Design Concepts to Satisfy User Data Accessibility of IoT Devices Postulated by the EU Data Act with Bosch Smart Home as a Case Study

Short Paper

1st Felix Fischer
Faculty Applied Computer Sciences and Biosciences
Hochschule Mittweida University of Applied Sciences
Mittweida, Germany
fische11@hs-mittweida.de

2nd Paul Seidel
Faculty Applied Computer Sciences and Biosciences
Hochschule Mittweida University of Applied Sciences
Mittweida, Germany
seidel6@hs-mittweida.de

3rd Dirk Labudde
Faculty Applied Computer Sciences and Biosciences
Hochschule Mittweida University of Applied Sciences
Mittweida, Germany
labudde@hs-mittweida.de

***Abstract*—The Data Act of the European Union (EU-2023/2854) came into effect on the 11th of January, 2024. By September 2025, the 20-month grace period for adoption will end, after which full compliance with the regulation will be necessary. Implementation depends on multiple factors, including the infrastructure in place currently. Accordingly, a typical operation of a smart home device is chosen as the basis for this paper. Sharing the data with the user is of particular interest to the user in this scenario. We describe different implementations designed to satisfy the requirements established by the EU Data Act. In our view, there exist four distinct design solutions to address those requirements. Access can be granted via the data-generating device, a user-hosted server, a smart home hub or the existing cloud. Design proposals are discussed to explain the benefits and disadvantages of each solution. We arrive at the conclusion that expanding existing cloud Application Program Interfaces (APIs) would be the preferred method for most Internet of Things (IoT) devices. However, sharing data via a local controller could also be an effective solution.**

Keywords-eu data act; data availability; iot; smart home; design.

I. Introduction

This paper is an extension of our previous work on the design concepts to satisfy user data accessability of IoT devices postulated by the EU Data Act, presented at IARIA INTERNET 2025 [1].

Under the General Data Protection Regulation (GDPR) framework, users can currently request access to their data, usually provided within a three-month time frame following the request [2]. However, the upcoming European Union (EU) Data Act will impose stricter requirements on data accessibility [3]. Specifically, the EU Data Act mandates that manufacturers ensure that users have direct access to product and related service data. Manufacturers have to fulfill the requirements before the 12th September, 2025. These data must be provided "by default, easily, securely, free of charge, in a comprehensive, structured, commonly used, machine-readable format" [3].

Wolfgang Kerber already looked at the text of the law from a legislative perspective. He came to the conclusion that the EU Data Act does not achieve its goals. "(a) empowering the users of IoT devices (especially the consumers), (b) unlocking large amounts of IoT data for innovation (for IoT-related services and across sectors), and (c) contributing to a fair sharing of the value from the generated IoT data" are mentioned as objectives. All are missed in his view [4].

In this paper, the technical implementation design is mainly considered. Consequently, the scenario of consideration of this paper will be defined in Section Section II. To fulfill these new requirements, various technical approaches are being considered. These approaches will be judged by selected criteria, which are discussed in Section III. Three primary methods for granting users continuous access to their data have been identified. The first approach is pulling data from the data-generating device, allowing users to extract data directly from the origin. The technical advantages and limitations of this approach are discussed in Section IV. The second method involves enabling users to set up a dedicated server for data collection, allowing the device to transmit data both to a cloud-storage service and to the user-managed server. The feasibility, advantages, and drawbacks of this approach are analyzed in Section V. Extending the previous method, a local server could be provided by other existing smart home devices, which then become a hub for all connected devices. This approach is presented in Section VI. The last option that comes to mind provides user access to data via the existing cloud infrastructure, potentially through an Application Programming Interface (API). This method is detailed in Section VII. This paper concludes with a discussion on the most suitable design choice in Section VIII, followed by a case study of real implementation from Robert Bosch Smart Home GmbH in Section IX. Finally, we close with recommendations for future research in Section X.

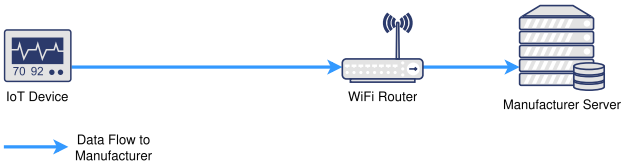

Figure 1. Typical data flow of IoT-devices at home.

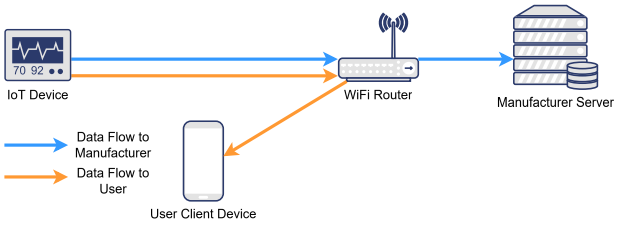

Figure 2. User pulls data from device directly.

II. Scenario under consideration

The term Internet of Things (IoT) device covers a wide range of devices. The considerations in this paper are limited to the classification of IoT devices in the home sector. The term smart home is often used for that. Despite this focus on a small sub-area, it should be possible to cover many other areas affected by the EU Data Act. Therefore, the term IoT device continues to be used here.

For this paper, a typical network setup of an IoT device is assumed. This is shown in Figure 1. The end device connects to a multifunctional device (WiFi Access Point + Switch + Router + Modem), which is commonly referred to as a "WiFi Router". This multifunctional device forwards the recorded data to a server of the respective manufacturer. There, data are collected and prepared for the user, but also for the manufacturer itself. The data flow is unidirectional towards the manufacturer's infrastructure, shown with blue arrows in Figure 1. The user can then typically access a visual representation of the data of their own devices via a web interface or a smartphone app. The access of the raw values collected by the IoT device is not mediationary at the moment.

Depending on the area of application, however, the user does not receive a complete data set. Such visual representations of data are rarely machine-readable without some form of preprocessing, a fact the EU Data Act aims to change. Furthermore, the manufacturer can generate broad knowledge by combining data from different users. In Figure 1 and the following figures, only data streams that are necessary for data provision under the EU Data Act are illustrated.

III. Implementation Considerations

The solutions presented here offer various advantages and disadvantages. In order to be able to evaluate the different approaches, the most important factors to be considered are included.

First, a change in the way of working can require a change on the IoT device itself. Consequently, the IoT device must be able to perform a firmware update. Especially for small devices such as wireless contact sensors for doors and windows or motion sensors, the capability to perform a firmware update is not provided and, therefore, impossible. Thus, a replacement of the hardware would be necessary. This could potentially result in a massive one-time payment.

However, there may also be additional costs due to other factors. This includes, among other things, the implementation costs for changing the software. These costs for programming new functionality come into play both when changing firmware and when adapting software running in the cloud. Especially because the data must be provided free of charge, the cost factor can not be neglected. Additional costs reduce profits. Special attention should therefore be paid to minimizing operating costs. This also includes costs for the provision of network traffic. In addition to computing power, the data stream itself has to be paid for. In the case of one-time paid devices, the running costs for data provision could otherwise wipe out any gained profit.

Finally, the provision of new interfaces increases the attack surface. By choosing a smart design, the attack surface can be reduced. Therefore, possible hacking attacks can be mitigated in advance.

In the analysis carried out here, the respective factors are presented from the point of view of a manufacturer of that IoT device.

IV. Access via the data generating device

The intuitive solution to provide the data according to the requirements listed above is to provide the user with an interface on the data-collecting device itself. As can be seen in Figure 2, the user can collect the data at their own will using this new interface (orange arrows). The current data flow to the manufacturer's server remains untouched (blue arrows). Consequently, the server software does not have to be modified. This means that there are no additional costs for the manufacturer due to the operation.

However, the device needs to be updated with a new firmware, which not all IoT devices support. This means that this option cannot be technically implemented for these devices.

Another positive aspect is that the user can be provided with an arbitrarily dense temporal resolution of the data. In addition, it can be positively emphasized that the data are immediately available to the user. However, most IoT devices lack sufficient storage capacity for a continued retention of the collected data. This means that the user has to query their data at an increased frequency.

IoT devices usually work according to the push principle. This means that they send data and are not actively waiting for a connection. This allows them to switch to a so-called deep sleep mode after sending the data, saving power. In this mode, the Central Processing Unit (CPU) switches to a very economical co-processor, which only waits for an expiring timer. As soon as the time elapses, the CPU is woken up again. This technique can massively reduce electricity consumption

and, therefore, the operating cost of a device. However, if the IoT device will await for a connection from the user, it cannot switch to deep sleep mode. As a result, electricity consumption increases. Thus, this will render small battery-powered appliances unusable within a few days or even hours.

In the event that electricity consumption does not have a restrictive effect, this method can be used to provide the data cost-effectively. This is especially the case if a so-called smart home controller is used. Such a controller is mainly present when a transmission technology that differs from WiFi is implemented and the device cannot communicate directly with the WiFi router. Thus, the controller acts as an access point for the alternative protocol. Common alternative protocols include ZigBee, Digital Enhanced Cordless Telecommunications (DECT), Bluetooth Low Energy (BLE), and Matter. In such a case, the data could be cached on the controller. Figure 3 illustrates, how data can be accessed from there, according to the solution described above in this section. The orange arrows show the data collection using the controller cache. The blue arrows illustrate the unchanged data delivery to the manufacturer server via the WiFi Router. For this implementation, an update of the smart home controller is required.

V. Access via User Hosted Cloud

One way to change the method from Section IV to a push method is to have the IoT device send the data twice: First, as before, to the manufacturer's server (blue arrows) and second, to a local server in the user's network (orange arrows). Strictly speaking, the server operated by the user can also be connected to the Internet. From this second server, the user can access the data at any time and without any other restrictions (green arrows). Figure 4 illustrates this setup.

By changing the access procedure from pull to push, the IoT device can switch back to deep sleep mode between transmission phases. Therefore, the energy consumption will have increased somewhat, but not drastically changed. Because transmission technologies, that are already in use are implemented, no hardware replacement is necessary. However, the firmware must be adapted in the sense that the IoT device sends the data twice and the user can enter to which server the data should be sent the second time. Thus, an update of the firmware will be necessary.

After the data have been transferred, the user has full control, but also full responsibility for their data. Especially

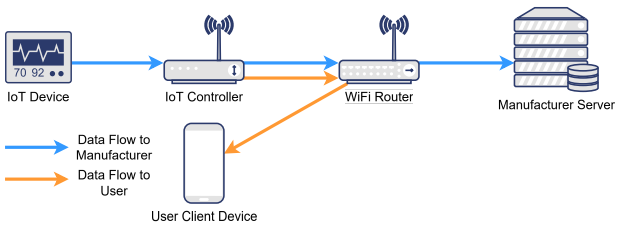

Figure 3. User pulls data from controller.

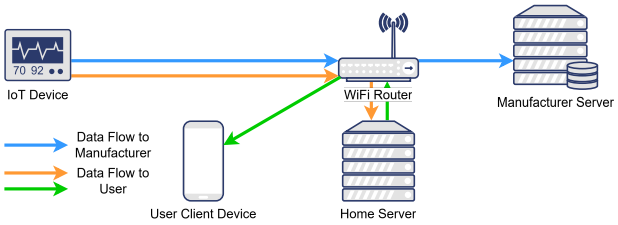

Figure 4. IoT device sends data twice and user accesses data via his local server.

from a data protection perspective, this fact is relieving for the manufacturer. Likewise, there are no further costs for the manufacturer's cloud operation. The management of the data, for its part, remains untouched.

It can be assumed that users expect that server software must be provided by the manufacturer for such a scenario. This must be additionally programmed and maintained, thus causing additional costs. However, these are likely to be quite limited, as the software is based on the already existing server software of the manufacturer's server.

The user has to operate their own server for this approach. It is unclear how this will be assessed as an obstacle to compliance with the EU Data Act by a judge. Making the data accessible in this way could be seen as a contradiction to making it available free of charge. Thus, there is a possibility that this approach will not be classified as meeting the requirements of the EU Data Act.

VI. Access via Smart Home Hub

Building on the approach in the previous Section V, existing hardware can be used as a local server. Some examples include Network Attached Storage (NAS) systems or smart home control solutions such as Alexa that are already in households [5]. Therefore using a pre-existing hub like a NAS or Alexa as the "user-hosted server" as described in Section V. The previous approach could be integrated into these systems. However, this results in the same advantages and disadvantages, except that no new hardware has to be employed.

VII. Access via Existing Cloud

The last approach builds on currently existing infrastructure. In this way, the data are already transferred to the manufacturer's server. Accordingly, it makes sense to set up direct access to the data for the user. As a result, an additional API will be provided or the existing one will be expanded. Figure 5 visualizes this implementation variant. The IoT device transmits data in the way currently used, illustrated by blue arrows. Afterwards, the user accesses the data from the manufacturers server, shown with orange arrows.

If the manufacturer currently does not store its data on its own server, this approach offers a clear disadvantage. The data is stored at the manufacturer's expense. However, the data are of no use to him. The manufacturer's server would, therefore, act as the user's free cloud storage. However, the authors are

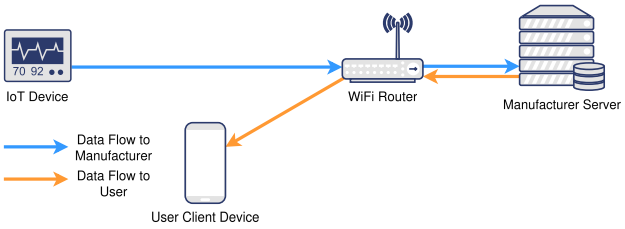

Figure 5. User pulls data from server of manufacturer.

not aware of any smart home device where this described case is observed. Manufacturers always have access to the data.

Expanding existing cloud access offers the clear advantage that no changes or updates of the Smart Home Network or the IoT devices therein are necessary. No firmware update or hardware replacement is required. Since there is no change to the existing hardware, there is no additional energy consumption.

In addition, an extension of server access to user data can be implemented comparatively quickly. Supplementary changes can be continuously integrated in the future. However, this method of implementation generates additional costs in computing power and network traffic on the server.

Access to the data for third parties could be realized via the same interface. For example, the user could grant third parties access to his data by releasing an API token. On the other hand, the manufacturer could also share the collected data with third parties via the same API.

Finally, it can be positively emphasized that this form of transition to the fulfillment of the EU Data Act is not perceptible to the user.

VIII. Discussion

Table I summarizes the analysis of the previous Sections. We provide an impact classification in the discussed aspects.

TABLE I
Summary of Design Concepts

Aspect	Device (S. IV)	User Cloud (S. V)	SH HUB (S. VI)	Cloud (S. VII)
Update needed on IoT device	yes	yes	yes	no
Implementation costs	medium	medium	medium	low
Operating costs for company	none	none	none	cloud capacity
Operating costs for user	electricity	hardware, electricity	hardware, electricity	none
power consumption of IoT device	high	medium	medium	low
Increased attack surface company	no	no	no	yes
Increased attack surface user	yes	yes	yes	no

Since every approach except the one in Section VII requires a firmware update, it should first be clarified whether a firmware update is technically feasible. Without the possibility of adapting the software on the IoT devices or the associated controller, the data provision can only be realized via an API on the manufacturer's server. In particular, if the additional cost that can be expected from making the data available via an API from the cloud is negligible, this implementation should be chosen. It represents the smallest change compared to the status quo.

If the positive factors from Section VII outweigh the negative factors and it is technically feasible, we recommend the implementation from Section IV. This is especially true if a controller is already in use, as shown in Figure 3. Here, it should be checked whether all user-related data are already provided. If this is the case, no changes are necessary to comply with the EU Data Act. Otherwise, the EU Data Act can be implemented cost-effectively by updating and caching the data on the controller with this procedure.

There are valid reasons for the other implementation options. However, it should be considered whether the two options mentioned above are not more suitable for one's own starting position. In particular, due to the uncertain legal situation described in Section V.

IX. Case Study Robert Bosch Smart Home GmbH

The deadline of the grace period ends on the 12th of September 2025. Robert Bosch Smart Home GmbH provided APIs for the user in advance to the EU Data Act. In the following, Robert Bosch Smart Home GmbH extended these APIs to fulfill the requirements given by the EU Data Act. We use this example to analyze the methods used to realize the EU Data Act demands. As a reminder, the requirements for data provision should be mentioned again. These data must be provided "by default, easily, securely, free of charge, in a comprehensive, structured, commonly used, machine-readable format" [3].

A. Requirement to be provided by default

Robert Bosch Smart Home GmbH uses github to document the API available in the Smart Home Controller (SHC). The public git repository https://github.com/BoschSmartHome/bosch-shc-api-docs can be opened by any user, even without an github account [6]. In addition, Robert Bosch Smart Home GmbH provides a website with an in-depth documentation of the Representational State Transfer Application Programming Interface (REST) API. The REST-API documentation can be reached by using URL https://local.apidocs.bosch-smarthome.com/ [7]. Both are lizenced under the Attribution-NonCommercial-NoDerivatives 4.0 International Lizence. Therefore, every customer could access the data cached in the smart home controller. Further more, the API is available without the need for changes on the controller by the user and therefore, active by default.

B. Requirement to be provided easily

In the github documentation at https://github.com/BoschSmartHome/bosch-shc-api-docs/tree/master/postman, Robert Bosch Smart Home GmbH includes an example of applying the REST API by utilizing the API platform

Postman [8]. By doing so, even unexperienced users can make use out of the provided interface. Furthermore, Robert Bosch Smart Home GmbH wrote down best practices to elevate the benefit of unexperienced users.

C. Requirement to be provided securely

The connection between the collecting device and the SHC is secured by a Transport Layer Security (TLS) Layer. The version 1.2 of TLS is implemented. Client authentication is provided using a self-signed X.509 certificate. This must be uploaded by the user to the SHC together with the password used in the SHC. Thus, prevent unauthorized addition of new clients. The user is responsible for the integrity of the private key used in the certificate. Each client will be registered separately with different certificates, improving the integrity of the system. A compromised client could be denied access in the future without affecting the availability of other clients' connections. The used certificate has to utilize an 2048 bit RSA key. This is still widely used in the Internet, but at least 3000 bits are required by the German Federal Office for Information Security, Bundesamt für Sicherheit in der Informationstechnik (BSI) [9]. TLSv1.2 still ensures confidentiality and integrity of the transferred data, but TLSv1.3 introduces new security mechanisms. Therefore, the provided REST-API could be improved using TLSv1.3.

D. Requirement to be provided free of charge

Access to the git repository and the documentation provided is available free of charge. In addition, data access to the local SHC is possible at any time without payment. The SHC is necessary for regular usage of the Bosch Smart Home Devices. Thus, there are no additional hardware costs for the user. However, the customer pays for the use through the power consumption of the network transmission in the local network and the operation of the SHC.

E. Requirement to be provided in a comprehensive, structured, commonly used, machine-readable format

Data collection is done via a REST-like API. Representational State Transfer, or REST for short, is defined as a uniform network interface based on a web server architecture. When it comes to the requirements for an interface, REST focuses primarily on machine-machine communication. Data structuring under REST can be ensured with various data formats such as Extensible Markup Language (XML) or JavaScript Object Notation (JSON). Robert Bosch Smart Home GmbH chose the latter for the implementation of the API on the SHC. The data is transmitted in the formatting of a JSON. JSON is a widely adapted format for data exchanges this kind of. The name JavaScript Object Notation suggests that this data format can only be used in the JavaScript programming language. However, this is not the case. JSON can be used with all common programming languages. It is one of the most commonly used data formats for exchanging data between machines and therefore a great choice of Robert Bosch Smart Home GmbH to make the API usable for the most people feasible. Thus, the JSON data format meets the requirement of comprehensive, structured, commonly used, machine-readable format.

F. Assessment of the realization by Robert Bosch Smart Home GmbH

In addition to the requirements of the EU Data Act, Robert Bosch Smart Home GmbH provides some recommendations for optimal implementation with best practice. These include, for example, Long Polling Subscribe for an up-to-date notification of changed values without delay. "Long Polling is a mechanism in which the client makes a request, and the server keeps the connection open until there is new information available. Once available, the server responds by sending the new information to the client and closes the connection afterwards. As soon as the client receives the new information, it immediately sends another request and the process is repeated." [10]

In summary, it can be said that Robert Bosch Smart Home GmbH decided to implement the EU-Data-Act like described in Chapter IV. The Figure 3 shows the implementation graphically. In this specific case, the orange arrows in Figure 3 symbolize the access to the REST API on the SHC. Since the Robert Bosch Smart Home GmbH provides all the recorded data via the interface and also enables active control, operation is technically feasible completely independently of the Robert Bosch Smart Home GmbH cloud. As a result of the selected implementation, Robert Bosch Smart Home GmbH will not incur any other running costs in addition to the development costs. On the contrary, costs could even be saved if users only used their own infrastructure with the new Rest API. To do this, however, Robert Bosch Smart Home GmbH would have to make it possible to switch off data transmission to the cloud. This is not currently the case.

Users of the API can suggest changes through PULL requests in the git repository and are thus in direct exchange with developers. Users are already creating issues on identified errors and missing documentation. This gives Robert Bosch Smart Home GmbH insight into the use of its customers' smart home hardware and can incorporate this into the design of future products.

The open documentation enables networking with cross-platform systems such as openHAB. As early as November 2020, Robert Bosch Smart Home GmbH devices began to be integrated into openHAB [11].

OpenHAB is a platform for linking a wide variety of smart home devices from various manufacturers on a common control unit. OpenHAB uses Robert Bosch Smart Home GmbH's REST API to integrate it and created matching bindings in the smart home appliance [12]. We have successfully integrated contact sensors, motion detectors, smoke detectors, smart sockets and water detectors from Robert Bosch Smart Home GmbH into OpenHAB in a smart home. The data can be collected without having to make any changes to the device. Thus, it was checked that the data is retrievable by default. Not only can data be read, but requests for actions can also

be transmitted. For example, the smoke detector can also serve as a siren for opening contact sensors. This was also successfully tested. In addition to the IoT devices mentioned above, Robert Bosch Smart Home GmbH also offers other technology for smart hones. These are not available in our laboratory. As a result, it was not possible to fully check whether the requirements of the EU Data Act were met for all Robert Bosch Smart Home GmbH devices.

X. Conclusion and Future Work

In summary, it can be said that there are different approaches to making data available to the user according to the EU Data Act. These different approaches allow a solution to be cleverly adapted to the current situation. In the case study of the Smart Home devices, Robert Bosch Smart Home GmbH has chosen the implementation as described in chapter IV with the special case of an local controller. Robert Bosch Smart Home GmbH provides a REST API for data access and direct device control on the local Smart Home Controller.

This work can be followed by demonstrations of implementation proposals of the designs shown here. For example, the approach from Section VII alone could be implemented in many ways. Depending on which framework is used on the server, the corresponding implementation changes.

For the design of various solutions in this paper, the scenario described in Section II was limited. Thus, the designs presented here do not basically cover all scenarios. In particular, we consider connected cars and cloud services to be worthwhile, advanced fields of research for a submission regarding the EU Data Act.

In this study, only implementations for private devices specifically in smart homes were discussed. In the business-to-business sector, there are different starting points and different requirements are placed on the finished product. This allows for other perspectives on possible solutions.

Acknowledgment

This paper was funded by the European Union and the Free State of Saxony (Germany).

Kofinanziert von der Europäischen Union

Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des vom Sächsischen Landtag beschlossenen Haushaltes.

References

[1] Felix Fischer, Paul Seidel, and Dirk Labudde. “Design Concepts to Satisfy User Data Accessibility of IoT Devices Postulated by the EU Data Act”. In: *INTERNET 2025: The Seventeenth International Conference on Evolving Internet*. Lisbon, Portugal, Mar. 2025, pp. 1–4. ISBN: 978-1-68558-234-0.

[2] Council of European Union. *Council regulation (EU) no 2016/679*. 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679 (visited on 06/01/2026).

[3] Council of European Union. *Council regulation (EU) no 2023/2854*. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202302854, Article 3, last access: 2026.06.01. 2023.

[4] Wolfgang Kerber. “Governance of IoT Data: Why the EU Data Act Will not Fulfill Its Objectives”. In: *GRUR International* 72.2 (Oct. 2022), pp. 120–135. ISSN: 2632-8550. DOI: 10.1093/grurint/ikac107. eprint: https://academic.oup.com/grurint/article-pdf/72/2/120/49174428/ikac107.pdf. URL: https://doi.org/10.1093/grurint/ikac107.

[5] Shubham Jain. “Amazon Alexa Enabled Smart Wi-Fi Switch”. In: *Journal of Multi Disciplinary Engineering Technologies* 13.1 (July 2019). eprint: http://jmdet.com/wp-content/uploads/2019/09/JMDET_13_1_15-NEW.pdf.

[6] *Bosch Smart Home Controller Local API*. Robert Bosch Smart Home GmbH, May 2023. URL: https://github.com/BoschSmartHome/bosch-shc-api-docs (visited on 06/01/2026).

[7] *Bosch Smart Home Local API for Main Resources*. Robert Bosch Smart Home GmbH, Sept. 2025. URL: https://local.apidocs.bosch-smarthome.com/ (visited on 06/01/2026).

[8] *Postman*. Postman Inc., Sept. 2025. URL: https://www.postman.com/ (visited on 06/01/2026).

[9] *Technische Richtlinie - Kryptographische Algorithmen und Schlüssellängen*. BSI TR-02102-1. Jan. 2025. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf (visited on 06/01/2026).

[10] *Best Practice*. Robert Bosch Smart Home GmbH, Dec. 2022. URL: https://github.com/BoschSmartHome/bosch-shc-api-docs/tree/master/best_practice (visited on 06/01/2026).

[11] Christian Oeing. *More information about the provided device data*. Nov. 2020. URL: https://github.com/BoschSmartHome/bosch-shc-api-docs/issues/40 (visited on 06/01/2026).

[12] *Bosch Smart Home Binding*. OpenHAB, Sept. 2025. URL: https://www.openhab.org/addons/bindings/boschshc/ (visited on 06/01/2026).