Literature Review & Project Statement

ABSTRACT

This paper will introduce the problem which many companies face: outgrowing existing workflows. As companies scale, workflows must be modified and revised to establish more concrete solutions to ensure data integrity and cohesive functioning of employees. A specific company based in the reinsurance business facing these troubles will have their data management problems reviewed in depth. The existing solutions will be evaluated as well as more general solutions which introduce more of the same issue: flexibility results in loss of integrity. Ultimately, a bespoke system is deemed to be necessary and an overview of the web technologies and the many moving parts that will need to be developed to create a functional, long-term, scalable solution will be presented. Finally, specific requirements of the project will be identified as well as measurable evaluation criteria which will be used to ascertain the ultimate success of the project.

Luca Ciraolo

Candidate 083178

Literature Review & Statement

1 Introduction & Problem Outline

Over the summer I had the opportunity to intern for AMRE, a Managing General Agent (MGA) representing a panel of International Reinsurance Companies based in Europe and Asia. These "securities" grant AMRE an authority to seek, underwrite and manage business on their behalf under the terms of the Binding Authorities (contracts between AMRE and each security stating the types of business that AMRE can reinsure as well as limits of risks or geography, maximum premium to be generated, fees, etc). Business comes to AMRE through intermediaries (reinsurance brokers).

Currently, AMRE underwrites approximately 40 contracts from 25 clients each year of account. Additionally, they must manage the run-off of previous years of account on behalf of their securities since 2014. As a consequence, they have to manage over 100 active contracts which require record keeping of results so that they can be reported to the securities quarterly. Each quarter, they record 400-500 statements of accounts and payment advice which must each be audited, evidenced by original documentation sent by intermediaries and then reconciled into their system. AMRE has 15 employees (analysts and underwriters).

Currently they store all their business data in 6 large spreadsheets. This was convenient when the company was very small as the employees were all familiar with spreadsheet software but as the amount of data and size of the company grows, this workflow is becoming more and more inconvenient and inefficient.

The data is spread across multiple files which requires a huge amount of human intervention to ensure data consistency. As the business is growing, increasing amounts of data need to be entered each day and because the workflow is so inefficient, AMRE has needed to hire more people to handle it. However, this has resulted in another major issue with spreadsheet software: only one user can edit a file at a given time. This means that hiring more people is only a short-term solution which is not at all scalable.

1.1 Current AMRE Workflow

In order to define the scope and specifics of the project, we must first analyse the current workflow at AMRE and look at the issues that we aim to solve.

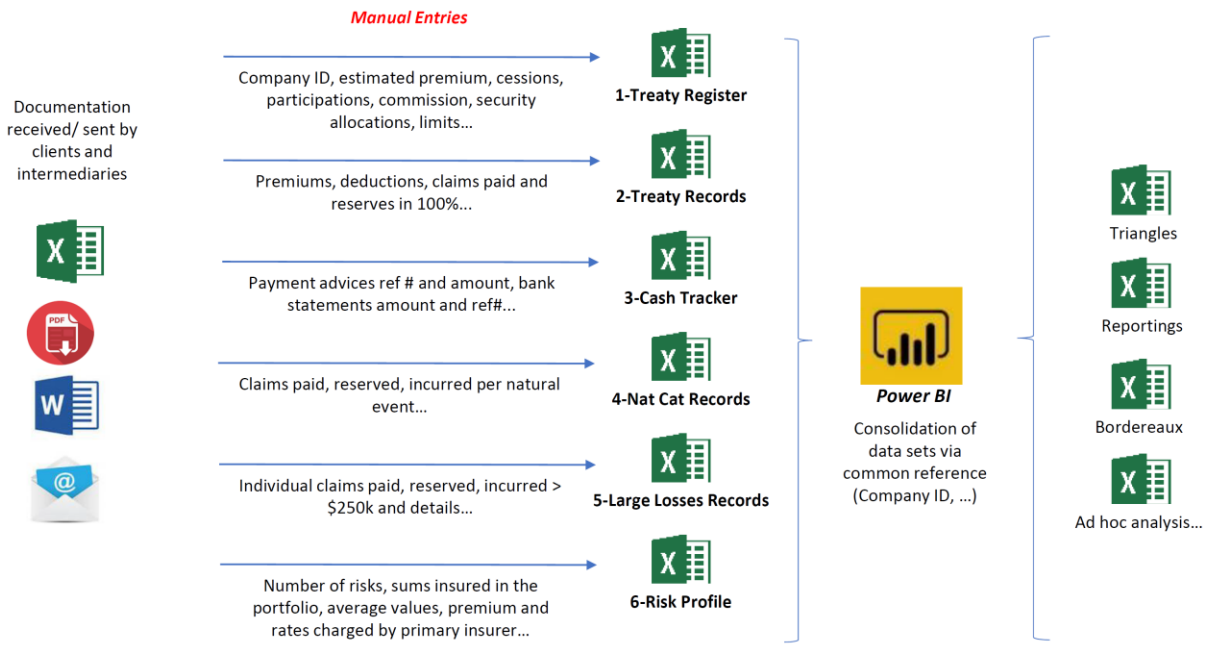

Figure 1: The current workflow at AMRE

1-Treaty Register: records of individual quota share contracts terms and conditions (estimated income to be generated by the contract, commissions, cessions, individual reinsurer's securities participation…). Information is entered in 100% and for AMRE's participation.

2-Treaty Records: records of individual quota share contracts results (premiums, claims, commissions, reserves…) on a monthly or quarterly basis. Information is entered in 100% and for AMRE's participation.

3-Cash Tracker: records of balances (Premiums Received –Claims Paid) due by/ owed to the clients. Reconciliation of balances with statements of remittance issued by brokers and payments in the bank. Calculation of AMRE's fees.

4-Nat Cat Records: records of client's total losses incurred, paid and reserved arising from natural perils (storm, fire…). Information is entered in 100% and for AMRE's participation.

5-Large Losses Records: records of client's individual loss incurred, paid and reserved > $250,000 for AMRE's participation. Information is entered in 100% and for AMRE's participation.

6-Risk Profile: records of individual quota share contract underlying portfolio (# risks, sums insured, average sums insured, premiums). Information is entered in 100% and for AMRE's participation.

1.2 Issues with the Current Workflow

- Storing all the data in Excel implies a lot of manual entries which are a source of human errors.
- Contextual information must be entered manually each time an entry is made in a dataset to create relationships between datasets under Power BI or Pivot Tables (eg. Client ID, Underwriting Years, line of business, reporting periods…).
- Entries must be entered in 100% and for AMRE's participation creating more room for errors.
- Data entry process is tedious and time consuming.

- Due to the nature of Excel, there is no audit trail of modifications of entries, whether the modification is made accidentally or intentionally.
- Excel does not allow for several users to access and make entries to the same dataset at the same time, which creates bottlenecks in the data entry process (especially record keeping of results).
- AMRE receives about 50 attachments or emails containing information relevant for the business daily (account statements, bordereaux of premiums and claims, advice of payments or other…). These documents are filed manually and are not "linked" to the entries in the data sets as it could be in a system.
- Due to the diverse currencies manipulated, fee calculations rules (one for each security) and status of entries this system has become too complex and is no longer viable.

This paper proposes to build a full web-based data/administration system to solve all these issues and greatly improve the efficiency of the AMRE workflow.

2 Existing Solutions

AMRE has looked into off-the-shelf solutions but the majority of existing software is aimed at primary insurance rather than reinsurance businesses. The few other companies which operate in a similar fashion have hired their own in-house development teams to build bespoke solutions to their specific workflows.

2.1 Software as a Service (SaaS) Solutions

There are however many software as a service solutions [1] aimed at businesses in general rather than specifically for the MGA reinsurance industry. According to The National Institute of Standards and Technology's definition [2], "[SaaS] is a cloud computing's architectural model in which the software itself is offered as a service to end user."

2.1.1 Microsoft Office 365

Microsoft Office 365 is a general business SaaS which gives subscribers access to industry standard software including word processing (Microsoft Word), spreadsheet (Microsoft Excel), presentation (Microsoft PowerPoint) software and many others [3]. AMRE has already been using Microsoft Excel extensively as their current data management solution. The main issue is that it is made to be very flexible to allow it to be used as a solution to many problems, so it does not enforce any standardization in the way the data is entered. There are no checks on data which means that any data aggregations can easily be broken if the data is entered incorrectly.

2.1.2 Business Intelligence Software

Business Intelligence Software is designed to provide insights and analysis into business related data [4]. It can be used to justify and guide decisions that are made. There is a huge amount of business intelligence software available as it is extremely valuable to businesses so there is a large market. Solutions include: Microsoft Power BI [5], Plotly [6], IBM Cognos [7], InfoZoom [8] and Oracle Business Intelligence Suite Enterprise Edition [9] and many more.

AMRE already uses Microsoft Power BI to create analytics dashboards and aggregate the data which is in their spreadsheets which works reasonably effectively when the spreadsheet data is correctly entered. Therefore, Business Intelligence Software is not a good solution as it does not solve the underlying issue that software developed to be flexible and apply to many situations is simply too flexible to be a robust data management solution.

3 Web Technologies

Existing SaaS solutions are clearly not a viable option, so the technologies which are used to build bespoke software solutions must be researched and analysed instead. As the project requirements state that the solution must allow multiple users to enter data at the same time, the solution will need to allow communication between client software through a network. The system will need to be accessible from outside the office therefore it will have to be accessible through the internet. A bespoke solution will have to make use of technologies which allow for communication over the internet.

Abdullah and Zeki [10] describe how many of these web technologies were used together to create social networking sites, specifically Facebook. They explain how web applications are split into two main parts: frontend and backend.

3.1 Back End

The backend of a web app usually consists of a database and a server-side application running on one or more servers hosted in a cloud environment [11].

3.1.1 Databases

Database software is used as an advanced data store. They handle concurrent updates to data without corruption and allow querying for specific data at fast speeds using indexing [12]. There are many different database engines as can be seen in Figure 2 which shows their prevalence over time. Database engines can be split into types, the most common of which are Relational Database Management Systems, Key-Value Stores and Document Stores.

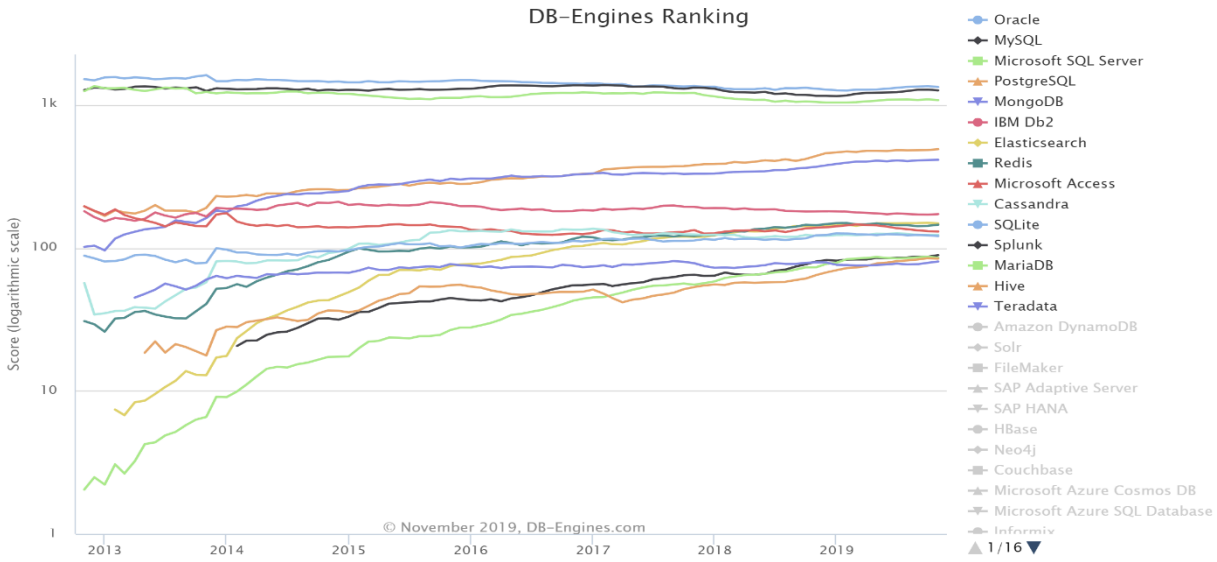

Figure 2: Database Engine Popularity Over Time [18]

3.1.2 Server-Side Scripting Languages

While databases are used to store the data, a server side scripting language is used to communicate with the database and perform operations on the data on behalf of the clients. These scripting languages are usually used to create an abstraction layer over the database, ensuring for example that the user is logged in and has permission to read or write data. These abstraction layers are

called Web Application Programming Interfaces [13]. Most programming languages can be used for web backends including Java, JavaScript (NodeJS) [14], PHP, ASP.NET and Python.

3.2 Frontend

The frontend is what the users (clients) use to interact with an application. Frontend applications can be written in most languages but in web applications, the most popular form of frontend is through a web browser. This is done through three different core technologies: HTML, CSS and JavaScript which are used by the browser to render the application. HyperText Markup Language (HTML) [15] is a markup language used to define the content to be displayed for example paragraphs, images and titles. Cascading Style Sheets (CSS) is "a stylesheet language used to describe the presentation of a document written in HTML" [16], for example the colour of an element or shadows and borders. JavaScript is the scripting language for web pages [17] and is responsible for interactive elements on web pages such as animations.

While it is of course possible to build web applications in pure HTML, CSS and JavaScript, there are a number of frameworks and libraries that have been developed and are used extensively in the industry to allow programmers to build applications in a fraction of the time.

3.3 Security

With an internet-facing application comes a significantly increased security risk. If care is not taken to ensure proper security measures are in place, attackers could potentially gain access to or modify confidential business data.

Unauthorized viewing and modification of database entries can be prevented in a number of ways including:

- Ensure database permissions are locked down to only allow the server-side application to make changes.
- Block all connections to the database except from the server-side application.
- Regularly updating the host operating system to patch vulnerabilities. [18] [19]
- Sanitize all form input to prevent SQL injection attacks. [20]
- Use transport level encryption (https) and certificates to prevent man in the middle attacks. [21]

3.4 Server-side & Client-side Rendering

Traditionally, clients would send a request to a server for a specific page that they would like to access. The server would then send the relevant HTML, CSS and JavaScript bundle which would then be rendered in the client's browser. However, the problem with this is that every time the user clicks on a link, they must wait for a response from a server which can result in a sluggish browsing experience [22]. In recent years, a new technique was developed whereby the server would initially send the client the entire application and then the client would be able to instantly render any page with no delay. The server would only be contacted in order to fetch or update dynamic data but the overall result is a much smoother user experience.

On the other hand, the downside to client side rendering is that the initial bundle which is sent to the client can be large and result in longer initial load times. The solution to this is a hybrid approach [23] whereby the initial page load is sent by server side rendering, and then while the user is looking at the page, the rest of the application is dynamically loaded in the background at which point the application switches to client side rendering for a smooth experience.

3.5 DevOps

"DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market." [24].

If we are to develop a long-lasting solution, it will need to be able to be modified and updated regularly. This will involve bugfixes as well as feature enhancements. The design of the system will need to be carefully thought out to allow other developers to easily understand how the different parts of the application work together. This will include careful documentation of code.

On the operations side, we will need to ensure that the system is scalable and can accommodate for company growth. It should therefore be designed and tested to work with at least 50 concurrent users. This will involve cloud server hosting with automatic scaling and load balancing. Thankfully, there are many technologies which have been developed to easily allow horizontal scaling by splitting the application into isolated modules which can be replicated and run concurrently in containers [25]. The most popular containerization technology is Docker. The application can be scaled automatically by a containerization platform which runs the containers like Kubernetes [26]. The system senses the load on the current containers and when it passes a certain threshold, will spin up more container replicas on new servers to spread the load. [27]

4 Project Specification

4.1 Functional Requirements

This paper proposes to build a full web-based data/administration system which must have the following features:

4.1.1 Data Entry

- Users must be able to add/edit/delete:
    - Securities (Reinsurers)
    - Binders (Binding Authority contracts)
    - Treaties (Reinsurance contracts)
    - Statements (Monthly/quarterly reports for each treaty)
    - Payments (Used to keep track of bank balances)

4.1.2 Multi-user Access

- Allow multiple users to visualize and enter/edit separate rows at once.

4.1.3 Access-control/security

- Certain users can only view/edit certain parts of the data using login credentials.

4.1.4 Audit History

- Record a history of any changes made to data so that there is accountability for changes made.

4.1.5 Backups

- Automatically take snapshots of all the data so it can be restored/viewed at any point in time.

4.1.6 Automatic File Organisation

- Be able to upload related files/attachments into the system so that they can be quickly viewed rather than having to find the files in a separate file system.

4.1.7 Relational Data Consistency

- Automatically handle relations between data to prevent human errors.

4.2 Non-Functional Requirements

4.2.1 Automatic Real-Time Data aggregations

Automatically update aggregations on the data whenever changes are made in real-time. Not as important because aggregations can be handled by the existing Power BI analytics software.

4.2.2 Nat Cat & Large Loss Data Entry

- Nat Cat and large loss recording can still be done in spreadsheets as it is not crucial to the daily functioning of the company.

4.2.3 Automatic Bordereaux Generation

- Reports for the securities which is done quarterly.
- Many companies request custom formats for the reports so it may be unfeasible to create many generated report formats.

5 Evaluation Criteria

In order to be able to evaluate how successful the project is, we must define specific tests which the project must pass.

- Is the codebase well documented?
    - Are all functions documented?
    - Are complex algorithms commented to a reasonable degree?
- Is there an adequate code coverage from automated tests?
    - Does every feature listed in the Functional Requirements section get tested at least partially?
- Can the system handle at least 50 concurrent users?
- Can users edit different rows of the data at the same time?
- Can the system be restored to a daily backup?
- Can users attach and upload files to forms?
- Can users view an audit trail of changes made to data?
- Does relational get referenced automatically without users having to enter complicated ID fields?
- Is access to data entry and data display limited to specific users?
- Can application updates be easily made without user intervention?
- Can the existing spreadsheet data be automatically imported into the new system with no data loss?
- Can the system's data be accessed by Power BI so that the existing dashboards still work?

6 Summary & Conclusion

In summary, this paper has introduced the problem which many companies face: outgrowing existing workflows. As companies scale, workflows must be modified and revised to establish more concrete solutions to ensure data integrity and cohesive functioning of employees. A specific company based in the reinsurance business facing these troubles has been identified and their specific qualms reviewed in depth. The few existing solutions have been evaluated as well as more general solutions which introduce more of the same issue: flexibility results in loss of integrity. Web

technologies to be used in the solution have been extensively researched, giving an overview of the many moving parts that will need to be developed to create a functional, long-term, scalable solution. Finally, specific requirements of the project have been identified as well as measurable evaluation criteria which will be used to ascertain the ultimate success of the project.

7 References

[1] A. S. a. D. N. C. Rumale, “Cloud computing: Software as a service.,” in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT).*, 2017.

[2] Mell, Peter, and Timothy Grance. "The NIST definition of cloud computing (draft)." NIST special publication 800.145 (2011): 7..

[3] Skendzic, Aleksandar, and Bozidar Kovacic. "Microsoft office 365-cloud in business environment." 2012 Proceedings of the 35th International Convention MIPRO. IEEE, 2012.

[4] Ouf, Shimaa, et al. "Business intelligence software as a service (SAAS)." 2011 IEEE 3rd International Conference on Communication Software and Networks. IEEE, 2011.

[5] “Power BI: Microsoft Power Platform,” [Online]. Available: https://powerbi.microsoft.com/.

[6] “Plotly - Modern Analytics Apps for the Enterprise,” [Online]. Available: https://plot.ly/.

[7] “Cognos Analytics,” [Online]. Available: https://www.ibm.com/uk-en/products/cognos-analytics.

[8] “InfoZoom,” [Online]. Available: https://www.infozoom.com/en/.

[9] “Oracle Business Intelligence,” [Online]. Available: https://www.oracle.com/middleware/technologies/bi-enterprise-edition.html.

[10] Abdullah, Hanin M., and Ahmed M. Zeki. "Frontend and backend web technologies in social networking sites: Facebook as an example." 2014 3rd international conference on advanced computer science applications and technologies. IEEE, 2014.

[11] M. Debusmann, K. Geihs, "Towards Dependable Web Services", 10 th IEEE Pacific Rim International Symposium on Dependable Computing , pp. 5-14, 2004.

[12] Manolopoulos, Yannis, Yannis Theodoridis, and Vassilis Tsotras. Advanced database indexing. Vol. 17. Springer Science & Business Media, 2012.

[13] Wittern, Erik. "Web APIs-Challenges, Design Points, and Research Opportunities: Invited Talk at the 2nd International Workshop on API Usage and Evolution (WAPI’18)." 2018 IEEE/ACM 2nd International Workshop on API Usage and Evolution (WAPI). IEEE, 2018.

[14] Tilkov, Stefan, and Steve Vinoski. "Node. js: Using JavaScript to build high-performance network programs." IEEE Internet Computing 14.6 (2010): 80-83.

[15] [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML.

[16] [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS.

[17] [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

[18] Kocher, Paul, et al. "Spectre attacks: Exploiting speculative execution." 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019.

[19] Lipp, Moritz, et al. "Meltdown." arXiv preprint arXiv:1801.01207 (2018).

[20] Boyd, Stephen W., and Angelos D. Keromytis. "SQLrand: Preventing SQL injection attacks." International Conference on Applied Cryptography and Network Security. Springer, Berlin, Heidelberg, 2004.

[21] Asokan, Nadarajah, Valtteri Niemi, and Kaisa Nyberg. "Man-in-the-middle in tunnelled authentication protocols." International Workshop on Security Protocols. Springer, Berlin, Heidelberg, 2003.

[22] Dharmapurikar, Makarand. "Method and mechanism for performing both server-side and client-side rendering of visual data." U.S. Patent Application No. 13/349,422.

[23] Naveen Kumar, S. G., et al. "A Hybrid Web Rendering Framework on Cloud.".

[24] [Online]. Available: https://aws.amazon.com/devops/what-is-devops/.

[25] Turnbull, James. The Docker Book: Containerization is the new virtualization. James Turnbull, 2014..

[26] [Online]. Available: https://kubernetes.io/.

[27] Cardellini, Valeria, Michele Colajanni, and Philip S. Yu. "Dynamic load balancing on web-server systems." IEEE Internet computing 3.3 (1999): 28-39.

[28] “Database Engines Ranking,” [Online]. Available: https://db-engines.com/en/ranking_trend.