Utilization of APIs for B2B2X Business Model

Mayumi Takahashi

Abstract

The NetroSphere concept proposed by NTT laboratories aims to divide network functions into small modular components and to flexibly assemble those components. This makes it possible to quickly form proportional networks that can offer required functionality according to user needs. As a result, carrier networks will achieve enhanced flexibility and elasticity while also drastically reducing costs. This article introduces the architecture to provide network functions and resources via APIs (application programming interfaces) to collaboration partners such as service developers.

Keywords: API, collaboration, NetroSphere

1. Introduction

The NetroSphere concept is a future network vision proposed by NTT laboratories [1]. The concept enables network functions to be divided into small modular components, which can then be flexibly assembled. This makes it possible to quickly form proportional networks that can offer required functionality in accordance with the user's needs for B2B2X (business-to-business-to-X) business models where various service partners and communication providers collaborate to provide services. This will increase the flexibility and elasticity of carrier networks while also drastically reducing costs.

Web-API[*1] (API) is considered to be one of the key technologies to achieve the NetroSphere concept. In this article, the architecture that provides network functions and resources via APIs and design rules for the APIs are explained.

2. Trends in usage of APIs for collaboration

2.1 Trends of service providers and overseas telecom operators

Well-known service providers such as Twitter, Facebook, and Amazon Web Service expose APIs in order to provide their own functions and information. Some large overseas telecom operators have also started providing network functions such as call control and payment services and have organized promotional programs such as ideathons and hackathons to bring business ideas into shape [2, 3].

Our view is that providing network functions via APIs will bring two advantages to these service providers and telecom operators:

1) Growth in revenue: When they provide functions and resources via APIs, collaboration partners can create new services with those functions and resources. Service providers and telecom operators can earn revenue by charging for the use of functions and resources that are provided via APIs.
2) Reduction in operating expenses (OPEX): Providing operational functions via APIs simplifies operational cooperation with collaboration companies. It permits business process automation and reduces OPEX.

2.2 Trends in standardization

Mobile telecom operators provide network functions via APIs to their collaboration partners. The

[*1] Web API: An API (application programming interface) is an interface to exchange information between software components. Web API is one kind of API. Via an API, an application can call external functions provided by other software components or applications.

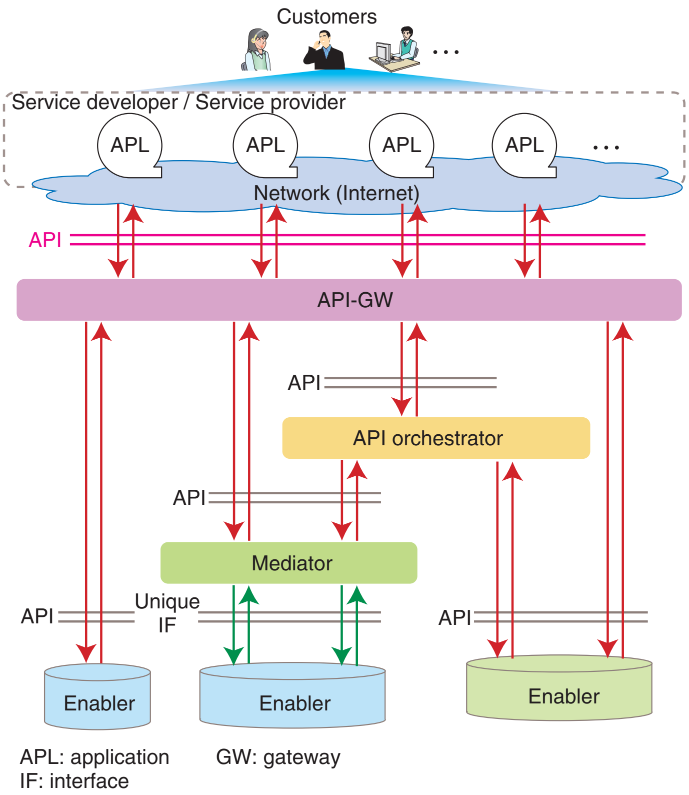


Fig. 1. API-providing architecture.

specifications of these APIs differ from operator to operator. This forces collaboration partners to write operator-specific programs. OneAPI is a set of APIs supported by the GSM Association (GSMA)[*2] that defines API specifications for elements such as SMS (Short Messaging Service)/MMS (Multimedia Messaging Service), device capabilities, and payments [4]. AT&T provides some APIs according to these specifications.

The TeleManagement Forum (TM Forum)[*3] has started to define API specifications for the interface between these business processes in order to facilitate the automation of business processes of telecom operators. TM Forum published some operational specifications such as those for trouble tickets and product ordering. Further, in the Open Digital Project, one of TM Forum's strategic programs, discussions have been initiated on the reference architecture called the Digital Service Reference Architecture (DSRA), which allows collaboration partners to create applications with a number of digital services provided via APIs [5].

As previously mentioned, some standardization organizations have defined the specifications of APIs, but their scope is limited. Thus, most operators provide APIs designed according to their own rules.

3. Technical considerations in providing network functions via APIs

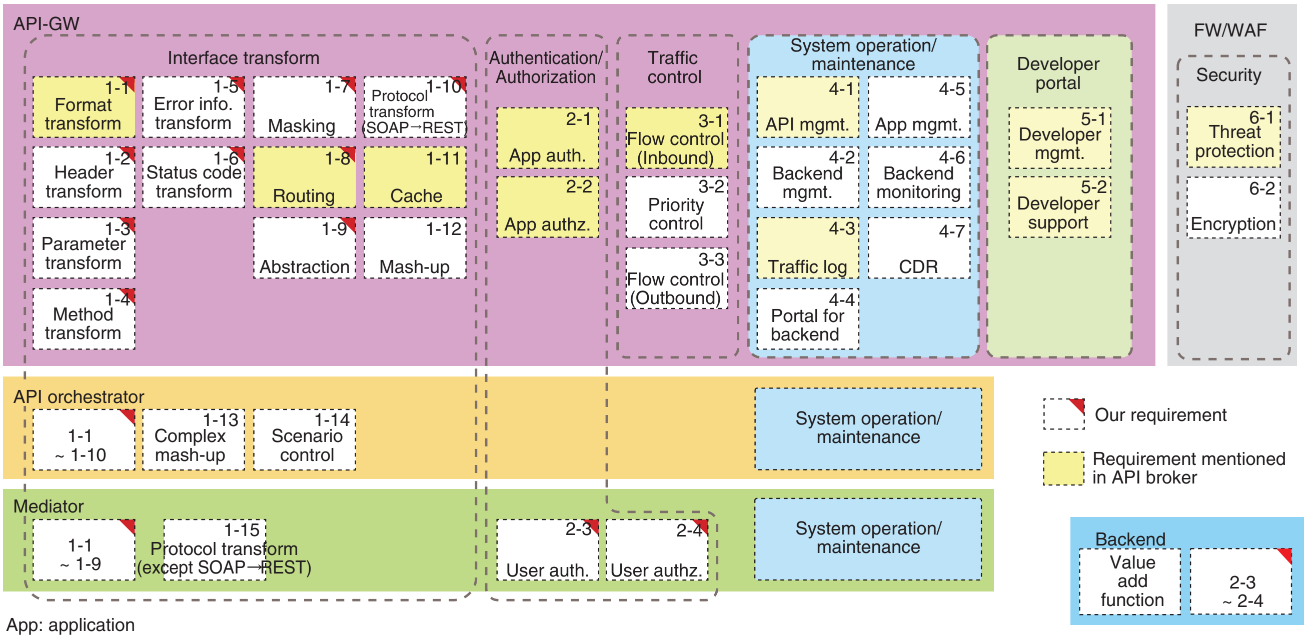
We consider that APIs represent one of the key technologies to achieve the NetroSphere concept and that they will be effective in providing the architecture for NetroSphere.

3.1 API-providing architecture

The architecture being studied to provide network functions via APIs is shown in **Fig. 1**. This architecture consists of three functional units: the mediator

[*2] GSMA: GSMA is an association of mobile operators and related companies devoted to supporting the standardization, deployment, and promotion of the GSM mobile telephone system.

[*3] TM Forum: TM Forum is a non-profit industry association for service providers and their suppliers in the telecommunications and entertainment industries. TM Forum defines reference models of interfaces between business processes and data models.



App: application
CDR: clock and data recovery
FW: firewall
SOAP: Simple Object Access Protocol
WAF: web application firewall

Fig. 2. Function deployment.

unit for protocol transformation, the API orchestrator unit for scenario control and API mash-up, and the API-GW (gateway) unit for authorization and traffic control. The details of these functions are described here and shown in **Fig. 2**.

(1) Mediator unit

Many existing operational systems have a unique interface to enable interconnection with other such systems. Our view is that it would be advantageous for network functions to be provided via representational state transfer (REST) APIs because the REST API style is familiar to service developers. Thus, we introduced the mediator unit to transform the unique interface into the REST API style.

(2) API orchestrator unit

Telecom operators do not want to directly expose their own backend systems such as their customer management and billing systems from the viewpoint of protecting the security of their systems and avoiding unnecessary complexity. The API orchestrator unit provides API abstraction and scenario control functions, which meet telecom operator requirements.

(3) API-GW unit

Service developers ordinarily have to call each end-point that differs from system to system when they use network functions provided by different systems. Moreover, each system may provide different API specifications such as those for the data format. This is not helpful to service providers. Thus, we introduced the API-GW unit to centralize end-points and absorb differences in API specifications. In addition, the API-GW provides a common function to expose APIs such as those for traffic control and authorization. The API-GW unit is divided into three functional units that have specific functions.

1) API broker: API broker provides a security protection function, authentication and authorization function (API-key, OAuth), and a traffic control function in accordance with the terms of agreement and system state, and an interface transformation function in accordance with design rules described in section 3.2.
2) Developer portal: The developer portal provides service developers with a management function and documents such as manuals and sample code.
3) System operation/maintenance: The system operation/maintenance function unit provides operational functions for the API-GW operator to manage enablers, APIs, and applications developed by service developers and also to

manage the state of the API-GW.

3.2 API design rules

The design rules of REST APIs such as the API naming rule and data format description rule are flexible. Thus, specifications of APIs provided by telecom operators may differ from operator to operator. In addition, APIs may differ from vendor to vendor even if the functions are the same because these APIs are designed according to vendor-specific design rules. This may cause difficulties when telecom operators renew their equipment.

To simplify things for service developers, we believe that telecom operators should provide APIs designed according to a unified design rule based on one of the two rules below.

1) Function-independent rules: These rules define common rules that apply regardless of the type of functions; they include naming rules of URLs, rules for determining data format, and authentication methods of APIs.
2) Function-dependent rules: These rules define functions themselves such as those for call control, service orders, and trouble tickets.

Our first step will be to focus on function-independent rules.

4. Future prospects

This article reviewed the API-related trends followed by overseas telecom operators. An architecture that provides network functions via APIs was also discussed along with API design rules. Our aim is to standardize the proposed architecture and describe the API design rules at a TM Forum event. We will define the aforementioned architecture that allows service developers to create applications with multiple functions provided by telecom operators.

In parallel, we will start providing an API testbed called Atelier N [6], which is positioned as a collaborative program. We would like to use this program to discuss issues related to creating services and providing functions with our collaboration partners such as service developers and function providers. We also plan to evaluate the proposed architecture and design rules with our collaboration partners.

References

[1] R. Sumi, K. Matsumoto, I. Inoue, and Y. Kajiyama, "The NetroSphere Concept and Network Architecture," NTT Technical Review, Vol. 13, No. 10, 2015.
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201510fa1.html

[2] AT&T, Developer Program, http://developer.att.com/

[3] Website of Orange Partner, https://www.orangepartner.com/

[4] GSMA, API Exchange, http://www.gsma.com/oneapi/

[5] Website of TM Forum, https://www.tmforum.org/

[6] Website of Atelier N, https://developer.api-trial.jp/en/index.html

Mayumi Takahashi

Senior Research Engineer, Service Platform Innovation & Development Project, Network Service Innovation Project, NTT Network Service Systems Laboratories.

She received a B.Ed. and M.E. in mechanical engineering from Saitama University in 1991 and 1993. She joined NTT in 1993 and began researching electro-magnetic compatibility for chassis of switching equipment and cables. She is currently conducting research on an API-providing architecture for the B2B2X business model.