ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 13 (2023) No. 5, pp. 574-584
DOI:10.33168/JSMS.2023.0537

Designing Platform for Food Delivery Services Integration in Indonesia Using Microservices on Enterprise Systems

Johannes Farrell Landutama[1], Ahmad Nurul Fajar [2], Isabella Delvina Gozali[1], Christopher Putra Setiawan[1], Earlicha Mathilda Lionel [1], Jonathan Pangesta[1]

[1] Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480

[2] Information System Management Department, BINUS Graduate Program – Master of Information Systems Management, Bina Nusantara University, Jakarta 11480, Indonesia

Johannes.landutama@binus.ac.id, afajar@binus.edu, isabella.gozali@binus.ac.id, christopher.setiawan002@binus.ac.id, earlicha.lionel@binus.ac.id, jonathan.pangesta@binus.ac.id

Abstract. Today, there are many choices and options for choose food service delivery. However, it can make (1). Inconvenience, it is discussed about have to change applications, open and reopen, tired of searching, (2). Time consuming, it is talk about how much time does it cost to find the perfect match, and (3). Not always the best, it is described about still no guarantee if it is the best choice. This condition is a challenge to provide a platform that can act as a middleware to integrate food services delivery applications. It can solve the problem in food delivery applications. We collect data from literature and focus group discussion to create systems requirement in this platform. We used service oriented analysis and design approach to develop this platform. Then, we used software stack front end using React Native, back end using Golang, and web services using Fire Base and Google Cloud Service. The results in this study are user and system requirements for eatback, systems design for SOA systems, systems architecture for eatback, and SOA layer for eatback. At the end, we proposed a platform that can be act as gateway for integrate diversity applications with various language programming, databases, and platform also.

Keywords: Microservices Architecture, SOA, food delivery, platform, integration

1. Introduction

There are many platforms in the culinary field in Indonesia. It caused which one to choose related to availability or price. It makes looking for the best offer. As we know, there are too many options, such as (1). Inconvenient related to Have to change apps, open & reopen, tired of searching, (2). Time consuming, related to How much time does it cost to find the perfect match, and (3). Not always the best, related to Still no guarantee if it's the best choice. Many platforms in the culinary field are increasingly growing and popular from year to year. The development of the food service market presents applications and an increasing number of suppliers. Meanwhile, the development of suppliers of food delivery, namely restaurants, is also growing. The addition of the number of applications and restaurants certainly has a good impact where price competition is becoming tougher and also provides a much wider choice and wide. However, it also presents new problems for users and restaurants itself. The increase in the number of restaurants has indirectly made it difficult for some less popular restaurant due to oversupplied. Likewise for customers, too many choices make it difficult for them to choose and reach out to less popular restaurants. In addition, a large selection of applications also makes it difficult for customers to switch applications to get the best price from different promos on each application. Difficulties faced by buyers: Inconvenient: not practical, Time consuming: takes a lot of time to search, and not always the best choice: not necessarily the right one. For these problems, Eatback is here to provide good solutions for customers as well as restaurants. Eatback is looking for cashback and promos from these food delivery platforms Indonesia and integrate all existing restaurant data in all food applications services. With eatback features integrated with the food delivery application so that just fill in the data in eatback, eatback will calculate all possible promos from all customer accounts to find the best deal. The advantage that buyers get by using Eatback is that they can get the best price quote from a large selection of platform offers ordering food online, while saving shoppers time to choose the best offers from multiple platforms without the need to switch apps. Besides that, online food ordering platform that integrates with Eatback also gain benefits such as increased exposure and traffic because buyers will know offers made through Eatback. This advantage will encourage these platforms to always try to provide the best offer to customers so that they can provide new innovations to the food ordering market online (online food delivery services market.

2. Related Works

Service-Oriented Architecture is a technology architecture model for solutions that are oriented towards different characteristics in support and realizing service orientation and strategic objectives related to service-based applications (Erl, 2011). ESB is a software framework that manages services and their integration with other services on a common platform. ESB also provides the infrastructure support needed to implement message routing, protocol translation, and message transformation. ESB also aggregates services from legacy application domains and provides loose coupling across services. With ESB, one can design, develop, implement, and monitor services at runtime. This improves the concept of reusability in SOA (Bhadoria et al, 2017). Information system integration is a way to systematize and coordinate the recording, design, and implementation of transaction categorization and aggregation structures, which ultimately allows for the establishment and manipulation of a natural, comprehensive virtual perspective on operational and resource flows (Chapman et al, 2009). Middleware is a software layer between the physical layer and the application layer. Middleware provides a set of programming abstractions to facilitate the integration and communication of heterogeneous components (Fersi, 2015). The API Gateway will often handle requests by implementing multiple microservices and aggregating the results. It can translate between web protocols like HTTP and WebSocket and web protocols that are not commonly used internally (Rademacher et al, 2018). Domain Driven Design is a complex approach to software development where: Focus on core domains Explore models in creative collaboration of domain practitioners and software practitioners Speak languages everywhere in explicitly constrained contexts (Evans, 2003). Domain Driven Design combines design and

development practices, and shows how design and development can work together to create better solutions. Good design will speed up development, while the input that comes from the development process will improve the design (Avram et al, 2006). REST Webservices builds integrations in a lighter and simpler way, and focuses on resources (Rettig et al, 2015). The main idea of REST is the concept of resources as components of the application that need to be used or addressed (Rettig et al, 2015). The advantage for integration development on resources has been explained (Rettig et al, 2015). According to (Guinard et al, 2011), resources as components of the application are needed to be used or addressed (T.Erl, 2011). According to (Luthria et al, 2009), SOA can make shared and reusable business functionality, or application logic. It is has become one of the trends in systems development (Luthria et al, 2009). SOA follows principle of services oriented to defines the use of services to meet software needs (Erl, 2005). Related to (Erickson et al, 2008), The Open Group defines SOA focused on the service orientation architectural model. SOA is a paradigm that have capabilities to manage and use services distributed in different domains (MacKenzie, 2006). It is also designed and implemented in a loosely coupled and can be accessed in various platforms (MacKenzie, 2006). The challenges of future microservices have been discussed and explained by (Jamshidi, 2018). Then, microservices feasibility of economics and budget in the implementation of microservices, has been conducted by (Singleton, 2016). The specific definitions and core principle of microservices architecture (MSA) has been proposed by (Christudas et al, 2019). A study related to small and medium enterprises has been done by (Dong Yoon Lee, 2019). Besides that, critical success factor implementation in SME's has been proposed by (Anusuyah Subbarao, 2022).

3. Research Methodology

In this study, we used service oriented analysis and design approach. We gather data from literature and focus group discussion. We gather system requirement based on phenomenon, literature and focus group discussion. We create system design based on SOA and services principles. We used software stack front end using React Native, back end using Golang, and web services using Fire Base and Google Cloud Service. The research methodology can be explained in figure 1 below:

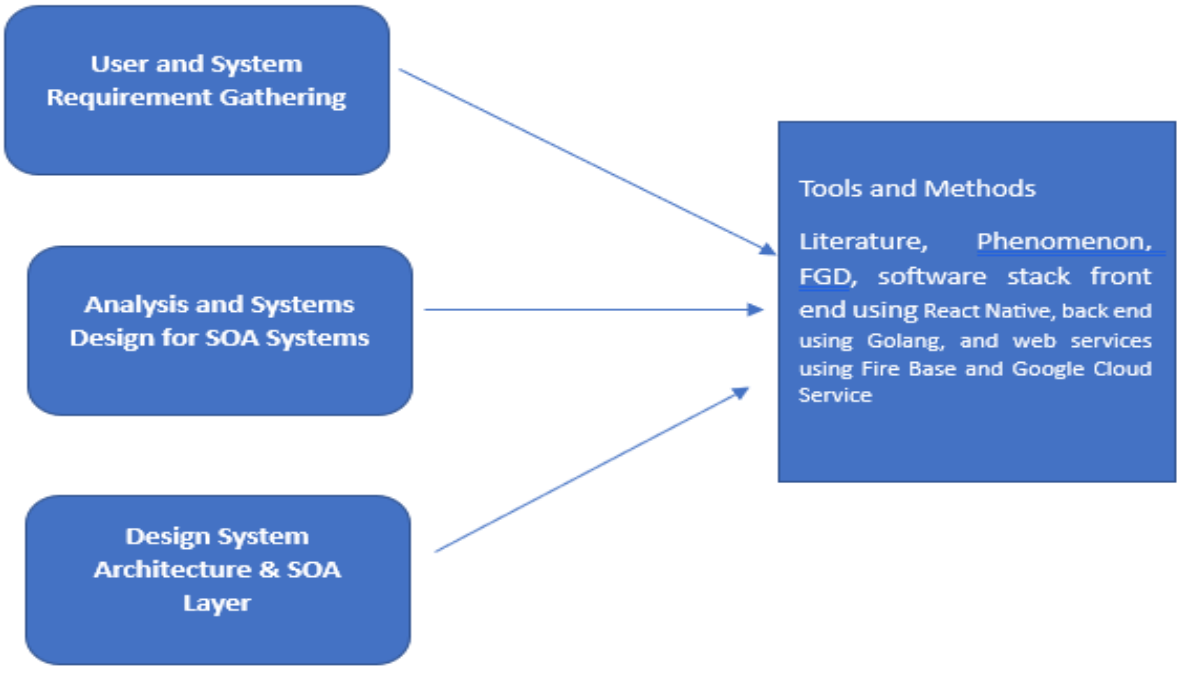

Fig.1: Research Methodology

According to figure 1 above, we can see the research begin from user and system requirement gathering, and then continued to analysis and design SOA systems, and the last stage is design system and SOA layer.

4. Results and Discussion

Chapter results and discussion consists of the results from user and system requirements for Eatback, the results of analysis and design SOA system for eatback, and the results of system design and SOA layer.

4.1. User and Systems Requirements for EeatBack

There are various food delivery platforms in Indonesia such as GRD, SHPFD, GFD, TRA Eats, and so on. Each platform has its own advantages such as GFD partner merchants, GRD with the system the most user friendly, SHPFD with the biggest discounts that are present every day, and etc. Of course, there are reasons why delivery platforms in Indonesia are interested in joining Eatback, such as increased traffic due to increased access to their platform, increased transactions, and improved performance platform due to the improvement of the two previously mentioned aspects. Eatback will use Firebase and GCP (Google Cloud Platform) as authentication service and the main server to store user data. User can signup and login to the eatback application to save the GFD account connection data, SHPFD, TRA eats, and GRD. Accounts are also used to store data user settings including profiling for the recommendation system. In the main service section of Eatback (Food Service Supplier), it will be connected 4 kinds of services from partners, namely: GFD, SHPFD, GRD, and Eats by TRA. Customer account if it has been connected to other food service accounts will be used to do listings and recommendations based on the lowest prices and customer food tastes. Recommendations from eatback will give you the best price customer as well as food that the customer may be interested in so that the customer does not need to check all four applications at once only through the eatback application only. Customers can then place orders at eatback by selecting food and menus from restaurants that have been selected at eatback. automatically eatback will do request an order to the restaurant through the food service application. This makes it easier customers to not need to open the food service application at all. For in-app payment purposes, Eatback will connect services from payment gateway such as Midtrans, so that payments will be made via eatback. Payment results will be forwarded to the food service related to the profit sharing system with eatback as specified in the existing partnership agreement. Listed price on eatback will be the same as the price stated on the food service. The eatback profit is a commission from the food service profit. With the eatback features above, customers can order food and search food much easier where all the restaurant data registered on the all food services are integrated in the eatback application. Partnering food service to get the advantage of increasing traffic and maybe even income because more customers to reach.

The summary of the features provided in the eatback application are:

- Integrated account: manage and integrate all of your food service accounts customers. Settings and extended profiling are also available in the eatback app.
- Integrated restaurant and food listing: provides a list of restaurants and food based on the best price recommendations and customer tastes covering all data that exist in the four food service applications.
- Quick checkout: can place orders on food service applications via eatback instantly because the account is already connected to eatback. Can do checkout multiple restaurants from multiple apps easily on eatback
- Easy share: share social media to increase food traffic.
- Advertisement: provide an opportunity for restaurants to do

advertising on eatback on the main eatback page, as well as on other platforms

4-2. Analysis and Design System for EatBack

We proposed Eatback systems, while it is can Find cashback and promo from food delivery platforms in Indonesia. It has 6 great features:

1) Surf and Scrape: Show and calculate every possible promo through scraping technology and decide who got the best deal
2) Quick Checkout: no hassle on filling the order twice! Once is enough
3) Easy share: share social media to increase food traffic
4) Integrated Account: manage and integrate all customer food service accounts
5) Advertisement: provide an opportunity for restaurants to advertise on eatback on the eatback main page, as well as on other platforms.
6) Integrated with restaurants: provide a list of restaurants and food based on recommendations for the best prices and customer tastes covering all data in the four food service applications.

The Impact for Customer and Integrated Platforms are follow:

Table 1. The Platform Impact

Impact	**Description**
Get The Best Deals	by choosing from the list of various food delivery service deals provided
Boost Exposure and Traffic	since more people will search and compare deals between integrated platform
Time-Saving	since customers do not have to switch between applications to compare the deals
Encourage More Innovation	in the online food delivery services market.

According to service oriented analysis and design approach, we create list of services that can be explained in table 2 below. It shows the service name, parameter, description, and resources that should be develop for eatback.

Table 2. List of services for EatBack

Service Name	Parameter	Description	Resources
user_profile	username, email, phone	To view user profile	Eatback
supplier_accounts	userId, GFDACT, GRDACTACT, SHPACT, TRAAccount	To view all linked food supplier accounts	Eatback
user_settings	location, foodPreferences	To view user settings	Eatback
list_merchants	Merchants data	To view merchants list	Eatback
payment_detail	orderId, paymentData, paymentMethod, paymentStatus	To view payment detail	
shareToPost	promoImage, promoUrl	To post food data to facebook	

orderDetail	orderId, orderItems, orderTotal, orderDate	To view order detail data	Eatback
listOrder	orderListData	To view all orders	Eatback
post_user_profile	username, email, phone	Update and insert user profile data	Eatback
get_user_profile	userId	Get user profile data	Eatback
get_supplier_accounts	userId	To get all linked Account Status	FOOD SUPPLIER SERVICE (getLinkedAccount)
post_login	Username, password	To log user into the system	Eatback
post_logout	userId	To log user out from the system	Eatback
get_oauth_token	userId	To get auth token to be used for other services	Firebase
post_settings	location, foodPreferences	Update and insert user settings	Eatback
get_settings	userId	Get user settings to be used in other services	Eatback
post_link_account	userID	To be hit by food supplier's system to notify logged in user	AUTH SERVICES (getOauthToken); FOOD SUPPLIER
callback_link_account	userID, refreshToken	To be hit by food supplier's system to notify logged in user	GFD, Grabfood,

get_linked_account	userID, oauthToken	To get user's linked data from supplier's system	GFD, Grabfood,
get_list_merchant	Keyword, filterData, sortBy	To get list of merchants in the supplier system	GFD, Grabfood,
get_list_items	Keyword, filterData, sortBy	To get food data from the supplier	GFD, Grabfood,
post_order_food	cartData	Service that will send buy request to supplier	GFD, Grabfood,
get_list_merchants	Keyword, filterData, sortBy	To integrate suppliers merchant data	FOOD SUPPLIER SERVICE (listMerchant
post_make_payment	orderId, paymentData, paymentMethod	To make payment request to payment gateway	To make payment request to payment gateway
callback_handle_payment	callbackData	To be hit by payment gateway to notify payment status	To be hit by payment gateway to notify payment status
get_payment_detail	orderId	To get payment details	To get payment details
post_product_feed	itemData	To update product data every 24 hours	

Then, we create user interface for Eatback Platform that can be shown in figure 2, 3, and 4 below. In figure 2 below, Eatback have features to make user ease of use the functionality. There are some software features for eatback that can be explained in figure 2 below. It shown the list of functionality that can be implemented in food delivery ecosystems.

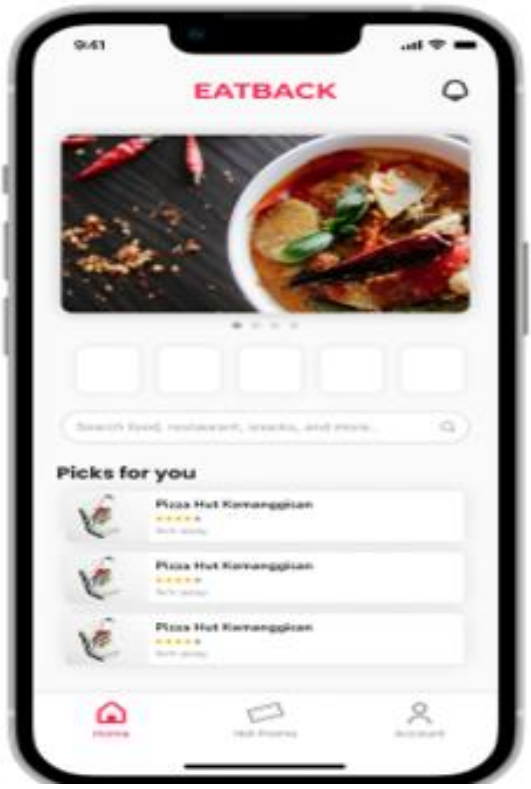

Fig.2: Eatback functionality

Figure 3 below shown the features in Eatback platform for best offering. This platform provides a recommendation system feature to provide menu recommendations according to the wishes of the user. For example, in this figure shown the GFD best offer, GRD best offer, SHPFD best offer, and TRA eats best offer.

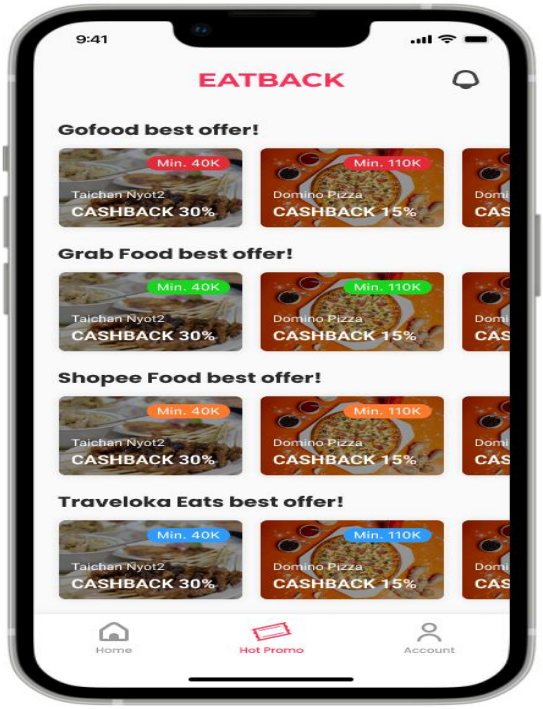

Fig.3: Eatback Offering Features

According to figure 3 above, we can see the featrures to inform the best offer from others applications. In this case, eatback can provde information about the best offering for food order such as cashback percentage. Related to integration systems approach, we develop Eatback platform that can connect to others applications in the ecosystem of food services delivery. Figure 4 below described the integration eatback platform with other applications.

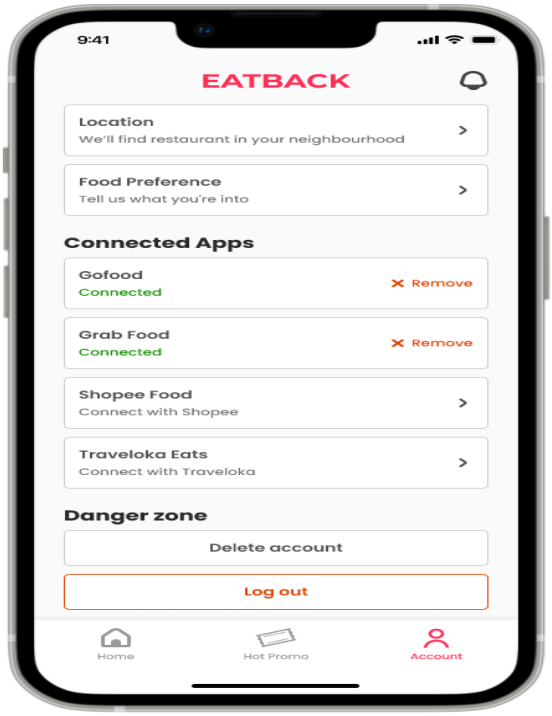

Fig.4: EatBack Integration

According to figure 5 above, this system platform has features that can connected to some applications, food preference, and get take location. This platform can provide preferred food preferences and according to price. In this figure, platform can connect to other applications such as GRD, GFD, SHPFD, and TRA eats.

4.3. Design System Architecture and SOA Layer

Eatback platform is developed to integrate ecosystems in food services delivery. As we know, there are many entities and applications in the ecosystem. In this study, we create Service Oriented Architecture (SOA) for Eatback Platform. It shown the mechanism of integration in eatbak platform. This platform can act as a middleware to support business to business and business to customers also.

The system architecture divided into applications and business processes which is give a mechanism for services requestor and services consumer to communicate the message exchange. According to figure 5 below, we create Eatback SOA Layer that consists of several layer. These layers are user layer, workspace layer, enterprise layer, and resource layer. These layers can communicate and integrate for data and information exchange.

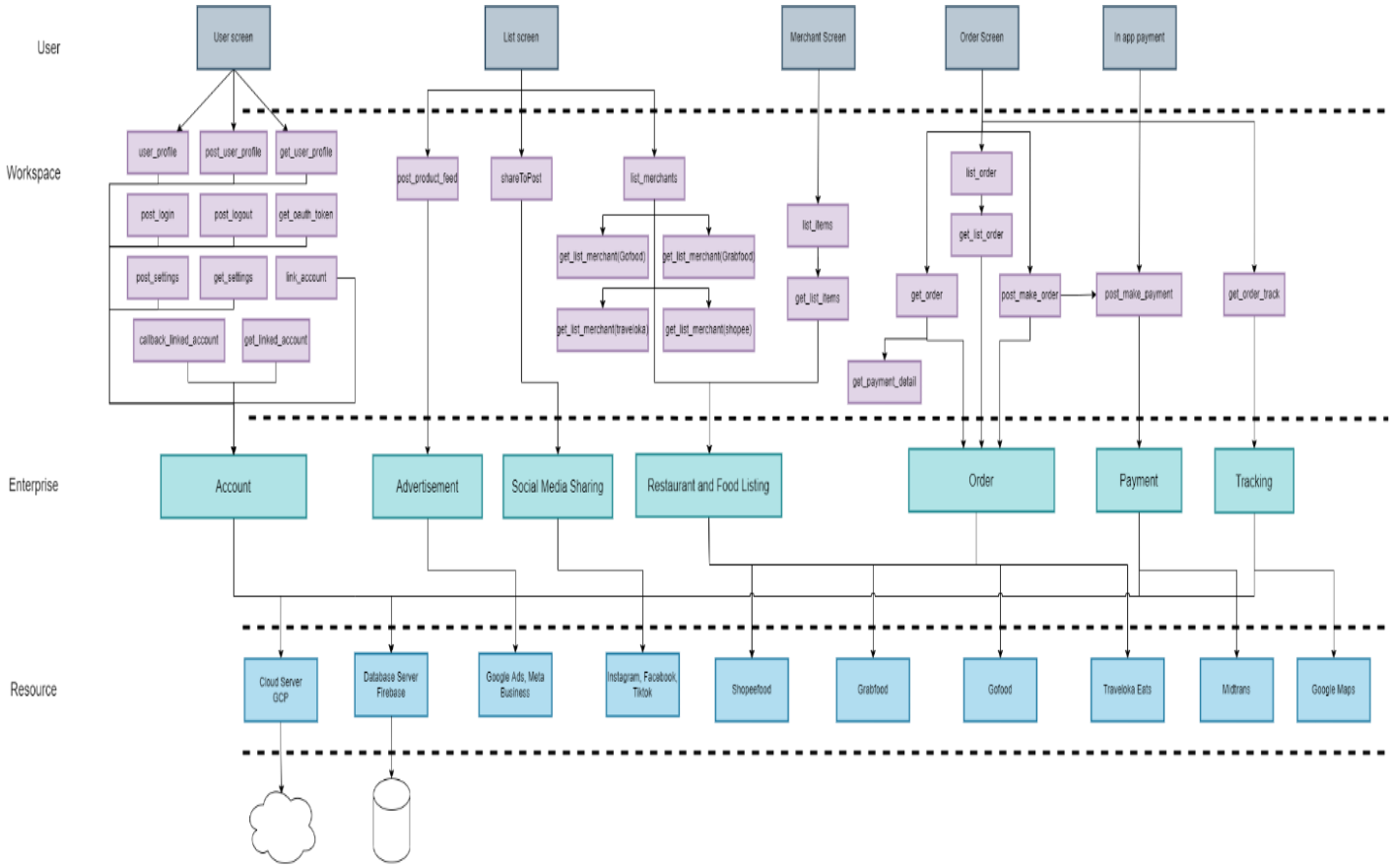

Fig.5: EatBack SOA Layer

According to figure 5 above, we can see the SOA layer for eatback platform. Related to enterprise layer, there are: account, advertisement, social media sharing, restaurant and food listing, order, payment, and tracking. Resources layer consists of cloud server GCP, database server firebase, google ads meta business, Instagram Facebook TikTok, SHPFD, GRD, GFD, TRA eats, payment gateway, and google maps.

5. Conclusions

Eatback is a platform based on microservices architecture. This platform is a middleware that can connected and communicate with different heterogenous applications. Eatback can be implemented for support customer and existing integrated platform. It has recommendation features based on data historical and user behavior.

References

A. J. Rettig, S. Khanna, and R. A. Beck, “Open source REST services for environmental sensor networking,” Appl. Geogr., vol. 60, pp. 294–300, 2015, doi: 10.1016/j.apgeog.2014.11.003

Anusuyah Subbarao, "Enterprise Resource Planning Critical Success Factors in Small Medium-Sized Enterprise", Journal of System and Management Sciences Vol. 12 (2022) No. 6, pp. 81-96 DOI:10.33168/JSMS.2022.0606

A. Avram and F. Marinescu, Domain Driven Design Quickly. 2006.

C. S. Chapman and L. A. Kihn, "Information system integration, enabling control and performance," Accounting, Organ. Soc., 2009, doi: 10.1016/j.aos.2008.07.003

Christudas, B., & Christudas, B. (2019). Microservices Architecture. In Practical Microservices Architectural Patterns. https://doi.org/10.1007/978-1-4842-4501-9_4

D. Guinard, M. Mueller, and V. Trifa, "RESTifying Real-World Systems: A Practical Case Study in RFID," REST From Res. to Pract., pp. 359–379, 2011, doi: 10.1007/978-1-4419-8303-9_16.

Dong-Yoon Lee, "A Study on Smart Factory Efficiency of Small and Medium-sized Enterprises Based on Electricity and Electronics". Journal of System and Management Sciences Vol. 12 (2022) No. 2, pp. 431-442 DOI: 10.33168/JSMS.2022.0223

Erl, T. (2005). Service-Oriented Architecture: Concepts, Technology, and Design. In City

E. Evans, Domain-Driven Design - Tackling Complexity in the Heart of Softwarey. 2003.

Erickson, J., & Siau, K. (2008). Critical success factors in SOA implementation. 14th Americas Conference on Information Systems, AMCIS 2008.

F. Rademacher, J. Sorgalla, and S. Sachweh, "Challenges of Domain-Driven Microservice Design," IEEE Softw., 2018.

G. Fersi, "Middleware for internet of things: A study," 2015, doi: 10.1109/DCOSS.2015.43.

Jamshidi, P., Pahl, C., Mendonca, N. C., Lewis, J., & Tilkov, S. (2018). Microservices:The journey so far and challenges ahead. IEEE Software. https://doi.org/10.1109/MS.2018.2141039

Luthria, H., & Rabhi, F. (2009). Service oriented computing in practice - An agenda for research into the factors influencing the organizational adoption of service oriented architectures. In Journal of Theoretical and Applied Electronic Commerce Research. https://doi.org/10.4067/S0718-18762009000100005

MacKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F., & Metz, R. (2006). Reference Model for Service Oriented Architecture 1.0. OASIS Standard. OASIS Open.

R. S. Bhadoria, N. S. Chaudhari, and G. S. Tomar, "The Performance Metric for Enterprise Service Bus (ESB) in SOA system: Theoretical underpinnings and empirical illustrations for information processing," Information Systems. 2017, doi: 10.1016/j.is.2016.12.005.

Singleton, A. (2016). The Economics of Microservices. IEEE Cloud Computing. https://doi.org/10.1109/MCC.2016.109

T. Erl, Governing Shared Services On-Premise and in the Cloud. Boston: Prentice Hall, 2011