On Open Gateway from GSMA – Is It a Revolutionary or Too Little and Too Late Deal?

Manfred Sneps-Sneppe
Ventspils University of Applied Sciences
Venspils, Latvia
manfreds.sneps@gmail.com

Dmitry Namiot
Lomonosov Moscow State University
Moscow, Russia
dnamiot@gmail.com

***Abstract*— In this article, we review a new GSMA initiative - Open Gateway. This is a set of programming interfaces for developers and service providers to access the infrastructure of a telecommunications operator. In its initial launch, this initiative includes 8 programming interfaces. They mainly focus on security. There are no software interfaces associated with voice calls at all, for example. The GSMA notes that this is only the beginning, and other APIs may follow. But in the first release, well-defined interfaces were chosen. How to evaluate this implementation? Will there be users at the new gateway? For what types of services (services) can this offer be suitable? Open Gateway did not appear out of nowhere. Telecommunications applications have been successfully developed for a long time without the participation of telecommunications operators. Will the new solution meet the needs of developers, and what may appear in subsequent implementations?**

I. INTRODUCTION

GSMA in February 2023 proposed a new initiative - Open Gateway [1]. The GSMA Open Gateway is an application programming interface (API) platform designed to provide universal access to carrier networks for developers. The idea is absolutely clear. This is the biggest attempt to create common programming interfaces for telecommunication services since Parlay/OSA [4]. The universal software interface will allow, in theory, to easily transfer services between different telecommunications operators, create common marketplaces, etc.

It is obvious that in the software world the idea of standardization and portability of programs dominates, while in telephony, for historical reasons, this was not the case. It is also obvious that the telecommunications world is moving towards standardization, since more and more services (services) are being implemented in software. In particular, cloud services offered by telecom providers are based on standard stacks. The same applies to working with big data.

According to the official press release, "GSMA's global Open Gateway initiative aims to leverage the expertise of the GSMA to support its members in unlocking and monetizing the power of telecom networks through open APIs and interoperability." Open Gateway, according to its name (and purpose), should serve as a gateway between services based on cloud infrastructure and physical telecommunications networks.

The main idea (task) of any standard API (and the GSMA is no exception here) is to speed up development and ensure application portability. This is exactly what the GSMA points out when it says that the gateway should remove the risk of conflicting conditions from different mobile operators for service developers. In this regard, we can say that standard APIs play an important economic role, making it cheaper to create and launch (operate) new services.

An important point is who this initiative is aimed at. This is important because it is depending on the addressee that specific APIs should be determined. On the GSMA website, cloud service providers and application developers, content and game developers, and software developers are the first to be mentioned. This means in particular that the GSMA proposal will have to compete with current development tools.

Other direct and indirect mobile industry customers targeted by the new gateway include regulators and corporate customers (and there are already a large number of existing applications and development tools).

The initial release includes eight GSMA open gateway network APIs: device location, device status, EDGE site selection, number verification, OTP verification, carrier billing, quality on demand, and SIM replacement verification. It is these services that are published in the CAMARA repository [2]. Notice here that there were 20+ specifications in Parlay/OSA.

Applications, according to the GSMS vision, may include identification, cybersecurity, billing, signaling, and geolocation. Separately, the control of unmanned aerial vehicles (identification, EDGE calculations) is noted. The scope of the API for geolocation GSMA sees gaming applications.

Telecom operators have been working on this initiative since February 2022. Then 21 mobile operators signed a memorandum of understanding. The first demonstrations of the Open Gateway initiative were launched at MWC23 in Barcelona, and the first real deployments of the Open Gateway APIs on carrier networks are planned for the third quarter of 2023 [3].

The only source of information about future plans is the FAQ on the GSMA website [3]. According to this document, more generic APIs will emerge during 2023-2024, creating an inclusive environment for scaling based on open technical standards.

What happened in February 2022 that led to the opening of this project? It seems to us that the final opinion has been formed that without third-party developers, no "killing applications" for 5G (and now for 6G) will happen. In our opinion, this is some kind of attempt to "save" investments in 5G by providing new applications, making this expensive infrastructure interesting for a large number of users, and, accordingly, improving the economics of the project. So in another way, the proposed gateway could be called the northern interface for 5G.

But here comes the next question. In the 20 years that have passed since the Parlay/OSA implementation attempt, telecommunication services and services have been successfully developed and implemented without the participation of operators. And APIs exist at many different levels of abstraction. Suffice it to say that open programming interfaces are fairly straightforward for VOIP systems. These are purely software implementations, and the issue of public interfaces is simply a matter of making available (exporting) certain program procedures (methods). Will the GSMA proposal be sufficient and successful? On the other hand, the tasks of telecom have changed over the past 20 years. For example, during the Parlay/OSA, there was no task of controlling (monitoring) drones [9].

Technically, the authors of the article were participants in the first attempts to implement Parlay / OSA by Erickson in Russia and the Baltic countries [5, 6]. As well as participants in many projects for the implementation of telecommunications APIs [7, 8]. Therefore, the development of this topic is more than just interesting for us.

The rest of the article is structured as follows. In section II, we look at the general landscape for standard APIs. Section III deals with the previous attempts to implement the API. Section IV is devoted to the current state of software APIs. In section V we talk about 8 APIs proposed by the GSMA. Section VI is devoted to the discussion.

II. On the basis of building open APIs

Today we are seeing significant investment in 5G infrastructure. For example, a frequency band auction raised $81 billion in the US [10]. But it is also clear that we are not yet seeing 5G services in everyday life, at least not to the extent expected. No "killer app" for 5G has ever emerged.

Most of the work on 5G usually describes the network in terms of its capabilities. That is, 5G means high bandwidth, low (ultra-low) latency, many connected devices, high reliability, etc. Neither of these explanations describe the actual user experience with 5G. These are the primary concerns of the operators. And in such explanations, 5G is just a new radio system. In this state, it is very difficult to justify the investment in this system.

An Ericsson study found that mobile consumers are willing to pay 20% to 30% more for 5G plans bundled with new apps and services. But about 70% of current 5G users are unhappy with the lack of new and innovative applications [11]. In other words, they are waiting for new applications. But to create applications, you need as much input from developers in this process as possible. No operator, no consortium of operators, can anticipate all user needs. The application infrastructure is developed by the developer community through marketplaces. Apple and Google have proved this perfectly.

Architecturally, the place for software APIs is provided by the structural division of telecommunication systems into logical levels. The SS7 ITU standardization in 1980 marked the beginning of the development of the concept of an intelligent network [12]. IN includes a set of nodes that rely on widespread call processing features and capabilities. Before the advent of the IN model, all services (services) were available only through the local telephone exchange (local node). With IN, you can centrally host a certain service (service) on a specific node (for example, SCP), and then any switch can connect to it and use this function. The original purpose of smart grids was to facilitate the introduction of new services into the grid. IN has created the basis for complex services that are needed throughout the network. Fig. 1 depicts the proposed levels of IN.

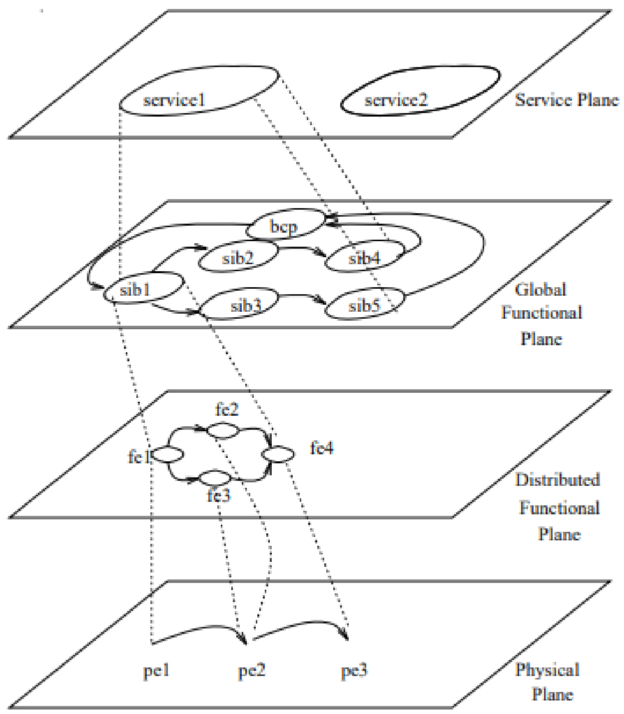

Fig. 1. IN planes [13]

Why is it important? The model defines the events that occur in the system. And the meaning of the API for service developers should be to customize the reaction to these events. This is exactly what is needed to build services (services) in a telecommunications network - the ability to determine your own reaction to system events: reactions to an incoming call, transmitted message, etc. [13].

In the mid-1980s, the second generation of IN - AIN (originally Bellcore) appeared. AIN takes the service logic outside the switch to an independent SCP (Fig. 2).

AIN's service-independent network architecture allows carriers to create and maintain telecommunications services and functions in a consistent manner using a common architectural platform. The goal is to ensure the rapid creation of customized telecommunications services [14]. In relation to the possible software architecture, it means much more points of interaction with developers.

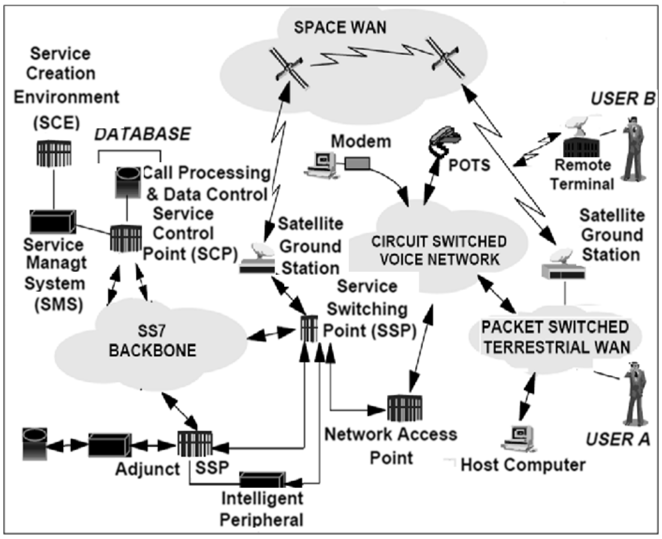

Fig. 2. AIN example [14]

Unfortunately, the right architectures did not lead to the next step - reflecting this model in software APIs. The lack of open interfaces has been one of the major problems with the IN. This gap left the concept of IN with no future.

In reality, only SDN (as a programming model) has provided access to its model in the form of an API [15, 16]. We note here that even in relatively recent architectures, the need to support third-party developers is not observed. For example, Fig. 3 depicts the D2D model [28, 29] in 5G (ProSe – proximity services). In it, we marked the points where it was necessary to describe the possibility of interaction with third-party programs. For example, a third-party application should be able to take control when a device is nearby, when a message is received from another device, and so on. None of this is in the current D2D specification [30], although this feature is specific to 5G and could serve as a basis for unique (previously absent) applications.

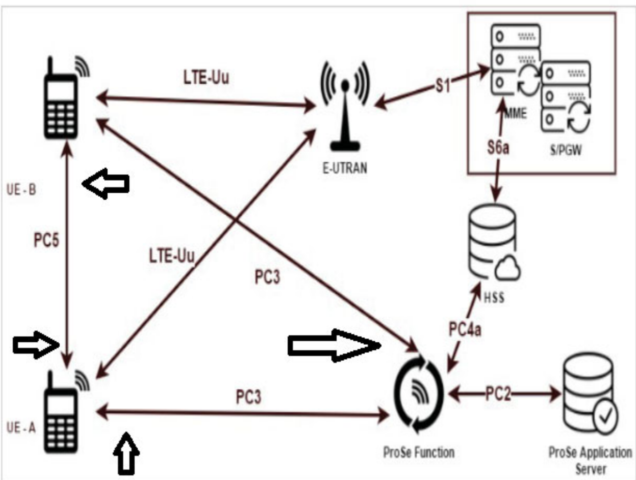

Fig. 3. On ProSe model and possible callback interfaces [17, 18, 13].

III. On Parlay/OSA open interfaces

Parlay was one of the most significant ways to introduce a single API for building telecommunication services [4, 19]. The first rides were based on the OMG Corba. Architecturally, this was the right decision - a third-party service made remote calls, and possible problems with the service did not affect the telecommunications provider (operator) in any way. But the implementation ended up being very cumbersome, making programming really difficult. The Parlay X specification was born, which was a set of standard web service APIs for the landline and mobile phone network that allowed software developers to take advantage of the capabilities of the underlying network. The interfaces themselves were designed at a high enough level of abstraction to make them easy to use: a single request to a web service to get a location, make a call, and so on.

The specification was developed by ETSI, Parlay Group, and 3GPP. In keeping with the times, APIs were defined using WSDL 1.1 and corresponded to the interaction of web services according to the WS-I profile. It quickly became clear that WSDL and the general approach to building web services did not live up to expectations, you need to move to REST and replace XML with JSON. But all this is already the history of the (unsuccessful) implementation of this API. In the context of this consideration, we are interested in which APIs were included in this specification. Here's the full list, with bold type highlighting what's (to some extent) in the new GSMA proposal:

Part 1: "Common" - Definitions and vocabulary used across all Parlay X specifications

Part 2: "Third Party Call" - Creating and managing a call initiated by an application

Part 3: "Call Notification" - Handling calls initiated by a subscriber in the network. Third-party applications can either forward (process) calls or simply receive notifications about them

Part 4: "Short Messaging" – Programmatically receive and send SMS

Part 5: "Multimedia Messaging" – Programmatically receive and send Multimedia Messages

Part 6: "Payment" - Pre-paid payments and post-paid payments in third-party applications

Part 7: "Account Management" - Account querying, direct recharging, and vouchers based recharging

Part 8: "Terminal Status" - Programmatically request the status of a terminal

Part 9: "Terminal Location"- Programmatically getting location information about a terminal

Part 10: "Call Handling" - Specify how calls are to be handled for a specific number.

Part 11: "Audio Call" - Provide multimedia message delivery and the dynamic management of the media involved for the call participants

Part 12: "Multimedia Conference" - Create a multimedia conference and the dynamic management of the participants involved

Part 13: "Address List Management" - Programmatically manage groups (aliases) of subscribers

Part 14: "Presence" - Presence information for users

Part 15: “Message Broadcast” - Programmatically send messages to all the fixed or mobile terminals in a specified geographical area

Part 16: “Geocoding” - Get the location address of a subscriber e.g. country, state, district, city, street, house number, additional information, and zip/postal code

Part 17: “Application-driven Quality of Service (QoS)” - Dynamically change the quality of service (e.g. bandwidth) available on end-user network connection

Part 18: "Device Capabilities and Configuration" - Programmatically request information about device capabilities and push device configuration to a device

Part 19: “Multimedia Streaming Control” - Programmatically control streaming of multimedia to a subscriber

Part 20: “Multimedia Multicast Session Management” - Programmatically control a multicast session, its members, and multimedia stream. Programmatically obtain channel presence information

Part 21: "Content management" - The content management web service enables uploading content into the network (or a third-party content provider) and consuming content from the network (or a third-party content provider).

Part 22: "Policy" - The Policy Web Service is defined to offer provisioning and evaluation functions for policies.

The Parlay/OSA API was never adopted by the developer community because the level of abstraction was too low. In fact, Parlay/OSA did not meet the main requirement for software tools - it did not save development time. A simplified version of Parlay X was adopted, which was based on the newly introduced web services, but it only supported a limited number of APIs at all.

Some specifications are no longer relevant today (e.g. Geocoding), but, in general, the new proposal from GSMA is much smaller (yet?) than the old Parlay/OSA specification. It can be noted the complete absence of actual telephone functions (calls, messages).

IV. ON THE CURRENT PRACTICE OF TELECOM APIS

For users (both end users and business users), an integrated communications platform is more interesting today than “classic” telephony. Therefore, in addition to the actual phone APIs, in 2023 it makes sense to consider higher-level software solutions. Namely, CPaaS (Communication Platform-as-a-Service, communication platform as a service) is a model for providing a ready-made software environment and tools for communicating with customers or employees of the customer’s company [20]. As well as CDP - Customer Data platform [21]. CDP is software that collects and combines data from multiple touchpoints to create a single, centralized customer database containing data on all touchpoints and product/service interactions.

Typically, CPaaS includes at least the following APIs [22]:

- Messaging — SMS/MMS/toll-free
- Messaging — Over-The-Top (OTT) channels (like WhatsApp and Messenger)
- Messaging — conversations (multi-party, multichannel)
- Voice calling — VoIP, PSTN, & SIP trunking
- Video calling — P2P, group rooms, & interactive live streaming
- Email — transactional & marketing emails
- Low-code/no-code — visual builders
- Verification/identity — one-time passcodes, strong customer authentication, line-type lookup

Fig. 4 provides an overview of the capabilities of CPaaS [23].

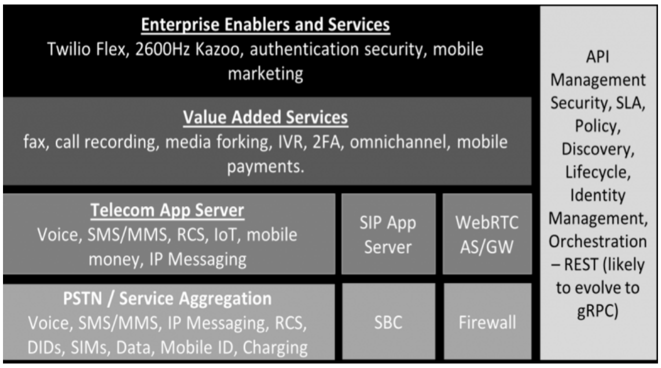

Fig. 4. On the structure of CPaaS [23]

There are already quite a few vendors in this market segment. One of the clear leaders is Twilio [24].

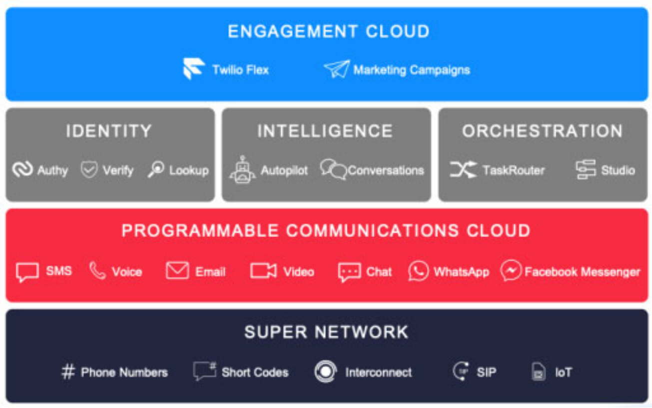

Fig. 5. Twilio [24]

Multi-access Edge Computing (MEC) offers developers and content providers access to APIs at the edge of the network. This can provide ultra-low latency and high bandwidth as well as real-time access to radio network information. MEC architecture is formed with two levels that are deployed along with the BS and the mobile core network entities referred as the edge/host level and the system level, respectively [27]. Technically, this is a REST API. But of the actual “telephone” APIs, only the Location API (ETSI MEC ISG MEC013) can be mentioned there. This API is (at least at the specification level) more advanced than the GSMA proposal (it describes more functions), for example. But the question remains again in practical implementation and the need, in this case, for a separate component (edge)

V. On GSMA Open Gateway

What exactly does the GSMA offer in its APIs? The current list includes the following APIs:

Sim swap

The API checks the last modified time of the SIM card associated with the mobile phone number (MSISDN). The response can be a timestamp or a boolean (yes/no) for a specific period (for example, the last 24 hours).

The application is obvious: cybersecurity, combating fraud in payment systems

Quality On Demand

The API allows an application to request a quality level for its data streams. Quality can be defined in terms of, for example, latency, jitter, or bandwidth.

Possible applications: The query checks if the network can provide the required level of quality. This is necessary, for example, in projects related to a remote control. As with some other APIs, the GSMA mentions drones here.

Device status

The API checks if the device is online and roaming.

Possible applications: IoT device connectivity monitoring throughout the life cycle of the device (e.g. activation, setup, remote operations, software updates, troubleshooting), anti-fraud (e.g. checking if the device is roaming when withdrawing money in another country).

Number Verify

The API requests that the mobile device be authenticated by the mobile network.

Possible applications are obvious: cybersecurity models. E.g. mobile phone ownership check. Can be used, for example, to enter applications, instead of requiring a login and password.

Edge Site Selection and Routing

The API allows the app to discover the closest Edge-Cloud host that the device can connect to. In practice, a special Edge Discovery protocol should still be used here.

Possible applications are - anywhere Edge cloud is required: mixed/augmented reality, HD video streaming, cloud gaming, and remote control of moving objects or vehicles. Drones fit in here too.

Number Verification (SMS 2FA)

The API delivers the shortcode to the mobile device via SMS. It is well-known two-factor authentication.

Possible applications are obvious: cybersecurity models. SMS One Time Password is used to verify that the user owns the mobile device associated with the mobile phone number used to connect.

Carrier Billing – Check Out

The API allows an online merchant to request payment from the user's mobile carrier. This is the implementation of payment for goods (services) using the owner's account with the telecom operator. The payment amount is added to the user's phone bill or subtracted from the user's prepaid balance.

The usages are obvious: mobile payments in e-commerce, games, etc.

Verify Location

The API allows an application to check if a mobile device is in close proximity to a given location. The phone number (MSISDN), the location itself, and the accuracy of the determination, as well as the maximum time elapsed since the determination of the location are indicated. The last parameter allows you to determine how long ago the location was measured.

The meaning of this API is to check the location of the mobile device. In our opinion, this will be one of the most demanded areas. Technically, these are geo-grid services. With regard to practical applications, indoor geo-fence will be of interest. And there are geo-marketing applications. For example, the application of some store (merchant chain) will be able to check if their user (subscriber) is inside some physical location (near it) in order to send a push notification or SMS with a trade offer.

Yes - mobile apps can confirm GPS location, but this introduces the problem of working indoors (where this option is most interesting), as well as the recently worsened problem of GPS spoofing. Accordingly, such an API will make it possible to use various trackers equipped with SIM cards, which will affect transport applications, logistics, and drones.

For security applications, the use of such an API will allow tracking of the presence of objects (objects) in a given area. The location of the user (mobile device) is used in anti-fraud programs of banks (payment systems).

Today, the listed services already have solutions without telecommunications operators. The use of GPS has long been not the only way to determine the location. Locations (coordinates) can be calculated from wireless network signals, location in services can generally be replaced by an estimate of network spatial proximity from the same wireless network signals, etc. Nevertheless, there is no single approach here, and a solution from a telecom operator will obviously be in demand.

Let us use this API as an example, you can see how it is technically organized (the rest APIs are organized similarly).

This is a REST API. Request - HTTP POST, where payload contains JSON with request parameters:

```
curl -X 'POST' https://sample-base-
url/location/v0/verify
-H 'accept: application/json'
-H 'Content-Type: application/json'
-H "Authorization: Bearer
eyJ0eXAiOiJKV0eXAiOi..."
-d '{ "ueId": { "ipv4addr":
"95.165.30.107" },
"uePort": 8081,
"latitude": 55.755821,
"longitude": 37.617356, "accuracy": 4 }
```

The response is HTTP status and payload in JSON format. For example:

```
200
{ "verificationResult": "TRUE" }
```

VI. On discussion

The technical simplicity of the proposed APIs is beyond doubt. But the question of using any API remains dependent on the organization providing access. How will the work of telecommunications operators with the developer community be organized? How easy will it be to access the API? It can be recalled that Parlay/OSA was supposed to be implemented according to the same scheme, as a result, none of the operators wanted to be the first in this process.

In the current list of APIs, security issues occupy the first place. It is interesting, that the actual telephone API (if we mean voice communication) is not presented at all. Is there evidence of a change in the role (importance) of telecom applications, or will it simply be supported in the next steps?

Note that if the main purpose of the new API is to support applications for 5G (or already for 6G), then voice communications are not at all the main thing there. 5G brings nothing new to voice communications. But, for example, 5G defines D2D interactions. It might make sense to focus on those APIs where there might be something new compared to the existing offerings.

As can be seen from the review, the new API from GSMA has many competitors. The current capabilities of existing APIs are much larger than the GSMA proposal. And these APIs already have their users. Accordingly, an important point will be the support of developers when implementing new APIs. Such support is needed both at the technical level (conditionally - Stackoverflow, etc.), and at the methodological level - what developers can get from the new API in general.

VII. Conclusion

In its current form, the open API from GSMA presents a rather small range of possibilities for third-party developers. First of all, for cybersecurity tasks (anti-fraud in payment systems, etc.). Of greatest interest is the API for testing location determination, since there are different approaches to determining the location on premises and there is no single leader (as in the case of GPS). Otherwise, the current offering from the GSMA is a very niche product. Considering that the deployment of even such a minimal set of calls is expected only at the end of 2023, the release of this API will not yet have a significant impact on the telecommunications services development market.

References

[1] GSMA Open Gateway https://www.gsma.com/futurenetworks/gsma-open-gateway/ Retrieved: Mar, 2023.

[2] CAMARA https://github.com/camaraproject. Retrieved: Mar, 2023.

[3] GSMA Open Gateway Frequently Asked Questions https://www.gsma.com/futurenetworks/gsma-open-gateway-frequently-asked-questions/ Retrieved: Mar, 2023

[4] Moerdijk, A-J., and Lucas Klostermann. "Opening the networks with Parlay/OSA: standards and aspects behind the APIs." *IEEE network* 17.3 (2003): 58-64.

[5] Sneps-Sneppe, Manfred, and Dmitry Namiot. "About M2M standards and their possible extensions." *2012 2nd Baltic Congress on Future Internet Communications*. IEEE, 2012.

[6] Sneps-Sneppe, Manfred, and Dmitry Namiot. "M2M applications and open API: What could be next?." *Internet of Things, Smart Spaces, and Next Generation Networking: 12th International Conference, NEW2AN 2012, and 5th Conference, ruSMART 2012, St. Petersburg, Russia, August 27-29, 2012. Proceedings*. Springer Berlin Heidelberg, 2012.

[7] Namiot, Dmitry, and M. Sneps-Sneppe. "On Data Program Interfaces." Journal of ICT Standardization (2015): 269-288.

[8] Namiot, Dmitry, and Manfred Sneps-Sneppe. "On software standards for smart cities: API or DPI." *Proceedings of the 2014 ITU kaleidoscope academic conference: Living in a converged world-Impossible without standards?*. IEEE, 2014.

[9] Ullah, Hanif, et al. "5G communication: An overview of vehicle-to-everything, drones, and healthcare use-cases." *IEEE Access* 7 (2019): 37251-37268.

[10] Verizon, AT&T and T-Mobile dominate $81 billion 5G spectrum auction https://www.cnet.com/tech/mobile/verizon-at-t-t-mobile-dominate-81-billion-5g-spectrum-auction/ Retrieved: Mar, 2023

[11] Facing Today's 5G Issues and Challenges/ https://www.jabil.com/blog/5g-challenges.html Retrieved: Mar, 2023

[12] Bahl, Manu, J. Daane, and R. O'Grady. "Evolving intelligent interexchange network-an SS7 perspective." *Proceedings of the IEEE* 80.4 (1992): 637-643

[13] Namiot, Dmitry, and Manfred Sneps-Sneppe. "On one d2d usage model for 5g networks." *2021 28th Conference of Open Innovations Association (FRUCT)*. IEEE, 2021.

[14] Sneps-Sneppe, Manfred, and Dmitry Namiot. "On telecommunication architectures: from intelligent network to network functions virtualization." *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2016.

[15] Zhou, Wei, et al. "REST API design patterns for SDN northbound API." *2014 28th international conference on advanced information networking and applications workshops*. IEEE, 2014.

[16] Sneps-Sneppe, Manfred, and Dmitry Namiot. "Metadata in SDN API for WSN." *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2015.

[17] 3GPP TS 24.334 V12.1.1, December 2014; Technical Specification Group Core Network and Terminals, Proximity-services (ProSe) User Equipment (UE) to ProSe function protocols; Stage 3; Release 12.

[18] LTE Direct http://www.3glteinfo.com/lte-direct/ Retrieved: Mar, 2023

[19] Mulligan, Catherine. "Open API standardisation for the NGN platform." *2008 First ITU-T Kaleidoscope Academic Conference-Innovations in NGN: Future Network and Services*. IEEE, 2008.

[20] CPaaS and the API Ecosystem Help Telecom Deliver Value to the Enterprise https://www.telestax.com/cpaas-api-ecosystem-help-telecom-deliver-value-enterprise/ Retrieved: Mar, 2023.
[21] Earley, Seth. "The role of a customer data platform." *IT Professional* 20.1 (2018): 69-76.
[22] What is CPaaS guide https://www.twilio.com/resource-center/what-is-cpaas-guide Retrieved: Mar, 2023.
[23] CPaaS segmentation https://alanquayle.com/2019/05/cpaas-segmentation/ Retrieved: Mar, 2023
[24] Twilio CPaaS https://www.twilio.com/ Retrieved: Mar, 2023
[25] Multi-access Edge Computing (MEC) https://www.etsi.org/technologies/multi-access-edge-computing Retrieved: Apr, 2023
[26] ETSI GS MEC 013 - Location API ttps://forge.etsi.org/rep/mec/gs013-location-api Retrieved: Apr, 2023
[27] Ranaweera, Pasika, Anca Jurcut, and Madhusanka Liyanage. "MEC-enabled 5G use cases: a survey on security vulnerabilities and countermeasures." *ACM Computing Surveys (CSUR)* 54.9 (2021): 1-37.
[28] Liu, Xiangyan, et al. "A novel D2D–MEC method for enhanced computation capability in cellular networks." *Scientific Reports* 11.1 (2021): 16918.
[29] Pencheva, Evelina Nikolova, and Ivaylo Ivanov Atanasov. "Mobile edge service for d2d communications." *2018 IEEE XXVII International Scientific Conference Electronics-ET*. IEEE, 2018.
[30] D4.2 Specification of remote and D2D protocol and APIs for HR exchange https://www.interopehrate.eu/wp-content/uploads/2020/10/InteropEHRate-D4.2-Specification-of-remote-and-D2D-protocol-and-APIs-for-HR-exchange-V2.pdf Retrieved: Apr, 2023