Building a Fintech Ecosystem: Design and Development of a Fintech API Gateway

Ersin Ünsal
R&D Center
Fibabanka A.Ş.
İstanbul, Turkey
Ersin.unsal@fibabanka.com.tr

Bilgehan Öztekin
R&D Center
Fibabanka A.Ş.
İstanbul, Turkey
bilgehan.oztekin@finberg.com.tr

Murat Çavuş
R&D Center
Fibabanka A.Ş.
İstanbul, Turkey
Murat.Cavus@fibabanka.com.tr

Suat Özdemir
Computer Engineering Department
Gazi University
Ankara, Turkey
suatozdemir@gazi.edu.tr

***Abstract*— With the rise of financial technology startups (so called fintechs) and the adoption of open banking business models, the banking industry is witnessing revolutionary changes. To adopt these revolutionary changes, banks are collaborating with fintechs to develop innovative products and services for the customers. Banks, which are ruled by conventional practices, have been using very secure and closed computer-based systems whereas building a fintech ecosystem requires integration to countless 3rd party software services (so called APIs) in a secure and robust way. This paper discusses the design and development of a Fintech API Gateway in terms of business needs and technical challenges. Additionally, business requirements, technical design decisions, development challenges and key performance metrics of the API Gateway are discussed and explained in detail.**

Keywords—financial technology, open banking, API, PSD2

I. Introduction

In the last decade, the whole banking industry bore witness to revolutionary changes. Due to the collapse of the banking sector in 2008, customer confidence in the banking sector was lost on a large scale. In addition to that point, with the remarkable developments in smart phone applications, customer expectations changed significantly. At this point, financial technology companies emerged rapidly as third-party players, that aim to disrupt the conventional understanding of banking. On the other hand, many banks realized that this development is an opportunity for them to integrate innovative solutions quickly into their operations enabling better efficiency and customer services [1]. Also, Payment Service Directive (PSD2) introduced by the EU in 2013, asserts new proposals whose aims are basically to encourage innovation, transparency for the banking industry. Therefore, in this respect, banks started to adopt new policies and put some emerging technologies into practice with fintech companies' help.

Technological developments transform the banking industry deeply despite strict controls and regulations over the banking industry [2]. Therefore, in the last decade, banks started to act in this direction by investing on financial technologies. Customers expect banks to help them by providing decent financial goods and services or accumulating their financial assets [3]. To achieve these prominent objectives, banks have started to adopt open banking related policies and to collaborate with fintechs.

In this paper, the details of experience gained in developing a Fintech API Gateway are discussed. The presented API Gateway is developed in a nationally funded research project. The problem is rigorously discussed in terms of business needs and technical challenges and resulting benefits. Finally, comments on the future scenarios of Fintech API Gateways are given.

II. Building a Fintech Ecosystem and API Business Models

A. Open Banking and Building a Fintech Ecosystem

Sharing financial information on the conditions approved by customers through electronic media in a secure fashion is basically called open banking [4]. Technically speaking, third parties are provided with financial data with the help of the Application Programming Interfaces (APIs) [3]. Bank customers used to interact with only banks through internet or mobile banking applications. However, these new developments may result in involvement of third parties like fintech companies. New PSD2 implementations expecting banks to open their data to third parties and disruption of fintechs created a new competitive environment leading banks to share their data with third parties. With data shared and third parties' involvement, the open banking paves the way for new goods and services that could aid customers and businesses in various aspects [5].

United Kingdom (UK) is one of the leading countries in open banking. Competition and Markets Authority (CMA), which is a non-ministerial government department in the UK [6], issued a package of reforms concerning 9 British banks. This reform plan suggests that third parties registered in open banking system could direct access to data shared by these British banks. For instance, third parties could attain bank account information anymore. So, in order for customers and small businesses to be able to compare goods or services in the direction of their own needs without having to interact with their banks, UK Open Banking allows them to share their information with third parties in a secure way [3]. The developments just mentioned indicates that legislators both in UK and EU give importance to open banking which could

978-1-7281-5628-6/20/$31.00 ©2020 IEEE

lead decent fintech ecosystems to be built on a strong foundation.

B. *Conceptualization of APIs*

Having these developments, the banking sector began to open their data to third parties. How they share data is a crucial topic and needs to be investigated. Application Programming Interface also known as API steps in at this point. APIs could be defined as heart of open banking. In fact, concept of API should not be considered as emergent technology since its foundation was laid in the 20th century. However, its usage on the web environment has become more prevalent in the last two decades. It has become a popular phenomenon benefited in various areas such as social media, banking industry and e-commerce. Josh Walker, who works for Forrester Research Inc. which is a Cambridge based market research company, says "Building an application with no APIs is basically like building a house with no doors. The API is how you open the windows and doors of computing so that different software can exchange information and work together" [7].

When APIs are analyzed, it is seen that they are some coding interfaces providing a communication opportunity between different software systems, even if these software systems are coded in different programming languages. To put it differently, APIs allow us to connect systems through an agreed-upon protocol that is vendor and language neutral [8]. To be able to execute the functions mentioned in the previous sentence, APIs should provide a set of operations defined by inputs and outputs and they should allow re-implementation without compromising its users [9]. In order to clarify it, some analogy could be drawn between an API and a waiter [10]. So, the customer places an order from the menu, then the waiter delivers it to the kitchen. Eventually, the waiter brings the order to the customer. In this instance, the waiter symbolizes an API which has some functions such as operating with inputs and outputs. The customer is the program or programmer requesting information from one another, while the kitchen is the database containing intended data. Finally, the menu is the protocol which determines the rules for how to request the information. Besides, it should be noted that the waiter executes its operations without compromising the other entities.

C. *API Business Models*

With the developments related to API technology, new business models also have emerged. APIs are seen as monetization tools and there are many different options. IBM based article [11] exhibits these options in a straightforward fashion. These options are separated into four subgroups first of which is free APIs. Most of the worldwide known social media websites supply free APIs. They do not receive direct revenue for API calls; however, this is a benefit to these companies aiming to increase their number of customers. The second option is 'Developer pays' scenario in which a business asset is shared with an expense, like premium subscriptions. The third option, on the other hand, is 'Developer Gets Paid'. The API owner prompts the developer to leverage its API in that option. Car or travel comparison applications could serve as models. The last main option is indirect APIs aiming to achieve some goals that drives the business models. Integration of the two companies through APIs could exemplify indirect APIs.

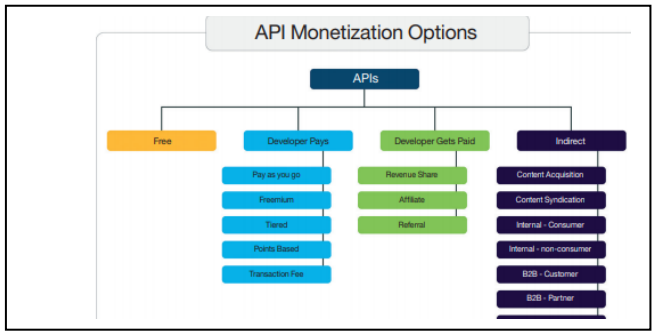

Fig. *1*. API monetization options. Adapted from 'API Monetization-Understanding your Business Model Options', by A. Glickenhouse and L. England, 2016, IBM, p.3.

D. *Banks, APIs and Fintech Applications*

As stated before, an API is a software interface that is able to interact with another which enables platforms to reveal their data to outer entities interested in utilizing it. APIs have become more and more popular tools provided the fact that they easily facilitate interactions among entities by removing the virtual borders. With that virtual visa liberalization, most of the operations on the web are conducted in a more agile and effective fashion. Also, PSD2 desires that any business - a social network, say, or a mobile app developer- can handle payments on behalf of its users. Merely the permission of users and access to the API requires [2]. So, in a way, PSD2 forces banks to reveal their data belonged to their customers to other entities to carry out transactions unless the customers desire the exact opposite situation.

With the help of the open banking, APIs, fintechs and exposed data, bank customers could easily handle their financial accounts and transactions. Bank customers are easily able to make purchases, receive their salaries, use mobile wallets, pay their bills or school fees and transfer their monetary assets [12] thanks to fintech applications. Providing customers with these innovative opportunities, banks would meet the customer expectations and enhance the customer experience which decrease the possibility of losing customers. Additionally, banks and partners (fintechs generally) can work together on conceiving joint product/service offerings [3] which enables banks to benefit from innovative fintech solutions directly. The other advantage of employing APIs is that the APIs could create new revenue channels due to the fact that banks have financial information of their customers and third parts could employ this information in aid of other sectors. For instance, some investors plan to open a new restaurant; nevertheless, they do not have decent information about what they will have on the menu of the restaurant. By acquiring spending information belonged to customers residing near the restaurant, the investors could choose what they will sell.

III. Challenges of Designing a Fintech API Gateway

APIs form a basis for open banking and its applications. Employing APIs leads customer satisfaction to derive and innovative financial solutions to prevail. With that said, the challenges of adopting APIs should be mentioned, especially

regarding security issues. First, while adopting API based policies fintech companies should accept the fact that they have to invest more money in constructing and administrating these APIs or gateways efficiently. Also, they should provide their customers with decent authentication process or protocols. Yet, under PSD2 directives and due to disclosing data, the financial institutions are still accountable for any fraudulent payments or events [2]. Authentication process is a major difficulty on its own considered the fact that if the institutions are specialized in being able to detect whether the customer handles the transaction. In other words, fraudulent transactions should be detected, prevented and undermined; thus, API and API gateway services supplied should be strictly secure. The other factor threatening this new open banking and API trend could be cyber intrusions. Since APIs are accessible easily, they form an attack surface for people desiring to profit by it for ill-intentioned purposes [2]. Although there are many techniques to detect and prevent intrusions, they might cause bigger attack surfaces at the end of the day.

IV. Fintech API Gateway System Components

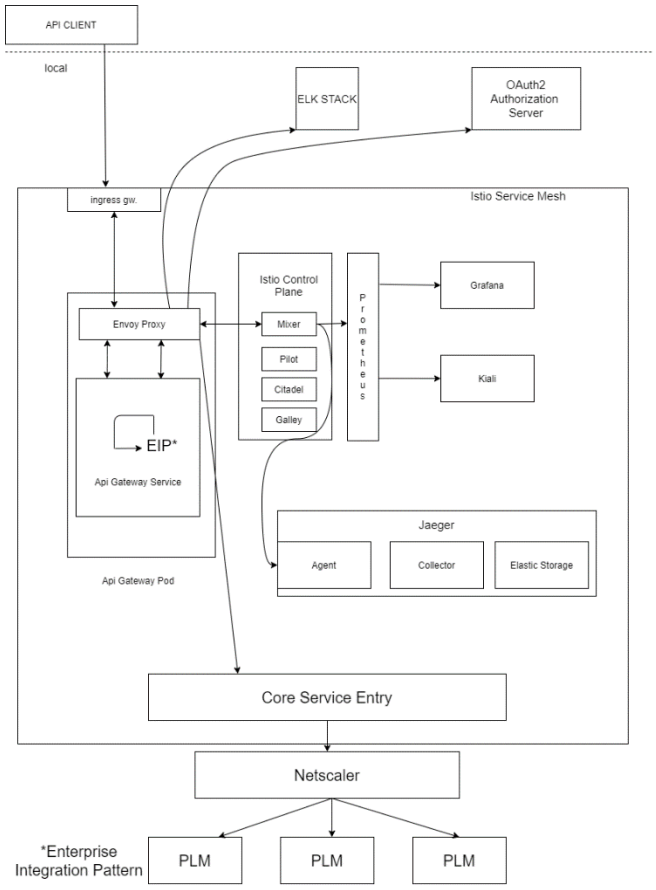

Fig. 2. Display of General System Architecture of Fintech API Gateway

A. General System Architecture of Fintech APIGateway

In this section, a cutting edge fintech API Gateway developed in Fibabanka R&D Center will be explained. The system could be roughly examined by dividing it into two basic parts such as outer environment and local environment of Fibabanka. While an API client, who delivers the request, is located in the outer environment, the other elements of the system are found in the local environment.

Service Mesh and API Gateway Pod could be described as underlying components of the local environment. While the former manages the traffic within the network, the latter is the receiver and internal distributor of the requests submitted. The other core part of the system is Product Lifecycle Management (PLM) tool which provides system with responses regarding requests.

B. Service Mechanism

A request is received from outer environment. At the service mesh stage, validity of the token and its authority to call for a service are checked. Then, API Gateway Service makes necessary transformations regarding the request. During this process of transformation, by using velocity template documents, API Gateway Service converts the format of the incoming request to a new format which could be operated by PLM.

API Gateway Service uses the component called Apache Camel which is located within. Thus, the API Gateway Service could completely benefit from standardized Enterprise Integration Pattern. In the next step, the request transformed is transferred to PLM through service mesh stage and PLM responds. Necessary transformation process for the response is executed by API Gateway Service. Again, by using velocity template documents, the service transforms the response into the format request is made. Finally, the response is returned to outside.

C. Design Decisions and Proposed Solutions

Technical requirements, design decisions and proposed solutions will be summarized for a better understanding of the Fintech API Gateway.

The first requirement was that Fintech API Gateway Service must operate identical at local environment and various deployment stages which are dev, test and prod. To be able to satisfy this requirement, container technologies have been employed. Thus, Docker [13], which has become de facto in this field, was preferred. Nonetheless, employing container methodology led another requirement that containerized application should not be administered manually. To undermine this problem, Kubernetes [14], which reveals best performance orchestrating container applications, was selected as an orchestration tool.

Other basic requirements such as obtaining metrics, L7 level traffic management, observability, rate limiting, fault injection used for testing, resiliency mechanism, policy stage and security layer must have been satisfied independently from the service application and also distribution mechanism must be enabled. To overcome these issues, the mesh technology was considered appropriate and Istio [15], which is one of the decent implementations of this technology, was determined to be employed.

Additionally, to be able to review the general situation of the system and the state of the service, system metrics must have been obtained. To satisfy these demands, it was decided that the distribution metrics provided by Istio would be obtained by using Prometheus which is a cloud native tool. Also,

metrics located on Prometheus [16] must have been visualized and system had to give alarm about the values indicated. It was seen that Grafana [17] having customizable dashboards, could be beneficial in visualizing these metrics on Prometheus.

Another requirement was that when a problem is encountered, the system must have been traced to be able to follow the tracks left by the request and the response. It was realized that the issue could be handled by using Jaeger [18] independent from the service. On the other hand, Jaeger needs to store the trace data obtained through Istio and memory should be allocated for that. However, the occupied memory would grow and should be persisted. This requirement should be met at cluster level and the stateless structure of the system should remain. To overcome these possible problems, it was noticed that Persistent Volumes [19], which are data consistency solutions located in Kubernetes, consolidated by Elasticsearch [20] tool could be put to use so as to satisfy the storage need. Hence, Persistent Volumes were decided to be used. Moreover, the needs for gathering knowledge of state of the services in Service Mesh field, analysis of Grafana dashboards and gathering jaeger trace data under a single roof emerged. Therefore, Kiali [21], which was developed specific to Istio Service Mesh, was employed as a problem solver. In addition to these requirements, distributed collection and visualization of logs and access logs recorded from service to standard output were needed. ELK Stack [22], which is specific, proven and previously used tool, was preferred.

Furthermore, set-up and maintenance of all tool applications should have been made by an employee which has to operate through package managers. Avoiding this operational workload and minimizing the human error factors must have been fulfilled completely. To achieve these objectives, it was agreed upon that the tools would be administrated by their own specific operators through operator lifecycle managers made specifically for these purposes. Another necessity indicating that Fintech API Gateway Service must fulfill the whole integration needs arose. For that purpose, not only Enterprise Integration Patterns [23] were acquired but also Apache Camel [24] including all prepared implementations on that field was employed.

API documents should have been prepared in order to inform API users about the outlook of the API structure. To this end, Swagger, which is a widely accepted API documentation tool among developers, was preferred for API documentation as a benchmark. It was determined that the developer who initiated a service would be responsible for the API documentation by inputting service-related data.

Another requirement which had to be satisfied was about data confidentiality in the context of security. So, services provided through core systems must have revealed only its desired fields. They should not have compromised all their fields. To be able to operate that transformation process, Velocity Templates [25] were acquired. Additionally, template organizing task was assigned to developers wishing to open their service from the core.

Finally, an authorization mechanism was required for security and policy issues. Also, this mechanism should have been implemented independent from the applications located in the cluster. So as to deal with these issues, OAuth2 [26] server application was coded using Spring Security Framework [27], which implements OAuth2 standards within the context of IETF [28] (Internet Engineering Task Force) standards.

V. Results/Discussion

The banking sector is generally seen as a static and strictly regulated industry which has many traditional roots. However, in the last decade, the banking sector has undergone unprecedented transformations. These revolutionary transformations were caused substantially by the fintech disruption emerged after the global economic crisis occurred in 2008. These emergent third parties' primary objective is to provide customers with effective and dynamic solutions supported by the financial technology. Also, recent regulations like PSD2, imposed by the EU and the CMA led banks to share their own data with third parties. This data sharing condition forms the basis of the open banking concepts. In order for the banking sector and the fintech ecosystem function in a coordinated manner, coding interfaces namely Application Programming Interfaces (APIs), which are able to provide a communication between the parties, require. APIs could connect systems coded in different coding languages through an agreed-upon protocol without compromising these systems.

Employing the API technology leads customers to have different type of customer experience. Thanks to the banks sharing their data with fintechs, customers could effectively handle their financial accounts or transactions. They could benefit from innovative solutions offered by fintechs. Not only customers, but also banks could gain advantages of fintech applications. For instance, banks and fintechs could work on projects aiming to produce joint products or services, which at the end results in enhanced customer satisfaction or expectation. Furthermore, this scenario would decrease the possibility of losing customers. Additionally, having the financial data of customers could provide banks with new revenue channels. With that said, there are some challenges concerning APIs. Constructing and managing APIs considering security and authentication issues are the main challenges a financial institution could face.

Considering all this positive and negative business factors, Fibabanka R&D Center developed a new Fintech API Gateway product to speed up the formation of a fintech ecosystem. During the design and development phases, some problems and requirements were faced and to be able to overcome them, some solutions and tools were employed. The addressed problems and requirements could be summed up as managing container applications, obtaining and visualization distribution metrics, tracing requests and responds, memory related issues, collecting access logs, integration related issues, confidentiality, security, authorization and documentation. Moreover, there may be some new technical issues which should be re-evaluated depending on the production results.

VI. Acknowledgment

This study is partially supported by the Scientific and Technological Research Council of Turkey under Grant no 3181420.

References

[1] Belli, M. “Banking and Fintech, Developing a Fintech Ecosystem in Istanbul, Learning Lessons from London”, BKM, İstanbul, Turkey, pp. 1-50.

[2] Mansfield-Devine, S. “Open banking: opportunity and danger”, Computer Fraud & Security, Volume 2016, Issue 10, 2016, pp. 8-13.

[3] Premchand,A. and Choudhry, A. "Open Banking & APIs for Transformation in Banking," 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2018, pp. 25-29.

[4] Pritchard, J. (2019, March 16), What Is Open Banking (and How Will It Affect You)? Retrieved from: https://www.thebalance.com/what-is-open-banking-and-how-will-it-affect-you-4173727

[5] “What Is Open Banking?” *Open Banking*, Retrieved from: www.openbanking.org.uk/customers/what-is-open-banking/

[6] “Competitions and Markets Authority” (2020, February 12). Retrieved from:

https://en.wikipedia.org/wiki/Competition_and_Markets_Authority

[7] Bly, R. “88 Money-Making Writing Jobs”, Sourcebooks, 2009, pp. 275.

[8] Lassoff, M. (2019, January 17), Understanding Application Programming Interfaces (APIs) Retrieved from: https://learningsolutionsmag.com/articles/understanding-application-programming-interfaces-apis

[9] Sandoval, K. (2018, September 20), Who Invented the API? Retrieved from: https://nordicapis.com/who-invented-the-api/

[10] Houghton, J. (2018, November 15), Understanding what APIs are all about. Retrieved from: https://medium.com/vody-techblog/understanding-what-apis-are-all-about-ff2513b76a55

[11] Glickenhouse, A. and England, L. “API Monetization-Understanding your Business Model Options”, IBM Cloud, New York, USA, April 2016, pp. 1-11.

[12] Doolhur, N. and Suddul, G. and Foogooa, R. and Richomme,M. "An open API to monetize mobile micro-services for emerging countries," 2013 Africon, Pointe-Aux-Piments, 2013, pp. 1-4.

[13] “Empowering App Development for Developers”, Retrieved from: https://www.docker.com/

[14] “Production-grade Container Orchestration”, Retrieved from: https://kubernetes.io/

[15] “Istio”, Retrieved from: https://istio.io/

[16] “Prometheus – Monitoring System & Time Series Database”, Retrieved from: https://prometheus.io/

[17] “Grafana: The Open Observability Platform”, Retrieved from: https://grafana.com/

[18] “Jaeger: Open Source, End-to-end Distributed Tracing”, Retrieved from: https://www.jaegertracing.io/

[19] “Persistent Volumes”, Retrieved from: https://kubernetes.io/docs/concepts/storage/persistent-volumes/

[20] “Open Source Search: The Creators of Elasticsearch, Elk Stack & Kibana”, Retrieved from: https://www.elastic.co/

[21] “Service Mesh Observability and Configuration”, Retrieved from: https://kiali.io/

[22] “What is the Elk Stack?”, Retrieved from: https://www.elastic.co/what-is/elk-stack?ultron=[EL]-[B]-[EMEA-General]-Exact&blade=adwords-s&Device=c&thor=elk%20stack&gclid=Cj0KCQjw6sHzBRCbARIsAF8FMpWZvfQsLPVu7rp49mewHumoqaHAG1VAg9XgXvYIIDyYE_DGDdz3VAAaAl_YEALw_wcB

[23] “Home – Enterprise Integration Patterns”, Retrieved from: https://www.enterpriseintegrationpatterns.com/

[24] “Home – Apache Camel”, Retrieved from: https://camel.apache.org/

[25] Retrieved from: https://velocity.apache.org/engine/1.7/user-guide.html

[26] “Oauth 2.0”, Retrieved from: https://oauth.net/2/

[27] “Spring Security”, Retrieved from: https://spring.io/projects/spring-security

[28] “Home”, Retrieved from: https://www.ietf.org/