



Secure And Efficient Attribution System To Enhance Digital Marketing

Attribution Engine

¹Sai Kailash B C, ²Dr Prakash Biswagar, ³Dr Abhay A. Deshpande

¹Student, ²Professor, ³Associate Professor

^{1,2,3}Department of Electronics and Communication,

^{1,2,3}R.V. College of Engineering, Bengaluru, India

Abstract : In this digital world, Business organizations always face problems to build and develop a system to maintain and process large amounts of data. There lies responsibility of every engineer to provide a feasible solution to the digital ecosystem. With the advancement in technology, integrating different features is the best solution. Attribution in digital marketing is the process of calculating the margin of profit for the effort of marketing a particular product. There are different ways of marketing to a consumer. It can be achieved through email, SMS, advertisement, campaigns and so on. It is very difficult to calculate the margin of profit with huge number of consumers and different marketing channels. An attribution system also known as attribution engine is required to deal with larger consumers and different marketing channels. This paper provides a systematic, efficient and secure attribution engine design. With the use of emerging technologies such as Amazon Web Services (AWS) and different software development methods, the Attribution Engine can be developed in a simplified manner.

IndexTerms – Attribution, AWS, AWS Lambda, API, Amazon SNS, Marketing.

I. INTRODUCTION

There are different driving events in a campaign. Advertising through SMS, email, ad display on websites, offline campaigns are some of the driving events in digital marketing. Similarly resulting events are generated due to these driving events. Email open, SMS open, impression click are some of the resulting events which lead to a transaction. A transaction is completed when the consumer purchases the product due to the campaign. Attribution is a process of calculating the margin of profit through three different strategies. First touch, last touch and fractional linear weightage are evaluated based on the driving events and resulting events which lead to a transaction. The design has to be made considering the speed of processing the data and security of the data.

To meet the above requirements, AWS lambda serverless architecture is used for faster processing. Automated workflows are used to generate driving and resulting event entities. A compatible user interface (UI) is provided to the clients to decide the attribution strategy, driving events and resulting events. APIs are developed for automated workflow creation and execution. Amazon S3 is used for storing intermediate files, control file and response files. Control file contains details of attribution engine execution, types of driving and resulting events. Response file contains details about the margin of profit for a transaction. Amazon SNS is used to trigger the lambda and notify that the attribution engine execution is completed.

II. KEY CONCEPTS

The paper deals with certain services offered by AWS which can be used to integrate and are part of the architecture. These are secure and efficient which make the business organizations deal with privacy of the data.

2.1 AWS Lambda: Lambda resembles methods or functions used in programming. It is a serverless cloud computational service provided by AWS. It is cost efficient as the price paid to use is only when the lambda is being triggered. It not only is cost efficient but also fast in executing the code. It is designed as a serverless event driven computational service which makes it compatible to be triggered by over different AWS services. It is the infrastructure for the code as a Service. AWS Lambda also provides easy debugging to the developers through AWS CloudWatch. The logs can be visualized so as to validate the code if there is no proper response returned by the lambda. The lambda can be triggered multiple times at the same instant of time which allows developers to access at the same time. The code can either be written in the editor or also upload as a .jar file generated by any IDE. Large business organizations are benefitted due to these features.

2.2 AWS CloudWatch: Cloud Watch refers to monitoring the services as well as the infrastructure provided by AWS. To debug the errors in a lambda during execution a single line of code `logger.log("Cloud Watch")` can be added. The Cloud Watch is a collection of logs with the time stamp as a file. This allows the engineers to open the logs when the lambda is invoked multiple times. CloudWatch is not only restricted to lambda but also other AWS services. Apart from debugging, the CloudWatch also collects metrics for latency control, memory usage and transaction capacity. The CloudWatch provides statistics in the form of a graph for monitoring APIs to calculate its performance.

2.3 AWS IAM: Identity and Access Management by the AWS allows only certain engineers working in an organization to access the services through Single Sign On. There are custom based policies defined for different services and can only be viewed or accessed to allowed developers or engineers. This provides a secure system as the services and infrastructure can be accessed only through single sign on. It can be coupled with lambda, S3, API Gateway making it an integrated secured system for all AWS services used. The permissions for policies can be edited based on the requirement. IAM permissions are also used to generate the required credentials for Virtual Private Cloud and also instances running in a VPC.

2.4 Amazon S3: It is data storage service offered to store any kind of data in any format. This has a unique feature to create folders and sub folders making it compatible to developer requirements. The data can be migrated from one S3 bucket to another S3 bucket easily with different regions trying to access the same. It can also act as an invoking action to the lambda. This also is infrastructure as a service. It is an object storage service which aims to provide scalability, low latency in accessing the files. The storage capacity of a single file can be as large as five terabytes. This capacity is greater than the average size of a file being used in the larger business organizations.

2.5 Amazon SNS: Simple Notification Service is a messaging and alert service directly delivered to business clients and also to other applications. SNS acts as bridge between two applications or an application and a service. It can be used to invoke the lambda for further processing. It can contain details about the application execution. The SNS messages can also be encrypted which makes it private to users through the Amazon Key Management Service. SNS can easily be integrated with applications being written in Spring boot and Python framework which are the widely used.

2.6 DataHub: This is an open-source platform to manage intermediate data known as meta data. It provides different features required for handling meta data and perform CRUD operations. This enables end to end data automation, observability and data governance. It can be easily deployed to AWS through Elastic Kubernetes service. This is used in business automation. The different queries can be executed and have different stages to deal with the metadata. AWS services can be used in the automation services.

III. ARCHITECTURE

The architecture of the design is shown below. This is a brief layout of the design of an Attribution Engine. The Attribution Engine uses four lambdas that are triggered sequentially. The entire process is sequential because the result of one process acts as a trigger to another process. In this architecture different AWS services are used more than once to deal with larger amount of data. The sales teams of any business organization have data related to marketing, feedback, transaction and so on. This data is called Promotion data. Marketing is done through different channels such as email, SMS, phone calls, offline advertisements. All this data is generalized as promotion data. There exists a scalable and compatible user interface to put forward the strategy, date range, driving and resulting events. A backend application is required to fetch these details and update the lambda for further process of executing the automated workflows. Certain APIs are exposed which contain the details provided in the UI and can also act to invoke the lambda. API Gateway trigger feature is also used. The workflow follows certain business rules for processing the data. Each workflow has certain stages to deal with the metadata and this data is used to invoke respective lambda to generate a file containing the outcomes of each workflow completion. This file is dropped in S3 which automatically triggers the next workflow and complete the entire process to generate a strategy file at the end for the specified date range. SNS is used to send a completion message or failure message for attribution engine execution. Amazon CloudWatch is used for debugging and monitoring if there is a failure SNS message or failure of any process in the business automation.

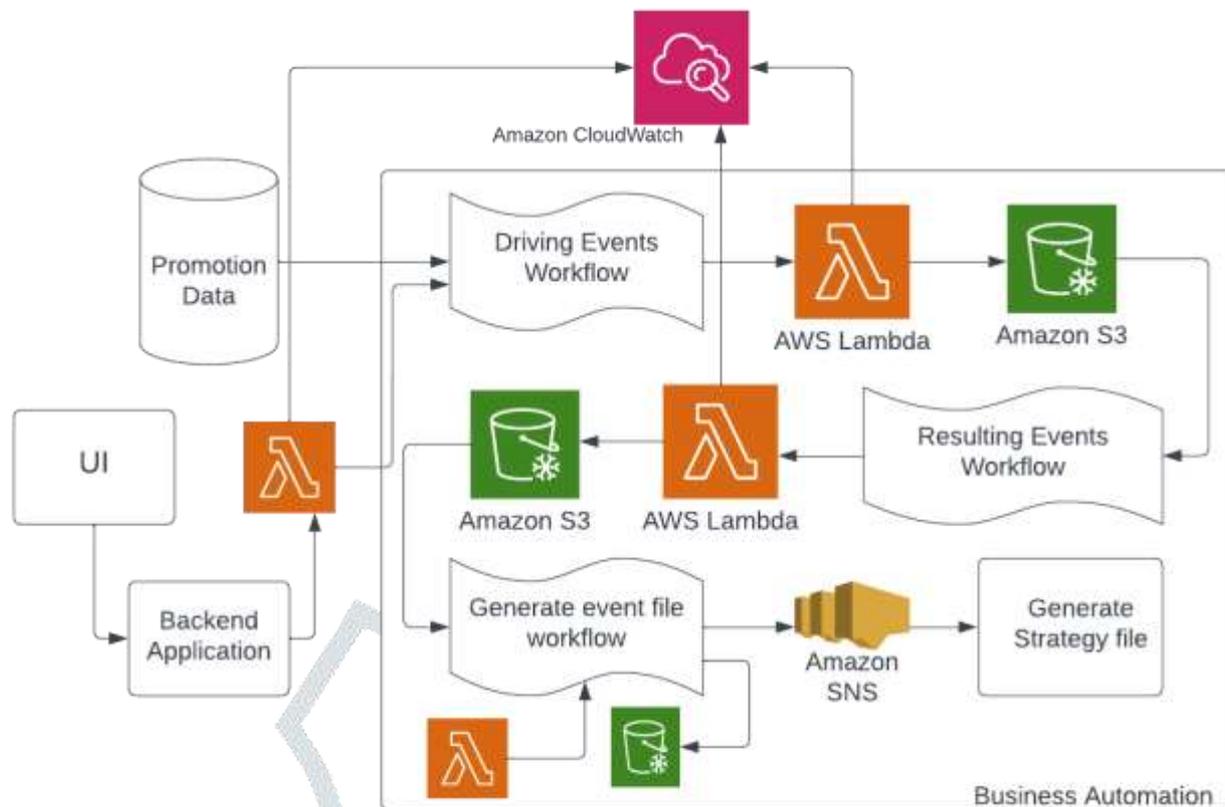


Fig-1: Attribution Engine software architecture

IV. FEATURES

The Attribution Engine exhibits certain features as follows:

- The Attribution Engine is flexible to modify depending upon the Promotion data.
- Developers can easily add new features in the backend application and update the lambda.
- The entire process is automated after defining the events and date range in the UI.
- Can alter the workflow stages through calling certain APIs.
- Payload modifications to invoke the API are based on the data entered in the UI.
- Alert messaging notifications for failure stages through CloudWatch.
- SNS message to acknowledge the completion.
- Generate the strategy file after the process completion.
- Advanced Attribution can be applied and have certain Machine Learning models to analyze the data.
- Date range is specified to make the process faster as there are larger transactions involved in a promotion.

V. METHODOLOGY

The user interface is developed using Angular. There are certain tabs like Attribution Strategy with a drop down having to choose First Touch, Last Touch or fractional linear weighted. Driving Event drop down such as email sent, advertisement, SMS sent, Push notification, phone call promotion and so on. Resulting event drop down such as email open, link open, buy the product, unsubscribe campaign and so on. There is a date range in which the Attribution Engine process the events that have taken place during the specified period. Since the transactions, events, data are extremely massive, the process is narrowed to a certain period. This data is referred to as Attribution Definition.

The backend application is designed to capture the data being entered in the UI. The Attribution strategy, driving events and resulting events are captured through the UI. Certain APIs are developed to invoke the lambda. The payload used to invoke the API contains the details entered through the UI. If the payload is relevant and verified the API triggers the lambda through Gateway and the attribution engine is initiated. The API returns 200 as response if it invokes the lambda. The backend application is developed through a Spring Boot Model, View, Controller (MVC). The payload is real time and changes every time there is an information entered through the UI and submit the Attribution Definition. API response looks like this

```
{
  resultCode : 200
  "statusMessage" : "Lambda Invoked"
}
```

The lambda contains code to trigger the first workflow which is Driving Event workflow. The code is uploaded as .zip or .jar file. This is saved and need not be uploaded every single time. For further development process, the lambda can be updated. There are certain stages in the driving events workflow so as to validate the data from the promotion database. The promotion database has different columns related to campaigns, driving events, resulting events, marketing channels, transaction details, feedback details and so on. The workflow creates intermediate entities. These entities are used to store the intermediate data also known as meta data. The data is generated through different SQL queries based on the Attribution Definition. Some stages are Populate, load burst events, count check and so on. These are sequential and are generated through SQL queries. After all the sequential stages of execution, a driving event file is generated by triggering the second lambda after all the stages in the workflow status is marked as Success. The file contains details about the transactions and resulting events that have occurred in the specified period. This file is dropped into S3. This S3 file drop is used as trigger for resulting event workflow.

The resulting event workflow reads the driving event workflow and creates temporary entities to deal with the data and business rules and business rules are being written for different resulting events to calculate the fractional weight of each resulting event. This depends upon number of business rules that the organization follows to allot weight for each resulting event. It might depend upon the type of the product, price, profit, campaign and so on. The weight is allotted to each resulting event for a particular campaign and for a single consumer. After the weight is allocated, the third lambda which is the resulting event lambda is invoked to generate a resulting event file and is dropped to S3. This acts as a trigger to generate a summary of file called as event file.

The event file is generated gathering the details of resulting events and also other information about the engine run, real time driving event and resulting event list along with details of products and weight of resulting event. This is .json file. This is when the fourth lambda is invoked. A SNS message is sent to the respective authorities to acknowledge the completion of engine execution to generate the event file. This message is also sent to generate the strategy file which contains margin of profit through the weight allocated. This weight can be used to calculate sales profit, improvement and so many economic and business growth factors. The event file contains some of the details like this:

```
{
  attributionId:09827ee98727ff-349061gj;
  awsaccountId:7899223456edff-790127456udh-003;
  drivingevents: ["EMAIL", "SMS"];
  resultingevents:["click","open"];
  date:"2022-05-27T04:17:00.000Z";
}
```

All the lambdas can be updated for any modifications in the business rules. There can be new APIs exposed because all the lambdas do follow MVC. Amazon CloudWatch is used for infrastructure (lambda) monitoring. Any failed execution in any stage of the workflow, logs can be used for monitoring and also debugging so that the lambda can be updated again knowing the messages thrown at a particular instant of time. The SQL queries if generates no data the count check stage fails and the logs indicate zero count through the lambda.

To have a system health check on the Attribution engine, a dry run could be performed with duplicate campaign promotion data and check the latency, number of requests per minute and functioning of the API through their performance. Lambda performance is evaluated by the runtime and also through the CloudWatch which provides statistical analysis. Roles and permissions are defined for each lambda which are the policies provided by IAM. Each lambda is equipped with different variables such as S3 file path, AWS Account Id, SNS trigger and so on.

The strategy file contains details about the profit of margin for the products. The event file details and data from the strategy file are used to evaluate business growth or decay for the sales in digital marketing. The Advanced Attribution process comes to light to evaluate the interest of different consumers, product performance and efficiency of sales through various number of factors from the data being generated through the Attribution Engine. Suitable machine learning models can be used to monitor the transaction and sales so that the sales team can improve the performance on the campaigns to enhance digital marketing. This remains as the future work design and development which will be dealt in the later section of the paper.

VI. SUMMARY

This paper designed and developed a secure Attribution system known as Attribution Engine which deals with large data and also efficient and secure system to enhance digital marketing. The software architecture uses most of the AWS concepts and technology which is widely used in the Business organizations. The APIs can be developed with the changing business rules that every organization has on it's very own. This Attribution Engine is an integrated architecture framed to meet the scalability and compatibility. The configurations that AWS provides have made the entire Engine look easier and simpler to understand. Most of the process is automated except giving the input to the UI. This is generally the business automation that is emerging and been able to meet lot of requirements. Developers and Engineers work together to improve the performance by introducing new features. This

can be easily done if the lambda is updated by new set of code written. The .jar file is generated in the local machine after having new implemented new features. There is a lot of future work which can be added to this Attribution Engine and will be dealt in the next section. The goal of enhancing the digital marketing is met with the Attribution Engine.

VII. FUTURE WORK

Advanced Attribution in sales is the future work of this paper where the new features are need to be developed to analyze and evaluate the consumers interest in different products. This should be done through different machine learning and neural network models which take the weight of the resulting event and the margin of profit for different marketing channels as the input to evaluate business and sales growth to benefit the organization. There can be new stages in the workflows which take the meta data, process and evaluate the performance of sales and also suggestions to improve sales through the promotion data. This enhanced Attribution will not only increase the sales but also brings to light about the efficient products that consumers buy. In this way the quality of the product can be improved based on the number of consumers buying it and also the feedback given during the campaign.

REFERENCES

- [1] Amazon AWS. <http://docs.aws.amazon.com>.
- [2] Winzinger, S. and Wirtz, G., 2019. Model-based analysis of serverless applications. 2019 IEEE/ACM 11th International Workshop on Modelling in Software Engineering (MiSE) (pp. 82-88). IEEE..
- [3] Berman, R., 2018. Beyond the last touch: Attribution in online advertising. *Marketing Science*, 37(5), pp.771-792.
- [4] Anderl, E., Becker, I., Von Wangenheim, F. and Schumann, J.H., 2016. Mapping the customer journey: Lessons learned from graph-based online attribution modeling. *International Journal of Research in Marketing*, 33(3), pp.457-474.
- [5] Arsanjani, A., Zhang, L.J., Ellis, M., Allam, A. and Channabasavaiah, K., 2007. S3: A service-oriented reference architecture. *IT professional*, 9(3), pp.10-17.
- [6] Kannan, P.K., Reinartz, W. and Verhoef, P.C., 2016. The path to purchase and attribution modeling: Introduction to special section. *International Journal of Research in Marketing*, 33(3), pp.449-456.
- [7] Villamizar, M., Garces, O., Ochoa, L., Castro, H., Salamanca, L., Verano, M., Casallas, R., Gil, S., Valencia, C., Zambrano, A. and Lang, M., 2016, May. Infrastructure cost comparison of running web applications in the cloud using AWS lambda and monolithic and microservice architectures. In 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid) (pp. 179-182). IEEE.
- [8] Buhalis, D. and Volchek, K., 2021. Bridging marketing theory and big data analytics: the taxonomy of marketing attribution. *International Journal of Information Management*, 56, p.102253.
- [9] Lin, C. and Khazaei, H., 2020. Modeling and optimization of performance and cost of serverless applications. *IEEE Transactions on Parallel and Distributed Systems*, 32(3), pp.615-632.
- [10] F. Petrillo, P. Merle, N. Moha, and Y.G. Guéhéneuc, "Are REST APIs for Cloud Computing Well-Designed? An Exploratory Study." ICSOC