# The History of SEIL/SMF

## 4.1 Introduction

In recent years, with the widespread use of mobile lines and the development of IoT technology, more devices are connected to the Internet than ever before. But it isn't as easy as simply connecting to the Internet. You need to upgrade the firmware to address vulnerabilities, and manage the operation of devices such as adjusting settings as the usage environment changes. IIJ has developed the "SMF" (SEIL Management Framework) centralized management system to enable the easy operatation and management of a large numbers of devices. Here, we will describe the features of SMF while looking back at its history.

## 4.2 SEIL Development

About 20 years ago, in 1998, IIJ announced its independently developed "SEIL" router products. IIJ, as an ISP, developed SEIL out of a desire to make the Internet something that anyone can use. The ISDN lines that were widely used at the time required you to set the phone number of the access point in the router. To enable users to always use the optimal access point, SEIL was equipped with a function that automatically acquires the access point telephone number.

This protocol for automatically updating the access point helped to alleviate operations and management costs for users. But there is only so much that can be done when information is provided from the ISP unilaterally. The development of SMF began with the goal of creating a system that enables users to freely configure networks and aids them in managing their own network operations.

## 4.3 The Birth of SMF

In the early 2000s, when we were moving ahead with development of SMF, people were gradually beginning to use cheap Internet VPN technology to build secure communication paths over the Internet, instead of using expensive dedicated lines. There were two issues when operating an Internet VPN: 1) the cost of initial setup when installing equipment, and 2) the difficulty of troubleshooting. SMF solved both of these issues through its "auto-connect" and "centralized control" functions.

"Auto-connect" is a function that enables the user to connect to the Internet automatically by simply connecting a cable to a device under factory default settings. Many low-cost Internet access lines used for Internet VPN must have a "connection account" configured as part of the initial settings before connecting a device. This means that it was necessary for devices to be collected by an administrator and configured before being sent out to the installation site, which raised the cost of implementing Internet VPN. SMF solved this issue by dividing the device startup process into two stages*1. When devices compatible with SMF are started up, they begin operation using a configuration for connecting to the SMF server. At this time, a dedicated SMF account that is embedded in the devices is used as the connection account (Figure 1: (1)). The SMF server returns the configuration set by the network administrator to any devices that connect to it. Upon receiving this configuration, the devices update their settings, and then reconnect using a standard ISP connection account before starting any intended operations (Figure 1: (2)). This makes installation work as simple as inserting the cable, and enabling the "auto-connect" function allows users to freely change configurations.

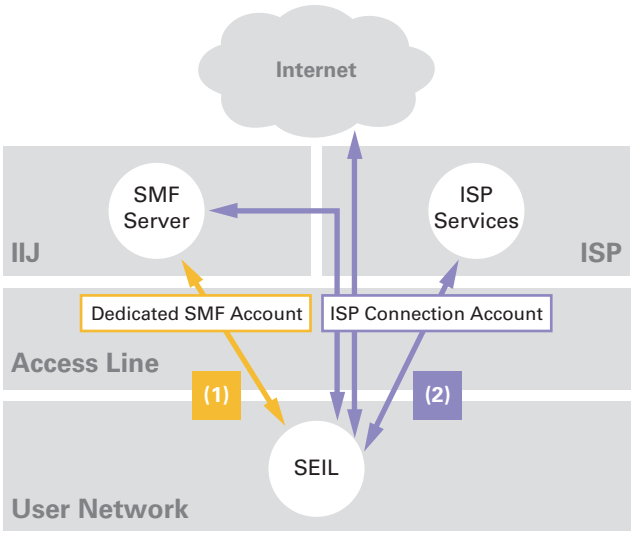


**Figure 1: Auto-connect Using Multiple Connection Accounts**

"Centralized Control" is a function that enables you to manage many devices at the same time. You can update configurations in response to network architecture changes, push distribution of operations management commands, and monitor to check

*1 Patent No. 3774433.

that devices are operating correctly, all at once. We have implemented a range of measures in SMF for managing many devices at once. For example, SMF adopted an active monitoring system that had already produced results within IIJ at the time of development. This monitoring system sends a ping (ICMP echo packet) to devices at fixed intervals, confirming the device and line are operating correctly through the receipt of a response. However, as the number of devices connected to SMF increase, the performance limitations of this monitoring system become apparent. To compensate for this, we developed a new monitoring system that uses a passive monitoring method, called the SMF Heartbeat protocol. In the SMF Heartbeat protocol, the devices send UDP packets at regular intervals, and an error is detected when a device does not send back a heartbeat packet for a set period of time. This protocol is also utilized as a method for regularly sending information from devices, and it solves the scalability issue.

## 4.4 SMFv2 Development

By incorporating SMF technology in the SEIL routers we developed in-house, we solved issues with the initial setup and operations management of routers. But issues with initial setup and operations management are not limited to just routers. Almost all devices connected to the network require some kind of configuration and management. In light of this, we began development of SMFv2 (version 2) to apply SMF technology to devices other than routers.

The main development goal of SMFv2 was "versatility". We refined the SMF technology that could only be used with IIJ's SEIL products into something compatible with devices other than routers, as well as products from other companies. To achieve this, we compiled the SMF technology into a C programming language software library, and called it "libarms," which enabled it to be incorporated into a wide range of devices*2. Devices with libarms embedded are automatically connected to the centralized management server using SMF's auto-connect technology. We also prepared the "SMF SDK," a software development kit, so the centralized management server can be freely customized according to the device. In this way, we provided a structure enabling SMF auto-connect and centralized control to be used on a wide range of devices besides SEIL.

By the way, connectivity with the SMFv2 server becomes an issue when SMFv2 is used on devices other than routers. Currently, when connecting to the Internet using IPv4, it is common to cut back on the number of IP addresses by using NAT. Since communications from the Internet are blocked on hosts behind NAT, it is not possible to push operations management commands from the SMF server. To counter this, we improved the SMF protocol to enable NAT traversal control. When using conventional push protocols, libarms on a device functioned as an HTTPS server, and accepted messages from the SMF server. However, in NAT environments, HTTPS requests from the server do not reach the device. For this reason, devices in NAT environments establish an HTTPS connection with the server, and exchange bidirectional SMFv2 messages on this connection to receive push communications that traverse NAT.

## 4.5 The SACM Cloud-based Centralized Management System

In 2011, a new system was created based on technology that we had cultivated during the SMF development process to date. This system was called "SACM" (Service Adaptor Control Manager). With SMFv2 as a base, it further reinforced the functions required for the centralized control of a large number of devices.

SACM was developed as a cloud system from the initial design stage. Even when implementing devices with libarms embedded, there were cases where on-premise systems built using the SMF SDK had issues with the subsequent development of service functions and the operation of centralized management servers, but with SACM, all these tasks are handled by IIJ. We also focused on the ability to customize, in order to offer OEM solutions for brands that the user is using. Through the development of SACM, we succeeded in lowering the hurdles for utilizing SMF technology, and now there are dozens of OEM partners using SMF*3.

There are also cases where users may want to partially incorporate SACM functions while still using existing systems. SACM supports the REST API as a link between systems. This has made it possible to make devices behave as if they are controlled directly via HTTP

*2 You can download libarms from the SMF portal site (https://www.smf.jp/product-service/libarms.html).
*3 We provide an SACM trial environment to help verify the operation of libarms (https://dev.smf.jp/). Along with libarms, it is available to use free of charge.

requests, rather than simply reading and writing data on the server. By combining libarms-embedded devices with the functions of SACM, and connecting these to external systems via API, we have enabled the creation of new types of services.

One example of this is the IIJ Smart Metering Service for B-Route*4. We developed SA-M0 and SA-M1 as gateway devices that obtain power data from smart meters*5 and send it to the cloud, and these have libarms embedded in them. By leaving the control and monitoring of devices to the proven SACM system, we were able to focus the development on obtaining and transmitting power data, and we were thus able to provide stable services in a timely manner. Everything from the auto-connect of devices, to the execution of commands for obtaining data, and the function for managing the configuration of devices (Figure 2), is achieved through the link between systems using the REST API (Figure 2).

## 4.6 Further Evolution of SMF

The implementation of SACM as a service has made introduction easier, and SMF technology is now used in fields that were not originally conceived. Demand for the operations control of devices is gradually changing due to the spread of mobile technology and the development of IoT technology. Here, we will discuss two features currently being developed to meet new needs for centralized control.

The first is "Legs." This further evolves the NAT traversal bidirectional communication protocol developed for SMFv2, enabling it to handle data in arbitrary formats that the applications on the device depend on. It provides functions for simultaneous transmission of commands to many devices, and also sends event notifications from the device to the server.

The second is "Machinist." The SMF Heartbeat protocol had a mechanism for collecting defined monitoring information from devices. Machinist implements a more versatile information collection function, and has improvements that enable you to gather the information you want at any time. The collected data is visualized (Figure 3), and there is also a function for executing certain actions, such as automatically sending a notification to the user or executing external APIs according to changes in data values.

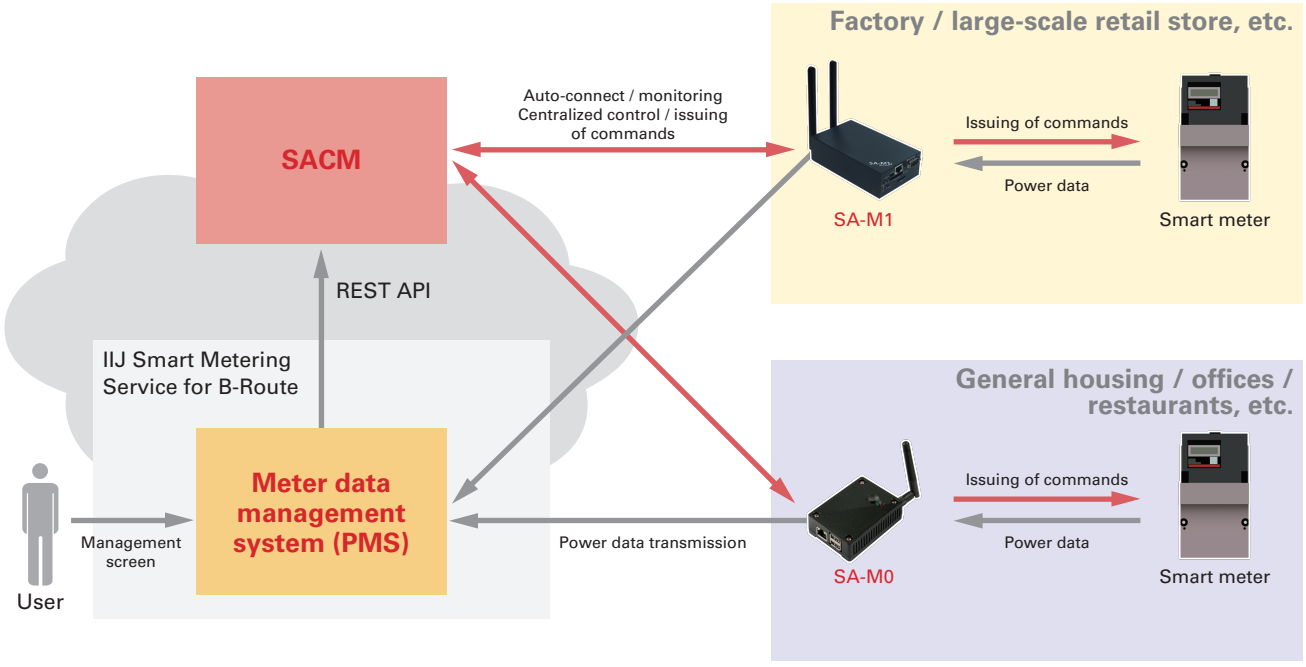


**Figure 2: Linking with the IIJ Smart Metering Service for B-Route**

*4 IIJ Smart Metering Service for B-Route (https://www.iij.ad.jp/biz/smart-meter/).
*5 An electrical power meter equipped with a communications function and capable of remotely obtaining information on power usage.

Each of these is a separate component, with each function designed to be enabled and operated independently. This means you can pick and choose just the SMF functions you need.

## 4.7 Conclusion

Here, we have described the features of SMF and taken a look back over its history. While the core of auto-connect and centralized control remain the same, SMF technology is gradually transforming in response to changes in the Internet usage environment. We are currently focusing our efforts toward developing solutions for utilizing SMF technology in the IoT field. As always, we will continue to develop SMF to meet the new needs that are created each day.

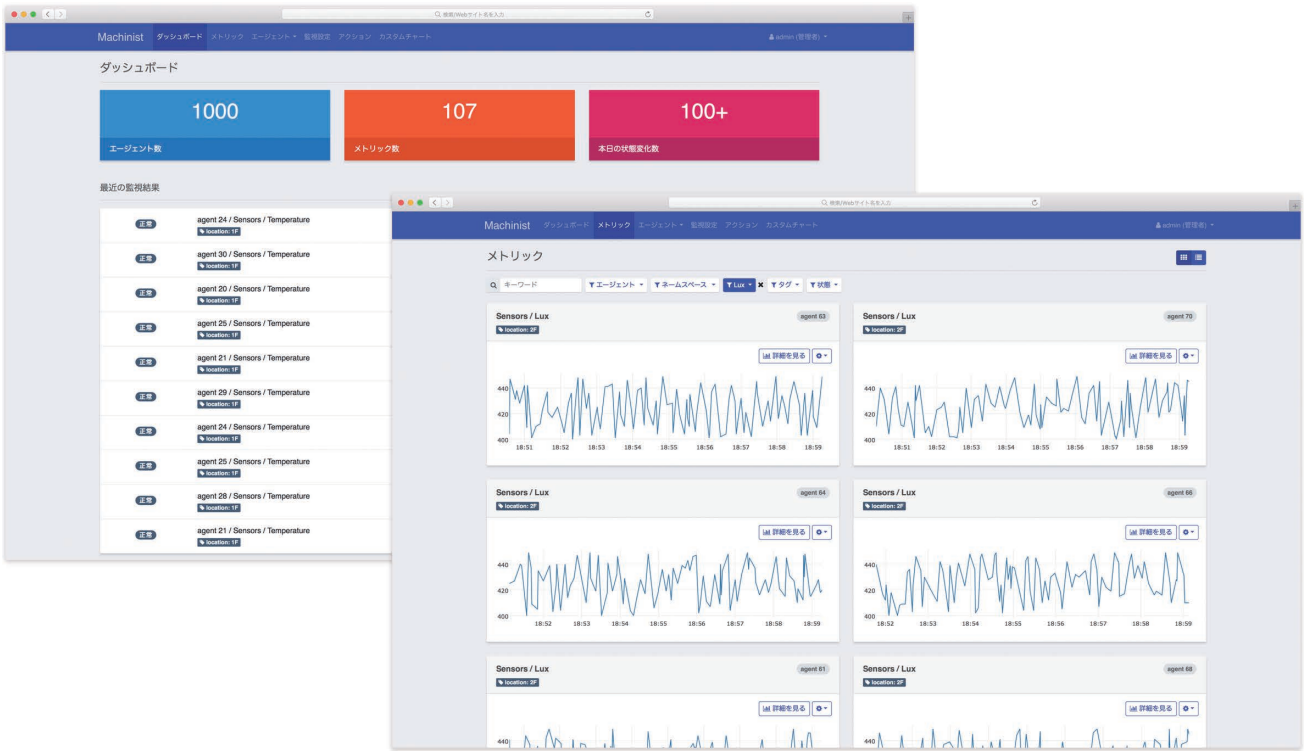


**Figure 3: Machinist User Interface**

Author:
**Tomoyuki Sahara**
Mr. Sahara belongs to the Device Engineering Section, IoT Platform Development Department, Network Division, IIJ.
Ever since joining IIJ in 2003, he has been engaged in the development of router products.
His responsibilities include the development of functions related to IPv6 and routing, quality assurance, responding to vulnerabilities, and the development of SMF.

Author:
**Kiyotaka Kumagai**
Mr. Kumagai belongs to the Sensor Network Section, IoT Platform Development Department, Network Division, IIJ.
Impressed by the mechanisms and approach of SMF, he joined IIJ as a new graduate in 2006.
He has been involved in the development of SMF services ever since he joined the company. He is currently mainly engaged in the development of SACM.