Open Architecture for Loyalty Management System

Rahul K K

P G Scholar
School of Computer Science & Engineering
Reva University
Bangalore,India

Prasanth T

Associate Professor
School of Computer Science & Engineering
Reva University
Bangalore,India

***Abstract*—Loyalty Management System is a system which is intended to handle the programs which will be rewarding and attracting the customers of a brand to stay within the brand and reward them. Loyalty Program , otherwise called as points program where the prospective customers register for the loyalty program, keep accumulating the points and use the points for redemption. Industries like airlines,banking,telcom have heavily adopted loyalty program for catering the needs of their customers. The problem with the solutions implemented is it is customized to match that industry requirement. Open Architecture is a fast-growing widly adopted framework in the current techno world. In this paper we try to propose an open architecture for loyalty management system. With the architecture proposed in this paper we can use the system across all the domains, rather than focusing on the current domain specific loyalty systems. In this architecture we use the concepts of open api , reference from tm forum loyalty management apis to derive the outcome**

Keywords— Loyalty Management System;System Design, Open Architecture; Open APIs

I. INTRODUCTION

The competition arises in various business is inevitable and to compete with the competition business successfully requires keeping your current customers with the business itself, as it reduce the cost associated with the acquisition of a new customer and allows word of mouth marketing as well, which can bring silently many new customer to your business. Adding a new customer to our existing customer base comes with the additional cost of marketing cost, branding, provisioning the network to support the capacity required for supporting the new customers joining. So always the business side of a product will always looks into retaining the existing customer , by providing incentives ,motivating customers. Loyalty Management system is an well-established method to reward or incentivize loyal customers and increase the motivation of the customers to stay with the company. Existing loyalty systems are defined in a way that it is domain specific and tightly coupled with the solution also either in layered or monolithic architecture. In this paper we try to propose a solution which can be used across the domains and leverage the open api / architecture specifications laid down by various standard definition services. As per the study made out in this paper, it identifies that annual spent on loyalty program is approximately $1.2 billion on loyalty program in US alone [1]. Unfortunately because of the technical issues , or the complexity of the solution build, not so well planned business programs leads to the loyalty system to wind up before the end of life or results in the target being not met [1]. The main problem here is the points earned by a customer on spending a large amount will be quite small, which leads a not so attractive participation from the customer point of view. To solve these problems, Coalition loyalty program come into place, where the points from different business can be collected cumulatively and can be spend on different business so that the program will be more attractive and luring for the customer [2]. As mentioned in [3] more than half of the participants in the loyalty system is assuming that to reach a reasonable point balance to redeem a meaningful reward will take a long time and 53% of the enrolled members think that most of the rewards awarded by these programs are not meaningful and useful. To address these problems which we have highlighted here we intent to come up with an open architecture so that the core components of the loyalty program will be configured once and then based on the need of the business which is bringing in , customized modules are written which will act as an wrapper for the core services. The core services which will be required in a loyalty program is Account Management module , which manages the loyalty account for its lifecycle , Tier Management module which handles the usecases for the customer tier based on the spend or usage of services done by the customer, Accrual engine which manages the point accrual and deduction of the points , Transaction module – which will be acting as the module responsible for keeping the transaction history and the expiry module which is responsible for the managing the points expiry

program. To understand the customized solutions to be build based on the business domain consider following domains , the business domain of airlines industry , a typical loyalty program (termed as frequent flyers program) in the industry will require enrollment to the loyalty program , Accrual of the points , Tier programs and redemption , expiry of the points as well [4]. Here the accrual of the points varies from other industries, where the points are normally accrualed when a financial transaction is done or closed, where as in airlines industry , accrual happens only when the travel is completed after issuing a boarding pass. Similarly for tier management , there will be multiple tiers managed in airline loyalty where the status points defines the tier you belongs and based on the tier you are belonging you are entitled to get certain rewards , where as the 2nd type of tier which will decide on the multiplier for the amount spent when accruals are made. [5] Coming into fiancial industry , the loyalty programs involved in financial sector differs in certain part, firstly the enrollment happens as and when you open a account, or take a credit card and the points earned for certain financial transactions like bill payment, utility payment, card usage on POS[6] and these accrual doesn't happen real time , instead it happens once the transaction is settled, typically after couple of days. And the redemptions in banking industry is normally done, as an offline process where a redemption request is raised for a product and the confirmation is provided after couple of days. And coming to the Telcom industry , almost everything , be it redemption or accumulation happens real time as soon as the event happens. In this paper as we discussed earlier, we are putting forward an architecture on Open Digital Architecture (ODA) so as we can keep the core of the system as same and add plugins/ wrappers which can cater the need as per industry brought in . Loyalty Program on its core allows business to upsell and crosssell and indirectly allows company to earn more revenue by the amount of money spent by the customer[8].

II. DESIGN OF LOYALTY SYSTEM

Following section describes about the functional design of loyalty management solution proposed.

A. Customer lifecycle management

Any customer who wants to part of the loyalty program has to first enrol to the system. This enrolment can happen by 2 ways based on the program requirement either it can happen on auto enrolment (: case when a credit card customer automatically be part of loyalty program when the credit card been activated) or on demand (: Airline miles programs). This decision can be driven or it can be from authority compliance as well. Once a customer enrols to the program then there will be a unique loyalty id and wallet which holds the points balance will be created. Below diagram a typical of a loyalty account.

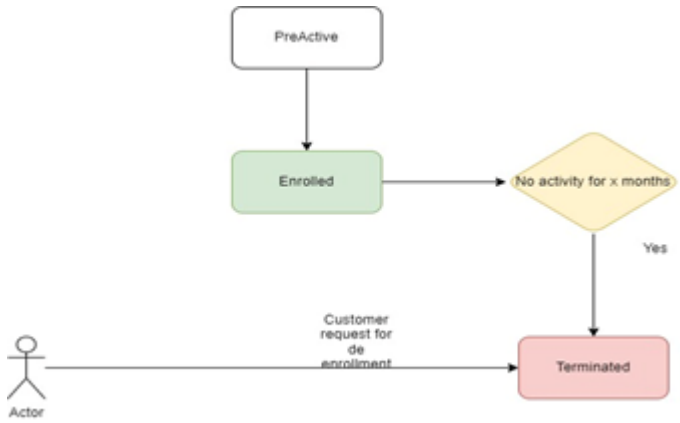

Figure 1: Loyalty Account Lifecycle

B. Point Management

Point Management is the functional part which is responsible for managing the points in loyalty account. Points can be modified in 3 different methods.

- Earn Points – This is the functional part which is responsible for accrual of points
  - Payment or Any monetary activities
  - Goodwill Points
- Burn Points - This is the functional part which will result in points being consumed from the wallet balance. Scenarios when point being consumed are as follows
  - Redeeming points for a service/product
  - Payment of a bill

C. Expiry Management

Any point which is earned in a loyalty system will have a validity which defines the date before which the point has to be used. If the point is not used beyond this date, then it will be considered as lapsed

D. Transaction Management

Any activities which is done on a loyalty account is stored as a transaction and can be retrieved by respective customers to view their activities (earn, burn, lifecycle changes, expiries). This will act as selfcare where the customer can verify how the points are getting updated.

III. ARCHITECTURE OF LOYALTY SYSTEM

In this section we describe about the reference architecture we propose as part of this loyalty management system architecture. There are couple of architecture decisions we have took as part of the design which is as follows.

- Any non-transactional communication between the micro services will be using on pub – sub model
- Any transactional communication between microservices or from external world will be over REST API or Batch Transfer based
- All inbound API call will be through the API gateway, no direct apis from microservices will be exposed externally.
- Any transactional update will be published as an event by the origin microservice to be consumed by other microservices
- No events will be generated by Transaction Management microservices

The proposed architecture as follows

.

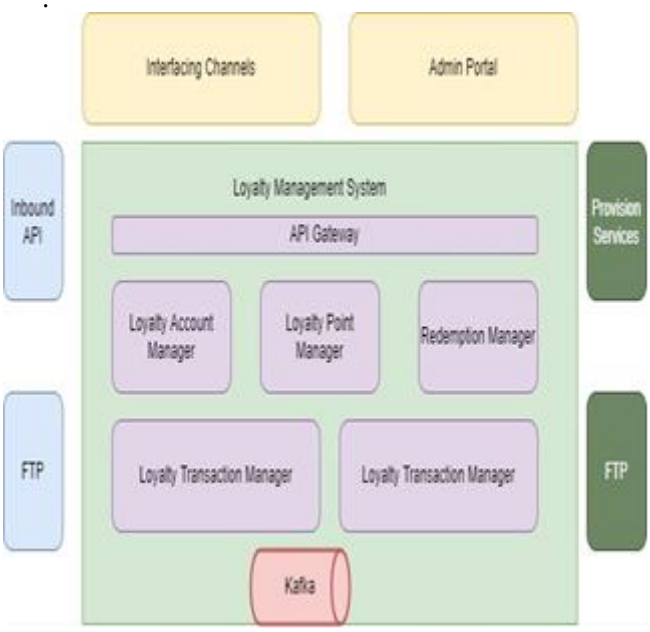

Figure 2: Loyalty System Architecture

A. API Gateway

API gateway is the module which will be the module exposing the api of internal services to the interfacing channels. API gateway will be also responsible for securing the apis with authentication and authorisation, ensuring the request is coming from the right sources, also to implement the scatter gather pattern to collect data from multiple micro services. Following are the benefits which we will receive as part of introduction of api gateway.

- Any addition, removal, change in underlying apis will not be bothering interfacing channels ,as api gateway will act as an abstraction layer
- Security configurations can be made independent of microservices.

B. Loyalty Account Manager

Loyalty account manager is responsible for managing loyalty accounts, its lifecycle, movement between different states. Up on creation of a loyalty account system will generate a loyalty id which can be uniquely identify loyalty account of the customer. In the loyalty account there will be other demographic information's which can be used for defining specific loyalty rules. Up on account creation or modification this microservice will generate events which can be consumed by other system, following are the events identified to be generated by this microservice account creation , lifecycle state update, account deletion.

C. Loyalty Point Manager

Loyalty Point Manager is the module which will be responsible for managing the points of the customer. This microservice will generate a wallet for the loyalty id when it receives(consumes) an account creation request event from account manager microservice. Whenever there is a need to add or subtract points this is the micro service which will get impacted. There will be apis for adding and subtracting points which will be exposed by this microservice which will be consumed by api gateway as needed. There is an exception of internal microservice call by points expiry microservice whenever there is a expiry happening. Any addition of points will have an additional entry/update in expiry table which will keep the points to expired. Any burn api will be looking to this table and consume out the earliest expiring points as part of burn which will make

sure that earliest expiring point will be consumed first. Following events will be published by this microservice, earn points, burn points, refund of points

D. Redemption Manager

Redemption manager is the module responsible for managing the different redemption mechanisms which includes activation of services/ purchases using points. Additionally, this microservice will interact with external loyalty programs to transfer points between different partner programs. Whenever there is an redemption request is received by loyalty management system, api gateway will make sure that there is enough points for redemption using account manager microservice, if there is enough points then point are reserved from point management microservice and if it is success then redemption microservice api is triggered to provision the service subscribed for, if the subscription is success then the reserved points are committed else reserved points are rollback.

E. Loyalty Transaction Manager

Transaction manager is the only microservice which doesn't publish any events. This microservice is kind of read only microservice, where there is no data changes are made by this service. Transaction manager microservice will be listening to all the topics and consume the events published and store the data in a partitioned table. This will allow this micro service to expose api which can be used to read the transaction for a period or last x number of transactions.

F. Points Expiry

Point expiry is the microservice which will run in active passive mode and will be only active for a smaller period. Most of the loyalty management systems goes for a weekly or monthly expiry, we will consider monthly expiry as this is the most common method of expiry used. This will scan all the non-zero-point records for the expiring month and initiate burn point api to remove the expired points.

IV. IMPLEMENTATION RESULTS

As part of implementing this reference architecture we have identified following advantages.

A. Abstraction with Introduction of API Gateway

Usage of API gateway as an entry point give us the flexibility to provide an abstraction of the core business applications which exists in the downstream. This will give the flexibility to replace an module which is there in the application layer without impacting the apis exposed to customer facing channels. For an example if the transaction history module needs to be replaced with another module which hosts all the transactions including loyalty points, then it can be easily done with minimal efforts. The effort which needs to be applied will be only on api gateway whereas previously each consumer of this api must be complied with the changes.

	For change in an API	
Number of Connected Channels	Impact Points on existing system	Impact Points on New Architecture
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1

Table 1 : Impact on Change in API

B. Plug and Play Loyalty System

Core of the system which is proposed here will be fitting into any loyalty system which uses points as the currency across the domains. Customization efforts will be applicable only on external interfacing modules based on the domain of the system. For example, if the loyalty system is used in telco redemption, then typically the redemption will be of addon packs whereas airline loyalty system will be using points for booking airline tickets.

C. Reduced Synchronous bandwidth

Since we have made an architectural decision to use events streaming as the main data transfer mechanisam between the modules this helped significantly in reducing blocking i/o, synchronous transactions and reduce the load over network. For example, the input to transaction history modules, if it is going as http calls will increase the network load and increase the response time for the respective modules, as the source modules has to initiate an http connection and receive the

acknowledgement that packet has been received by transaction history module. In this architecture, instead of this source module will publish an event which can be consumed by any of the module which is having access to the topic.

D. API Optimisation

Existing API which was exposed to external platforms is on XML over HTTP. We observed that the parsing and transfer of data is a bit heavy and take additional time to process the request. To optimize the solution, we planned to change the APIs exposed to REST API. With this we have observed 2x increase in the API performance. Additionally for the asynchronous API communication we used ZeroMQ for the api level communications, which was giving 5x performance over to the traditional API which was getting used.

Service	HTTP/XML	REST	ZeroMQ
Enrolment	40	85	210
viewUserDetails	80	185	437
Accumulation	60	140	330
Redemption	55	130	275
History	80	180	430

Table 2 : Transaction Per Second on various Interface

V. CONCLUSION

The architecture described in this document can be used as an open loyalty architecture and can be used across the domains to fit in various use cases arise to the domain. The core features of the loyalty platform is kept aside and only wrappers/custom modules required to support the delta requirement arise as part of the domain needs to be developed. In this model we can consider as an extension application of Artificial intelligence in recommending loyalty program journeys or providing redemption offer recommendations as well.

VI. REFERENCES

[1] L. a. R.W.Palmatier, "Understanding loyalty program effectiveness: managing target and byestander effects," *Journal of Academy of Marketing Science,* vol. 44, pp. 88-107, Aug 2014.

[2] B. &. B. &. Z. &. B. &. D. &. K. P. &. M. &. M. &. W. Els, "Advanced Research on loyalty Programs: a future resarch agenda," *Marketing Letters,* 2014.

[3] T. A. M. E. Osman Sonmezturk, "Loyalty Program Using Blockchain," in *IEEE International conference on Blockchain*, 2020.

[4] S. Andika Syafiq Baskara, "Design of Loyalty Systems based on Service Oriented Architecture," *International Journal of Advanced Trends in Computer Science and Engineering,* vol. 8, 2019.

[5] Andika Syafiq Baskara,Sfenrianto," Design of Loyalty Systems based on Service Oriented Architecture at XYZ Company" 2019 International Journal of Advanced Trends in Computer Science and Engineering, December 2019

[6] Pyotr Bochkaryov; Anna I. Guseva, "Application of the Ensemble Clustering Algorithm in Solving the Problem of Segmentation of Users Taking Into Account Their Loyalty" ,2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)

[7] Anna I. Guseva; Elena Matrosova; Anna Tikhomirova , "Evaluation Method of Loyalty Program Efficiency",2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)

[8] Evanschitzky H., Ramaseshan, B., Woisetschläger, D. M., Richelsen, V., Blut, M., & Backhaus, C, Consequences of customer loyalty to the loyalty program and to the company, Journal of the Academy of Marketing Science, July 2022