principles of Digital Insights Platform which is a key IBM data platforms asset and method. DIP provides a collection of services for building a data lake. The integration services create and schedule ETL jobs/workflows to move data from outside sources to the raw zone. The analytics services transform data from raw zone to analytics zone for data analytics.



The design of the raw zone fulfils the following architectural decisions:

- Data curation from different source systems in an organized manner to efficiently retain untransformed data in the way that it was created by operational sources
- Build data science sandboxes from raw data to perform data science modelling and analysis.
- Only approved applications/users with write privileges can access datasets in raw zone.

The design of the analytics zone fulfils the following architectural decisions:

- Data to be optimized for read-only queries to avoid any delays when power user fetch results
- Meet the ad-hoc reporting and data analysis requirements of the client.
- Reshape data into formats necessary for making critical business decisions.

Some of the other key architectural decisions are mentioned as below:

- Cloudera: It is an integrated data platform for managing data flows, streaming, data engineering, machine learning, operational data stores and data warehouse environments. Cloudera has support for enterprise-level integrated functionalities
- HDFS: Unlike traditional relational database systems, HDFS is a scalable storage platform. It can store & distribute large datasets across 100's of inexpensive servers that operate in parallel.
- Spark: It executes much faster by caching data in memory across multiple parallel operations, whereas MapReduce involves more reading and writing from disk. This gives Spark faster startup, better parallelism, and better CPU utilization.
- Dev Tools: It is difficult to maintain complex HIVE / SQL logic with joins of more than 10 tables. The data manipulations are faster and easier with fewer lines of code using Python/Scala based spark libraries. For example, transforming and querying XML / JSON data.
- Cognos: This BI reporting tool meets the requirement of customizing the BI platform with ERGO UI interface designs, logos etc. Also, Cognos is intended for use by business users without technical knowledge to help them make knowledgeable business decisions.

The following points summarize my interactions with internal teams and stakeholders:

- I helped senior architect with the key architectural decisions and taking part in regular client meetings along with the project manager. I understood the data that needs to be integrated with the central DWH system to build business-level KPI's. I ensured the modules delivered by the DWH team are performance tested before integrating with other modules. I explained various functionalities of MVP solution in prototype presentations.
- I followed agile methodologies while communicating status reports in team meetings. I maintained transparency with tasks progress. I have established a knowledge management process and a framework for capturing architectural, technical design details in Confluence and code development in GitHub. This ensured that pro-active actions can be taken whenever decisions were to be made on improvements, issues, risks and blockers.
- I maintained collaborations with DevOps team for necessary system upgrades, sizing, resource scalability, backups with respect to DWH development status. For example, I created JIRA requests to DevOps team for creating VM's in DEV environments, increasing disk space, scheduling backups in collaboration with senior architect. These requests were mostly regarding the DEV environments RAM, vCores, Hard disk to distribute the workload.

Result

I have delivered architecture that led to the implementation of business requirements and ensured deployment included learned best practices and activities to provide the proficient data platform for serving the client needs.

The benefits of the delivered solution include:

- Improved efficiency of business process handling of structured/unstructured data such as contracts
- Reduced maintenance costs while facilitating an improved Data platform solution
- Enhanced flexibility and portability of applications and data for accommodating changing business requirements easily

Key deliverables:

The architectural design involved providing solutions that match business requirements. The key requirement was to build a tool for the analysis of contract portfolios with ad-hoc reporting. To achieve this, my vision was to break down the main goal into smaller requirements which included setting up the data platform, ingesting and integrating data with data quality checks, administering clusters within 3 build environments and deploying the solution. Following are the detailed specifics of the solution

- Setup Cloudera and install services like HDFS, HIVE, Impala, Hue, Spark, Kerberos, KMS, Kafka.
- Ingest data (selected tables by the client in each iteration) from DB2, SAP, IBM MQ data sources into the raw zone layer with Spark, Streaming, Python, Scala, Bash.
- Check the data quality of the incoming sources and align it with target sources using Spark schema check tool. Bad records are logged for further analysis.
- Integrated the data platform with Cognos for analyzing, visualizing, storing the ad-hoc reports. Apart from visualizing the business KPI logics, Cognos also deliver Cognitive Services like AI and ML to help automate data preparation, data validation and visual data modelling.
- Designed DevOps framework to enable solution CI/CD. Tools like Nexus and Jenkins were selected, as they are a good fit into architecture, supports solution lifecycle management.
- Provided access to the integration team and then to the client for accessing the tables in the analytics layer to be able to build ad-hoc reports in Cognos with minimal training.

Some of the measurable KPIs:

1. Identified in which years there were more contracts filed
2. Understood outliers and missing information from time-series data. Identified which life insurance products are selling in more numbers and which are obsolete.
3. Analyzed the tariff classes and the death count benefit within each tariff class. This includes a user-friendly visualization of these classes with the below-mentioned customizations
   - The user can select the desired tariff class via a drop-down menu. The death sums were grouped into value ranges with interactive visualizations.
   - Upon clicking a visualization, drill-down visualization of a table shows in-depth details. The user can export the filtered details into CSV / XML formats for further analysis.
   - The user was given demonstrations of how with minimal training he can reuse the existing report for creating new SQL queries or doing further changes.
4. Identify the currently valid data and data valid on last month. Understanding the processing time of a task completed by the clerk. Net processing times calculation for manual tasks from month start till end. Calculating the average count of manual process steps within the unit of work.

The key deliverables and the measured KPIs lead to the above-mentioned benefits and higher customer satisfaction for our prototype presented at the end of July 2019. Also, we have received appreciation for our second prototype presented with the enhanced features in March 2020. During these prototype presentations, I ensured the team meet the business requirements and utilized the right visual analysis techniques for building the KPIs. In the end, the client can filter and store data of business value from the Cognos reports. By the end of release1, the project status was changed to green.

Architectural Thinking

Digital Insights Platform (DIP) Architecture framework was utilized to gather client requirements and to coordinate proposal deliverables. I designed an initial architecture of building a Cloudera Data Platform with HIVE as a central data warehouse (DWH) and integrated it with the Cognos BI tool.

The motive behind this design was to store the data coming from relational databases, message queues in Cloudera Hive. And, to provide end-user access to DWH for analysis/reporting with Cognos.

The key architectural decisions detailed in the solution section of this document were taken from the work products of the Digital Insights method. Below were some of the major architectural decisions taken in the project to design a big data solution:

Data Discovery: I understood the requirements, involved in decision processes, business analysis along with client and internal teams.

Created and configured HDFS Files: I did the solution of how data is going to be defined in the HDFS & how it will be ingested and curated into the below 2 layers of the data platform.
- Raw Zone - Provided a visual description to view the raw layer platform as part of the overall data platform
- Analytics Zone – A visualization of the developed core Datawarehouse tables for power users querying

Data Model (ART 0519): I designed to partition the HIVE tables for faster query retrievals. Design to setup HIVE metastore in external Postgres to effectively retrieve metadata in case of node crashes. Hashed multiple columns into one id column for unique identification or creation of primary keys. Described logical entities, attributes identifying and describing those entities, and the relationships between them. This artefact typically includes the following constructs:
- Entities
- Attributes - Primary keys, Foreign keys, Data types
- Relationships

Test Findings (ART 0551): I summarized the analysis of one or more performance and smoke tests. Provided a detailed assessment of the quality of the tested items and the status of the test effort including:
- Scope of testing, what was done and not done.
- Test results findings on functions working and functions not working
- Conclusions about test metrics and results
- Issues outstanding, recommendations and action plan

Architectural Methods

IBM's Digital Insights (DI) method was employed as the delivery strategy for the ERGO Data lake platform. The use of DI method architectural principles emphasizes user input and open communication during the overall process to ensure higher quality delivery. Regular WebEx meetings with the client ensured transparency during all the below 4 phases of the project.

1. Requirements gathering and planning
   - I followed a bottom-up approach for estimating the effort for ERGOs Data lake platform user requirement
   - I did solutions with estimated time, tools and costs involved for meeting individual functional and non-functional requirements.
2. Solution design
   - Crucial architectural and technical decision points mentioned in the solution section of this document were made during this phase.
   - I prioritised business value components for the delivery

- Defined ETL work scheduling for data migration and storage

3. Development & Testing
   - I ensured we delivered business value by design by reviewing the developed code and testing each user story received from the client
4. End-user access
   - I realised business value by quickly giving the user access to the developed core Datawarehouse tables (or) data lake platform through Cognos UI endpoint.

Risk Management

In any project, the role of an architect is one of the most challenging ones as it requires knowledge of multiple fields. This includes both technical and managerial skills. Since I see myself as an able candidate who can get trained for this challenging role, I took an associate data architect role to build a solution under the supervision of a senior architect meeting the functional / non-functional requirements and architectural principles.
During my tenure, I encountered the following major risks and brought them to notice with solutions of how to integrate them within the current solution

1. Authorization and authentication of data at rest and data in motion
   - Data at rest - By end of release 1, Cloudera cluster with Kerberos KMS Encryption was installed to protect data residing in HDFS
   - Data in motion – The Concept of TLS/ SSL encryption within Cloudera manager was documented for securing communications over a network
2. Data backup and recovery:
   - I took a key role in mentioning the importance of disaster recovery, regular backup and heap management in all clusters and acted as a bridge between DevOps and Developer team for meeting these requirements.
   - Backup strategies like Cloudera Disaster recovery, scheduled backups and restore with HDFS snapshots were proposed
3. Data quality checks
   - BDD test driven development with scenarios to test the ingested target data is similar to source. This includes null checks, inserted number of rows checks, schema checks etc
4. Data security and privacy
   - Restricted access to data through user roles
   - UNIX based user roles setup were designed, validated with security.

Lessons Learned

Designing and implementing a big data lake solution to target elite clients like ERGO not only generate billion-dollar revenue to IBM but also sets a standard for getting future signings with other insurance clients. Foreseeing the future expectations of the client can gain the client trust in the long term. Staying ahead of our competitors with unique selling points can keep the client choose IBM over other vendors in future projects.

A large project with more than 100 professionals working as different teams like Input and output data management, Business process management, Integration Bus, DWH, DevOps needs a common ground. Whenever possible, it's better to choose similar frameworks, programming languages, CI/CD tools that are used as standard within the project by other teams. Maintaining a well-documented portal like confluence and giving access to all teams including client enabled us to follow and implement common standards. Moreover, the reuse of the existing assets already developed by other teams has saved a lot of time and efforts.

Overall key learning from this project could be summarized as follows:

- Take some time but start with the correct architecture
- Adopt a discipline of regular client engagement to have a continuous feedback loop.
- The above two strategies help in demonstrating value early and often increases client satisfaction and confidence with IBM.
- Maintain collaboration and transparency within the team, between teams and with the client which includes timely communication of blockers and risks. This also helps in a better understanding of project scope, deliverables and risk management.
- Adopting data-first approaches, long term solutions, efficient coding methodologies not only increase speed, accuracy and consistency of delivery execution but also decrease the integration difficulties at later points and potential delays in delivering the next releases.
- Most of these strategies would make IBM a standard partner delivering big data lake solutions not just to ERGO but for other insurance clients as well.