

Bridging Accounting Data from Public Cloud to Private Cloud/On-Premise Accounting ERP System

Priyanshu Singh^{1*}, P. Ramakanth Kumar²

¹Student, Department of Computer Science and Engineering, R.V. College of Engineering, Bengaluru, India

²HoD, Department of Computer Science and Engineering, R.V. College of Engineering, Bengaluru, India

Abstract: Customers produce lot of transactional data for the suppliers these days. The reason being large volume of operations that need payment. These payments are channeled through different mediums. Various businesses face the difficulties of regulating and managing the accounts. The first important thing being, each order, stock item or certain operation on them is attached to a transaction. It doesn't seem a big matter in hindsight, but for the purposes of occasionally refunds, it is necessary to process them at the earliest. Traditional methods take days. It is necessary that payments, whatsoever type, is attached to a specific operation the customer carries. This complicated consolidation of all such type of data is possible using Accounting ERPs. Due to the emergence of highly sophisticated ERP software, taking right decisions has become faster. Funneling transactions through various streams is done using cloud-powered tools these days. Managing this data to organize, tag and formalize relationships between payment records and operations is what companies fabricate in-house. This paper build further upon reducing the technical barrier to streamline data-flow between cloud and on-premise systems.

Keywords: ERP, Transactions, Accounting, On-cloud, Kubernetes, CRON.

1. Introduction

Most executives, and accountants manually prepare reports for accounting. These use tools such as excel or other traditional reports that lead to waste of time. The time to make a decision also gets affected and results in overall downside.

Due to technological improvements, businesses now use ERP systems to manage their daily operations, transactions, working capital, finances, and human resources.

Enterprise resource planning is known as ERP. Enterprise resource planning (ERP) refers to a category of software that businesses use to oversee routine operations including supply chain management, risk and compliance management, buying, project management, and accounting. A complete ERP suite also includes enterprise performance management software. The functionality helps to enhance working with budgeting, prediction, and reporting the enterprise financial scenario.

Incoming and outgoing payments are made in real-time via accounting ERP. Additionally, it describes how to designate

payments that have been finalized, those that have been paid in advance but are subject to restrictions, those that must be repaid and will be deemed void, etc. Depending on the requirements of the business, these payments are subject to various constraints. These reasonings are intrinsic to the business, and the business must protect them. During transactional accounting, the conversion of one object to another is prevalent. On the confirmation of a payment, refund or other operations, the transactional data is ultimately converted to the accounting entry. Thus, continuing the process. This conversion is iterative.

Over the past 10 years, there has been a significant shift from premise services to cloud-based services. This has the advantage of reducing the costs of managing various payment flows and processing them according to international standards. Wherever you are, in any situation, you can easily get there. The downside is that this data is prone to liability to other parties. This can make it difficult for companies to use the same system. For security reasons, it is advantageous for enterprises to perform transaction funneling in the cloud and collect the data required for processing.

2. Related Work

Reference [1] is a study about Cloud-hosted ERP systems, while [2] explains the advantages of switching between on-premises and on-cloud architectures. These changes are evaluated based on various aspects, including performance, and security of providers, and respective communities. [3] and [9] explore the implementation of the software like ERP platform and event management through Kafka technology.

References [4] use JavaScript (React framework) to develop applications from a front-end perspective, [5] back-end (Spring Boot), and [6] Rest API. It describes the integration that was done. Reference [7] describes his automated testing of developed applications using Cypress (integration testing). This is very important before customer testing.

3. Accounting

Business objects like invoices, quotations and purchase-order are indispensable to a business flow. The reports for an

*Corresponding author: priyanshusingh6215@gmail.com

organization account for the same. Using transactions and accounting data to generate a report for the same.

Modern scenario demands that these processes are done in real-time. That is, the continuous channel of the transactions and other data coming in should be processed on the go. This is called stream-processing. This helps enterprises react to changing times and large-scale supplies.

Multi-currency integration is essential for group companies with subsidiaries, franchises, holding companies or other entities in multiple countries. As a growing business, you probably already need to transact, bill and report in multiple currencies. At some point in the growth process, we may also consider opening a branch abroad.

Setting up a local subsidiary has many advantages over simply selling across borders. This could be for compliance reasons or simply to get closer to the customer. However, you can dive into the more complex accounting world of multi-currency integration.

From the days of manual maintenance and ordering consumables, advances in technology have revolutionized the market with ERP software that adapts to the needs of client organizations and helps drive sales, forecasting and planning.

A successful company requires accurate financial reports. This can only be achieved through an error-prone transactions-payments process. After literature review, the opinion of experts on a successful accounting enterprise being:

- Timely invoice generation is necessary.
- Accurate account-entry to profile matching.
- High-priority actionable on failed-transactions.
- Expenses beyond the reach of manual accounting must be covered.
- Precise date-keeping and prediction helps to avoid market deflation in terms of stock tribulations.
- As mentioned, stream processing helps to increase efficiency and is a key to winning.

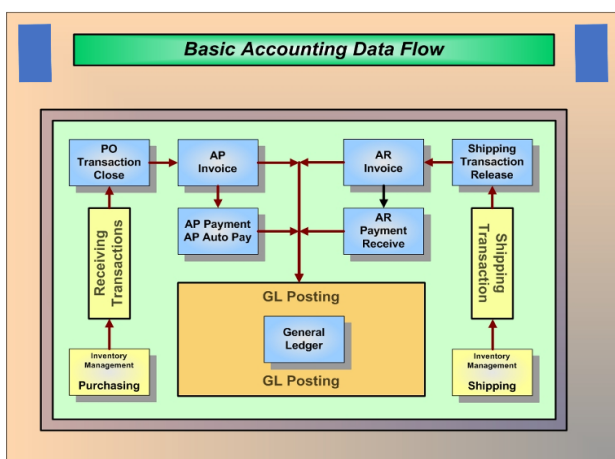


Fig. 1. Flow of accounting ERP

4. Technical Requirements

A. Front-End Development

1) ReactJS

React is a JavaScript framework. It has library of components

and frontend-rendering methods. Easily creates single-paged applications. This is so extensible that it has proved beneficial to be used in any user-interface application and provide necessary performance, far alone be flawless-functionality. The cherry-on-the-top being that the components provided are reusable and cuts down all the boilerplate issues one can face. It is easier to understand than traditional code-generator and binding for interactive UIs.

B. Backend Development

1) Postgres

Database chosen is a relational database here. The error-checking and precision a RDMS can provide is unmatched. The same goes for Postgres. Them ranging anywhere from mobile, to scientific, to enterprise, to geo-spatial for communications.

2) Spring Boot

Most of the enterprise support legacy code. This code is primarily in Java. We can use other backend frameworks like Django for Python but that would make our work a hassle. The objects used in the legacy backend need to be in compliance with the language used. Else, heavy-computation-cost serialization needs to be used. Spring Boot is made on top-of Spring Boot framework. It is easy to setup and configure. Runs out of the box and provides extreme customizability.

The software development rules used were also quite modern for the framework. Rapid Application Development. Automatic configuration and standalone applications are the new trend. Spring Boot promises great REST-API applications.

C. Deployment

1) Kubernetes

Whenever software is deployed, it needs to scale. Kubernetes ensures that. It provides a huge feature that leads to robust scale. Replicability, or multiple boots of a single instance that are readily available. Should the application or machine crash, another is replaced with the ill one. These pools of healthy machines ensure carefree operations after the deployment.

It is an orchestration system that automates deployment, with scale and systematic management. The K8 clusters include control and the worker planes. Our cloud-native as well as on-premise software can be hosted over Kubernetes. This is the case which demands us to create a containerized application for ourselves.

5. Architecture

Business objects need to be converted from one form to another for constant flow of the transaction stream. This is a critical process and needs to be precise. Any corruption of the data is unacceptable. The application contains the details of business transactions, so this is extremely essential to keep inflow.

Debit and payable on the given date time. This is the foremost objective. This can be ensured by timely date resolution and object transformation. The object transformation case is solved by using a similar Tech stack as our legacy Java code. The timely event stream processing has to be kept in check. This is done by proper scheduling, configurations of the

Kafka and robust K8 cluster.

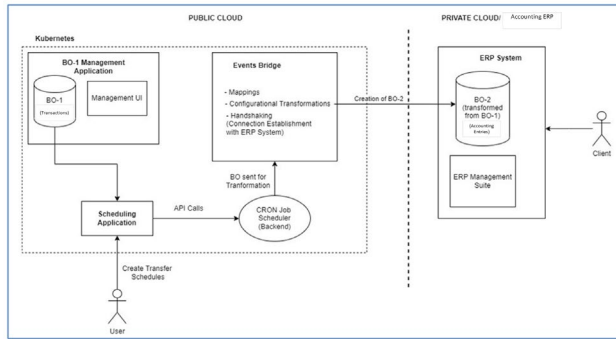


Fig. 2. System design

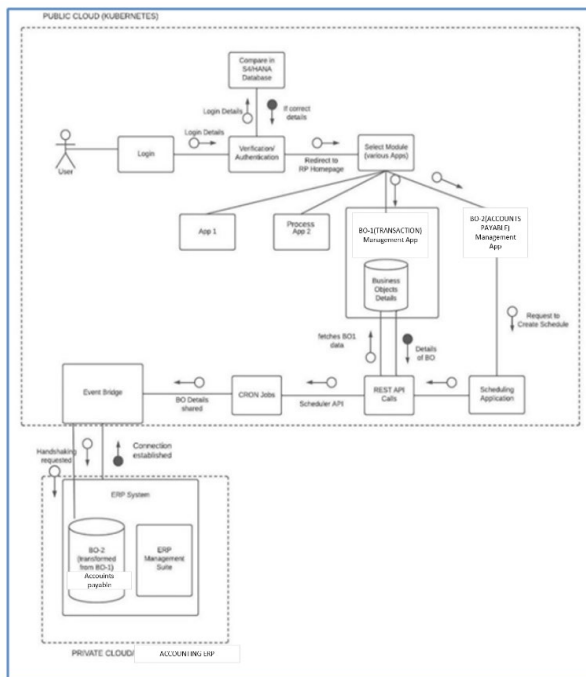


Fig. 3. Structure chart for the BO transformation process

The scheduling application is developed and integrated with Temporal. The robust fabrication of workflows ensures that no error seeps in and causes blockage in our event pipeline. This is supposed to be a recurring process. CRON job is a utility in Linux that allows scheduling and periodic running of the scripts. Details collected from the user are hence run in the server at specified times. This is a way of increasing the efficiency of automating tasks.

Once the plans are created to represent the different stages of the change from BO-1 to BO-2, the daily workflow becomes active. The Occasion Bridge consists of a back-end concept that coordinates the open cloud cluster and the customer's private on-premise/on-cloud ERP framework, allowing the modified trade protests to match his ERP framework and his BO-2 (i.e., accounts payable) is transferred as has been submitted. A purchase contract can be a formal archive sent by a buyer to a vendor or seller to authorize a purchase. A buy order represents what a buyer wants to buy and how much they want to receive.

The intelligent between the different pieces of the application are outlined, appearing the ask and reaction workflow.

Automation reduces costs and generates the right orders in the backend. This uses AI along with it. The event bridge is an analogy to the internet. Leading to service level optimization. Cloud operations facilitate business processes and enable uninterrupted access to new features.

6. Conclusion

In Maintaining the match between supply and demand is a central necessity in the chain of business forms. ERP executives have been encouraging the retention, review and understanding of information for a long time and are making progress day by day. On-premises computer programming is constantly being moved to the cloud to make computer program suites more available and reduce the burden on merchants.

Algorithms/models based on AI justification are sent to the server to perform the necessary calculations for renewal arrangements and other forms of trading.

Recharge agreements are the backbone of the retail business. Unused cloud-based administration services built and used by various organizations like SAP, Prophet, Blue There, etc.

This type of application is built by great association with worldview changing techniques in the cloud. Much work is underway and improvements in ML models is future work.

References

- [1] Scholtz, Brenda, and Denis Atukwase. "An analysis of the perceived benefits and drawbacks of cloud ERP systems: a South African study." Information Technology in Environmental Engineering: Proceedings of the 7th International Conference on Information Technologies in Environmental Engineering (ITEE 2015). Springer International Publishing, 2016.
- [2] Owen S. Cloud vs. hosted services, what's the difference?. IT NewsAfrica; 2011.
- [3] Chen, C., Liang, W., Hsu, H. (2015), A cloud computing platform for ERP applications, Applied Soft Computing, vol. 27, pp. 127-136
- [4] Xing, Yongkang & Huang, Jia Peng & Lai, Yong Yao. (2019). Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development. 68-72.
- [5] K. Guntupally, R. Devarakonda and K. Kehoe, "Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 5328-5329.
- [6] K. Guntupally, R. Devarakonda and K. Kehoe, "Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 5328-5329.
- [7] Mobaraya, Fatini & Ali, Shahid, "Technical Analysis of Selenium and Cypress as Functional Automation Framework for Modern Web Application Testing", 27-46, December 2019.
- [8] Grabski, Severin & Leech, Stewart & Schmidt, Pamela. (2011). A Review of ERP Research: A Future Agenda for Accounting Information Systems. Journal of Information Systems. 25.
- [9] Rompicherla, Rajesh and Bhaskar Reddy P. V., "Continuous Compliance model for Hybrid Multi-Cloud through Self-Service Orchestrator." 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE) (2020): 589-593.
- [10] Raul Castro Fernandez, Peter R. Pietzuch, Jay Kreps, Neha Narkhede, Jun Rao, Joel Koshy, Dong Lin, Chris Riccomini, Guozhang Wang, "Liquid: Unifying Nearline and Offline Big Data Integration," CIDR '15: 7th Biennial Conference on Innovative Data Systems Research, 2015.