Towards Longevity of Smart Home Systems

Peter Zdankin
University of Duisburg-Essen
Duisburg, Germany
peter.zdankin@uni-due.de

Marian Waltereit
University of Duisburg-Essen
Duisburg, Germany
marian.waltereit@uni-due.de

Viktor Matkovic
University of Duisburg-Essen
Duisburg, Germany
viktor.matkovic@uni-due.de

Torben Weis
University of Duisburg-Essen
Duisburg, Germany
torben.weis@uni-due.de

***Abstract*—Architectures of current smart home systems are not optimized for longevity of their installations. In this paper we analyze scenarios that can render smart home installations useless much sooner than their non-smart counterparts. We analyze current smart home architectures and classify their components and the protocols connecting them. On this basis we present potential threats to the longevity of such smart home installations and determine areas that need more research to provide the longevity and sustainability that users expect from such premium priced products.**

***Index Terms*—Smart Home, Longevity, Cloud, Internet of Things**

I. Introduction

Smart homes feature a set of accessories that can communicate with other devices and control or measure physical features in the real world [1]. Accessories that are able to be integrated in a smart home system are usually more expensive, therefore a long device lifetime is expected to justify the cost. This long device lifetime is not only defined through the physical lifetime of the accessories themselves. As smart homes are an evolving software system, the lifetime also depends on the longevity of that system. Longevity in the field of software engineering is a recognized problem. The German Research Foundation (DFG) funded a priority programme for researching and solving longevity software development problems which spans several research areas such as software architecture, formal methods and security [2]. However, in the field of smart homes, the problem of longevity is not well understood and researched. Threats such as discontinued external services, breaking updates or even trade conflicts need to be considered in the development of smart home platforms. We classify a set of components and architectures, that can be created by connecting these components, to make smart home systems comparable among each other. Our goal is to work out the impact of certain threats against the longevity of these smart home architectures. From this, we derive topics that specifically require further research to preserve their longevity. In this paper, we first present widely used smart home components and architectures in Sections II and III. Afterwards, we identify threats that render smart home accessories unusable and suggest possible solutions to improve the longevity of smart home systems in Section IV.

II. Components

In this section, we showcase components that regularly appear in smart home systems and describe their intended use. For this, we analyzed popular smart home architectures and identified the most common components. Fig. 1 provides an overview of these components. However, the components are not interconnected in this visualization, because the connections are determined by the specific architecture of the smart home system. Below, we introduce the terminology used in Fig. 1 and in the further course of this paper.

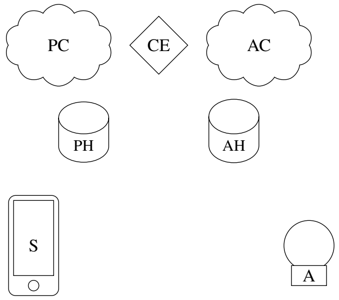

Fig. 1. Commonly used smart home components in smart home architectures. The Figure is made up of 3 main groups. The first group is the cloud group, which consists of the Platform Cloud (PC) and Accessory Cloud (AC) with a Connecting Entity (CE). A cloud can be used (but not limited to) as an intermediate gateway for internet communication between different devices. The PC and the AC can be connected to a CE to cope with different cloud protocols and implementations. The second group is the hub group. Smart home accessories (A) or smartphones (S) can connect to a Platform Hub (PH) or Accessory Hub (AH) to communicate with the cloud or other local devices. In the last group are the actual smart home accessories (A) and smartphones (S) that implement smart home features.

a) Smartphone: Current technologies allow all kinds of mobile devices e.g. voice assistants, smart-watches or tablets to be used for controlling smart home systems. To simplify matters, we limit ourselves to smartphones (S) as a direct link between users and smart home systems.

b) Smart Home Accessory: A smart home accessory (A) can range from a smart light bulb that can be turned on or off to a smart fridge that allows all kinds of technical features. The varying complexity of accessories allows advanced communication protocols between accessories and other components. As accessories may use energy-efficient protocols they might not be able to reach components such as cloud services directly and require intermediate devices such as Accessory Hubs to forward messages. However, it is often wiser to limit the set of communication partners depending on the chosen architecture, as each new communication path adds complexity to an already distributed system. Accessories are very versatile and we use this component to represent all possible accessories that could be part of a smart home.

c) Platform Cloud: A Platform Cloud (PC) is a cloud that is managed by the vendors of the smart home platform and can be used to forward messages to other clouds. Its purpose includes user management, device management, an API for external services and much more, depending on the implementation of the platform. Remote access usually occurs through the Platform Cloud.

d) Accessory Cloud: The Accessory Cloud (AC) is managed by the vendors of smart home accessories and allows to control or administer the smart home accessories. It needs to deal with many issues such as rolling out firmware upgrades to all of their smart home accessories, authenticating themselves to their accessories and fixing security issues if they should arise at some point during the lifetime of a smart home accessory.

e) Platform Hub: The Platform Hub (PH) acts as a mediator between the smartphone and the rest of the smart home system. Its purpose is to forward information to the Platform Cloud (PC), the smartphone, other hubs or the smart home accessories directly. Through utterances by users, the PH can be used to control the smart home, such that a smartphone is not necessarily required to control the smart home.

f) Accessory Hub: An Accessory Hub (AH) serves several purposes. Many smart home accessories lack complex connectivity, such as WiFi or Bluetooth, and require an intermediate device that receives data from other smart home accessories and can forward it to them through energy efficient protocols such as Zigbee or Z-wave [3], [4]. But it can also be used to connect smart home accessories of different vendors to the AH, although the set of available functionalities may be reduced for these external accessories.

g) Connecting Entity: The connecting entity (CE) connects a Platform Cloud (PC) with an Accessory Cloud (AC), as often enough different protocols are used, or some work needs to be performed to translate the request of the PC to the available commands of the AC. The CE can also be used to connect a PC or AC with itself. The CE is a piece of software that may be placed in the cloud and in some systems users are able to configure the CE themselves. It does not necessarily have to be provided by the platform vendor, as external services such as If This Then That (IFTTT) can be used to link the PC with the AC.

III. Architectures

A set of components can be connected to create a smart home architecture. Several approaches have been used in the past with various degrees of flexibility, centrality and ease of use for customers. In this section, we present some of the commonly used architectures and their major differences.

A. Cloud-Centric Architectures

In cloud-centric smart home architectures, a cloud is the essential component. A visualization of such an architecture is shown in Fig. 2. Two examples of such an architecture are the solutions of Amazon and Samsung [5], [6]. In these architectures, a command that originates in the smartphone or Platform Hub (PH) needs to be transmitted into the Platform Cloud (PC). Then, depending on the accessory, the processed command is either sent back to the PH and transmitted to the accessory, or forwarded through the Connecting Entity (CE) to an Accessory Cloud (AC). The AC can send the received commands either directly to an accessory or an Accessory Hub (AH), depending on the connectivity of the accessory. CEs such as Smart Home Skill, SmartThings Connector or IFTTT are mandatory when connecting a PC to an AC, as direct communication between both components is not possible in either approach. The benefits of such a system are the high level of control that platform vendors have over their devices, as administrative tasks, such as device update, can be performed through their cloud. The dependence on the cloud makes this architecture very vulnerable, because the PC, CE and all ACs are required for the smart home to function flawlessly.

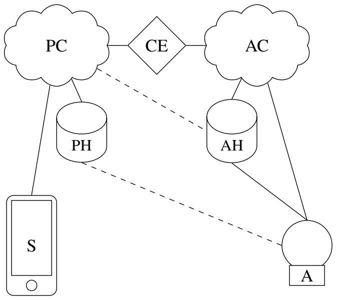

Fig. 2. A Cloud-centric smart home architecture requires the smartphone to send commands via the internet to the PC. The commands are then parsed, processed and forwarded to smart accessories again via the internet to a hub or AC. Smart Home Accessories send data to either their vendor-specific hub or AC. The dashed lines indicate an optional communication path from the PC to AH or Accessories, which may be present in some instances of this architecture.

B. OpenHAB

OpenHAB is a home automation platform that aims to enable a cloudless experience [7]. It is a piece of software that can run on a variety of different devices and requires a considerable amount of manual configuration. The development of OpenHAB is done through a community and all code is open source, which makes it accessible and robust because it can be ported to use different infrastructures and modified to the needs of the users. Unfortunately, the overhead of manually integrating all devices is complex and only possible to be done by people with the necessary technological capabilities. The effort of maintaining such a system can easily surpass the perceived benefits of having a smart home, such that users might feel less attracted to this solution. Although resources have been put into a simplification of it and many commonly used devices can be added at ease, the documentation clearly states that sometimes deeper understanding is required [7].

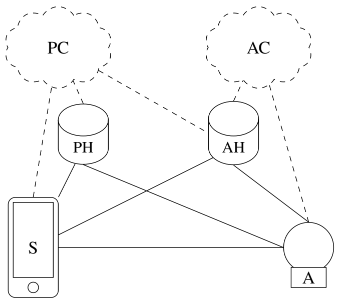

Fig. 3. The Homekit architecture does not require cloud connectivity. Therefore, no Internet communication is required to control smart accessories. In this system, most of the components are able to communicate with a smartphone. For remote use-cases, cloud services can be used for relaying purposes.

TABLE I
VULNERABILITY TO LONGEVITY THREATS. FILLED CIRCLES (●) REPRESENT GOOD PROTECTION AGAINST THIS THREAT, HALF FILLED CIRCLES (◐) REPRESENT THAT IT CAN BE PROTECTED AND EMPTY CIRCLES (○) REPRESENT WEAK PROTECTION.

Threat	Cloud-Centric	OpenHAB	Homekit
Discontinued External Services	○	●	◐
Breaking Updates	○	○	◐
New class of devices	◐	●	◐
Trade Conflict	○	●	○
Abandoned Protocol	○	○	○
Forced Incompatibility	◐	◐	◐
Second Hand Smart Homes	◐	○	◐

C. Homekit

As visualized in Fig. 3, the Homekit architecture differs clearly from the cloud-centric architectures in several aspects, such as the optional status of the cloud. It is possible to control the smart home entirely within the local network, if the accessories do not require external dependencies. If the smart home needs to be controlled from outside of the local network, a Platform Hub (PH) needs to be added to the system, which can forward these commands to the target accessories. If a Platform Hub (PH) is added to the system, many threats that apply to cloud-centric smart home platforms, may apply to Homekit as well.

The underlying Homekit Accessory Protocol (HAP) defines communication between for example an iPhone and smart home accessories over Bluetooth or IP [8]. The HAP defines a list of possible device types such as light bulbs or surveillance cameras and a list of capabilities such as color or brightness [8]. Physical accessories implement these general concepts such that it is possible to control the device solely through these capabilities. Therefore, an explicit Connecting Entity (CE) such as in the cloud-based approaches is not required, which removes a vulnerable component. The major issues with Homekit are its lack of support for Zigbee or other communication protocols, which require hubs that forward these messages to the endpoints. Furthermore, the HAP itself needs to be implemented for each smart home accessory that needs to communicate using this protocol.

IV. LONGEVITY

In this section, we introduce threat models that could affect the longevity of smart homes. We then apply these threat models to the previously described architectures and components and evaluate whether some of them are better equipped to deal with these threats. At the end of each threat, we describe possible solutions or ways to prevent these issues to enable a longer lifetime or robustness against these threats. Table I provides an overview of the results of this section.

A. Discontinued External Services

Many smart home systems require external services to function. For example, a cloud infrastructure is required in some systems to forward or interpret commands, manage user accounts or roll out important updates. If these external services are discontinued, the smart home could be inoperable. Companies may over time either become bankrupt or discontinue services that they created. One recent example of this is the shutdown of the Insignia Connect app, which has the effect that accessories that can be controlled through this app, lose this functionality and are degraded to non-smart devices [9]. There is little guarantee over the long-term availability of such external services, as they have a certain cost associated with them. If this cost cannot be afforded, the external service could be forced to shut down and therefore smart homes should be able to deal with this threat.

1) Cloud-Centric Architectures: If an external service such as the cloud is shut down, there might be no way to control the accessory, as the chain between the controlling smartphone or speaker and a physical device in the home breaks as soon as a single link breaks. Therefore, the longevity of such a system depends on the longevity of the Platform Cloud (PC), the Connecting Entity (CE) and all Accessory Clouds (AC), from which either can be discontinued at some point in time. Samsung SmartThings recognizes this problem and is planning to implement a local processing of home automation instead of remote processing in the cloud to make the platform more robust [10]. Discontinued external dependencies pose a big threat to cloud-centric smart homes, because it could not only threaten specific devices, but even the complete architecture as they have such an essential role in this architecture.

2) OpenHAB: As users are theoretically able to set up their smart home in such a way that the logic and communication between devices is done locally through OpenHAB hubs, the number of external dependencies may be reduced to the necessary minimum. Some applications may require external services, but as these services are replaceable, a discontinued external service is less threatening for such a smart home. Thus, it is more robust against discontinued external services than other platforms.

3) Homekit: The Homekit Accessory Protocol (HAP) describes the communication between a Platform Hub (PH), where a smartphone could also act as a hub, and a Homekit enabled accessory. This communication can be done over Bluetooth or IP and uses external services only if the smart home accessories require them themselves. If not, the communication can function entirely without external services. Thus, the platform is as dependent on external services, as the accessories in an instance of a Homekit platform are.

4) Prevention: To prevent this kind of threat, smart homes should not necessarily depend on external services that cannot be replaced. Platforms should either use external services as an optional feature or allow users to provide this service themselves and be independent of the cloud. This could be done by providing the external services in a container that could be integrated into a local network and would allow us to keep this system running, even when the external service is decommissioned. A solution like this would require further research, as it is not understood how to practically achieve this.

B. Breaking Updates

Over time security issues may be found in smart home accessories, or functionality could be requested. Fixing these issues or adding functionality requires firmware updates on the accessories. Such updates have broken existing functionality in the past and left devices or systems in an inoperable state and required attention from users to fix the issue [11]. For example, an update that was published by Logitech to fix a security problem on their Accessory Hubs (AH), also contained an updated API, which caused problems in some smart home installations [12]. Currently, it is difficult to predict all implications caused by updates of smart home accessories or systems. To predict these implications, other ways to handle updates need to be used, for example by abstracting the behavior and dependencies of devices to a public interface or API. Efforts are made to establish a standard for communication between accessories [13]. Changes to such an interface through updates could be analyzed statically and predict problems that may arise. Abstracting behavior in Smart Home Platforms is not novel but has not yet been applied to the fullest possible extent [5], [8]. Furthermore, smart homes are a distributed and heterogeneous domain and a multitude of distinct devices may likely be upgraded at a point in time. While the implications of a single update may be evaluated, the implications for the complete system can be obscure. When a variety of devices may be updated at the same time, the consequences can be unclear and users cannot possibly decide on the right procedure.

1) Cloud-Centric Architectures: In cloud-centric architectures all clouds, hubs and accessories may receive updates, which could lead to breaking functionality in the smart home. APIs could evolve and change over time, such that high-level communication breaks and accessories could stop providing certain services, which causes incompatibilities between accessories. While the developer documentation to their respective architecture is openly accessible, no guidelines exist over how updates to firmwares or APIs should be created and which rules need to be followed by the respective vendors of components in order to have a robust system [5], [6]. The Platform Cloud (PC), as a central component in this architecture, may even receive updates itself and cause errors for the smart homes of users. Therefore, breaking updates are a threat to these architectures.

2) OpenHAB: The case for OpenHAB ist difficult to answer, because if smart home accessories receive updates they still may break existing functionalities, but because of the deep configurability of OpenHAB, necessary modifications can be applied to fix the system. This requires extensive documentation for the accessories and a high domain knowledge of the person that maintains the smart home after such an issue.

3) Homekit: In Homekit all accessories are described in terms of services, which have certain characteristics. For example, a light bulb service may have an On/Off characteristic which allows to change the power state of the accessory. Therefore, if an update is rolled out, the list of available characteristics and services is known and can be compared to the currently installed version. If this list changes, it could have implications on other accessories, as they may depend on them. If Homekit would add a functionality that collects the available characteristics of the accessories in a smart home, a dependency graph of the smart home could be created. The changes that are introduced through an update could be analyzed on this graph and the user could be informed about certain devices that might not function after an update. For this reason, breaking updates are currently a threat to Homekit, until such a measure is implemented.

4) Prevention: Firmware updates that break existing functionality require further research, because the current solutions only deal with it partially, if at all. It needs to be evaluated, how updates in a smart home should be executed, as for example an approach in which all devices are updated immediately could lead to instability. Solutions that preserve existing features or interact with the user over the consequences of updates are still not available and would require different semantics for updating processes. By abstracting smart devices into their public interfaces, updates could state how they impact these interfaces and thus inform users about the consequences of certain updates. To validate that accessories implement these interfaces, test suites could be introduced to the development process, to reduce the risk of unintended breakage.

C. New Class of Incompatible Devices

If during the lifetime of a smart home, new classes of devices are introduced to the market, which were not previously considered, they may not be available for all smart home platforms. While vendors may continue supporting the smart home platform, they may not necessarily continue further development on it. As a result, older architectures might become obsolete while in use, as these new classes of devices are not available for them. For all architectures that leave little control over the system, this creates a dependency on the platform vendors to eventually support these new kinds of devices. Smart home platforms can either allow users to add new types of devices themselves or leave them no option but to wait for eventual official support.

Prevention: Platforms should be extensible enough such that users or Smart Home Accessory vendors could add new devices themselves. This would require platform vendors to give up a certain degree of control over their system and allow external parties to modify existing software. The benefit of this loss of control is a much more robust architecture, that can deal with unforeseen device types. How the process of adding new classes of devices to an existing platform through external peers should be executed requires further research. One solution could be a central authority that publishes the specifications of a new class of devices and their new capabilities in some formalization. Then a smart home platform vendor could translate this formalization to their platform and solve this threat.

D. Trade Conflict

Smart home platforms and services are trade goods that may be part of trade conflicts, in which these goods become unavailable for a population. Past cases have shown that with too short of notice services, software and websites may be targets of trade conflicts, such as the ban of Android for Huawei or Github for Iran, Syria and the Crimea [14], [15]. The users of these services have little options and need to search for alternatives, if no end of conflict is in sight. In the context of smart homes, this may require a completely new set of accessories or a different platform. The most vulnerable parts to this scenario are external services that are not part of the smart home itself but are required to forward messages between devices and the user. It is also possible to restrict access to components that are required for the smart home, such as Smartphones or Platform Hubs (PH). Such actions have happened before such as the ban of iPhones in China, which are an important asset when using Homekit [16].

1) Cloud-Centric Architectures: Both, Amazon and Samsung SmartThings, are vulnerable in the case of possible trade conflicts that restrict the use of certain services. Smart Home Accessories, their Accessory Cloud (AC), the Connecting Entity (CE), and the Platform Cloud (PC) and the hubs, each single instance could be a target of a trade conflict and as such become unavailable. The implications could reach from certain devices being unusable to the entire smart home becoming inoperable and therefore trade conflicts pose a big threat to cloud-centric architectures.

2) OpenHAB: As OpenHAB is Open Source software, which can be set up on all kinds of different devices, it is more difficult to target this platform with trade conflicts, than other platforms. All devices in the smart home are configured manually and no external services are mandatory, but optional for certain accessories. If these specific accessories are targeted by a trade conflict, they can be easily replaced with other ones that are still available. Thus, OpenHAB seems to be robust against this kind of scenario.

3) Homekit: Homekit does not require external services to communicate with the devices in the smart home. Nevertheless, the closed source approach of Homekit makes it difficult to use Homekit on its full extent on devices other than Apple devices. Unfortunately, these devices have been targets of a trade dispute already [16]. If these devices become banned, it becomes ever more difficult to control the smart home, as for the full utilization of Homekit a device acting as a hub and a controlling device is required. Therefore, until Homekit supports a broader variety of hubs and control devices, the longevity of a smart home that uses Homekit is closely tied to the availability of Apple devices in a country.

4) Prevention: To contain the damage done through trade conflicts, software should not be too dependent on specific hardware, that can easily be targeted such as Homekit, which requires certain Apple devices. But even if hardware restrictions fall and software to control a smart home is available on a broad set of devices, this software could be the target of a trade dispute. If users would have access to the source code of that software and were able to modify it themselves, this could add robustness to the system, and as such architectures like OpenHAB could deal with this threat better than others [7].

E. Abandoned Protocols

Over time communication protocols, technologies or devices may be abandoned, deemed insecure or for other reasons become unusable, like for example WEP or MD5 [17]–[19]. While for some cheap smart home accessories this may be solved with a new device, a smart fridge that just requires a new network card, because its communication protocol is insecure, would be an economic loss if it becomes unusable through this threat and would have to be replaced.

Prevention: It would be desirable if smart home accessories were modular enough, that this issue could be solved economically. If an abandoned protocol only affects the network interface, replacing this specific interface could solve this problem and enable a longer device lifetime. Another option could be an intermediate device that is used between the Smart Home Accessory and another device during a communication. The concrete implementation and viability of such an approach requires further research in this field but could help to mitigate the damage done by a forceful abandoning of a protocol.

F. Forced Incompatibility

There are smart home accessories, that function with accessories or Accessory Hubs (AH) of other vendors to a certain extent. The vendors of a AH may drop this compatibility and without further notice, certain accessories do not function anymore. This may reach from a reduced set of features, such as the inability to update the firmware over accessories with reduced functionality, to a total incompatibility in which devices cannot be controlled anymore until a component, such as another Accessory Hub (AH), is added to the smart home. Interoperability between similar devices of different vendors could help to reduce the number of required components in a single smart home and removing it would affect users negatively. An example of this are OSRAM light bulbs, which can connect to an Accessory Hub (AH) of Philips, but which cannot receive firmware updates without the AH of OSRAM, which defies the purpose of interoperability.

Prevention: Forced incompatibility is a threat with many facets. While solutions like guaranteed support by vendors could work, it could be possible that major updates on one device cause unintended side effects on other devices. The problem could be approached in two ways. Users that are about to update components could be informed about the implications of that update and decide whether to continue with this update and prevent incompatibility by using an older firmware version. Alternatively, devices could describe their requirements to other devices, such that updates are not considered, which obstruct these requirements, which might lead to security issues because some updates cannot be installed. Research needs to be done on how devices of different vendors can inform other devices about their requirements and how the owner of a smart home can be informed about the effects, that an update on one device could have on other devices.

G. Privacy and Security in Second-Hand Smart Homes

If people living in a smart home decide to either move to another place and sell their home or rent the smart home to other peers, a multitude of issues arise. If the ownership of a smart home needs to be transferred permanently, it depends on the platform whether this is possible in a feasible amount of work. While platforms that are linked to an account of some sort might introduce this feature, platforms like OpenHAB, which require a manual setup, might require more steps. Furthermore, if the smart home is only rented, the smart home accessories could still be owned by the landlord, in which case the landlord could invade the privacy of the renter by accessing data from smart home accessories.

Prevention: Transferring the ownership or renting a smart home should be possible by a smart home platform and is not understood currently. The process of transferring the ownership or renting it to other peers should be secure, as malicious or unintended use could be catastrophic. People living in a second-hand smart home should be able to inspect who is able to access data and prevent abuse of such a system. Furthermore, if the ownership is transferred, all user data of the previous owners should be deleted automatically, such that the new owner cannot inspect private information if some accessory was overlooked through manual deletion.

V. Conclusion

We have shown that current smart home systems are not designed with longevity in mind. Architectures relying on cloud-based services are especially vulnerable, because trade conflicts, discontinued products or simply a cloud outage can quickly render smart homes useless. Other threats emerge from new device classes or deprecated and insecure protocols because architectures are not flexible enough and especially expensive smart home devices have no convincing upgrade story.

The average lifetime of a home installation can easily be more than ten years, especially for expensive devices or devices that cannot be easily replaced, because they have been integrated into the house upon time of construction. Therefore, the longevity of smart home systems should be much longer than that of other end-user software.

We outlined several fields that warrant further research to improve the longevity of smart homes. This includes modular devices with a clear hardware update path, flexible architectures with a safe and reliable software update system and security architectures that allow house owners to pass ownership of the house and ownership of the smart things built into it.

References

[1] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *Journal of Cleaner Production*, vol. 140, pp. 1454 – 1464, 2017.

[2] P. D. R. H. Reussner. (2012) SPP 1593: Design for Future - Managed Software Evolution. [Online]. Available: https://gepris.dfg.de/gepris/projekt/198572722?context=projekt&task=showDetail&id=198572722&

[3] D. Gislason, *Zigbee Wireless Networking*, pap/onl ed. Newton, MA, USA: Newnes, 2008.

[4] C. Paetz, *Z-Wave Essentials*. USA: CreateSpace Independent Publishing Platform, 2018.

[5] Amazon. (2019) Understand the Smart Home Skill API. [Online]. Available: https://developer.amazon.com/de/docs/smarthome/understand-the-smart-home-skill-api.html

[6] Samsung. (2019) SmartThings. [Online]. Available: https://developer.samsung.com/smartthings-api

[7] OpenHAB. (2019) OpenHAB Documentation. [Online]. Available: https://www.openhab.org/docs/

[8] *HomeKit Accessory Protocol Specification*, Apple, 10 2019, rev. 2.

[9] K. Holdefehr. (2019) These Smart Home Products Will Soon No Longer Be "Smart"—What to Do If You Own One. [Online]. Available: https://www.realsimple.com/work-life/technology/shopping-gadgets/best-buy-smart-home-insignia

[10] Samsung. (2019) SmartThings. [Online]. Available: https://smartthings.developer.samsung.com/docs/rules/overview.html

[11] C. Lloyd. (2019) Your Smarthome Setup Might Break, and There's Nothing You Can Do About It. [Online]. Available: https://www.howtogeek.com/400112/your-smarthome-setup-might-break-and-theres-nothing-you-can-do-about-it/

[12] V. PALLADINO. (2018) Logitech disables local access on Harmony Hubs, breaks automation systems [Update]. [Online]. Available: https://arstechnica.com/gadgets/2018/12/logitech-firmware-update-breaks-locally-controlled-harmony-hub-systems/

[13] Z. Alliance. (2019) Project Connected Home. [Online]. Available: https://zigbeealliance.org/news_and_articles/connectedhomeIP/

[14] T. Mehta. (2019) [Update 11: Temporary License Extended] Google has revoked Huawei's Android license. [Online]. Available: https://www.xda-developers.com/google-revoke-huawei-android-ban-blacklist/

[15] R. Liao and M. Singh. (2019) GitHub confirms it has blocked developers in Iran, Syria and Crimea. [Online]. Available: https://techcrunch.com/2019/07/29/github-ban-sanctioned-countries/?guccounter=1&guce_referrer_us=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_cs=7ry9nvTX-nObWkIRlRyNSA

[16] Z. Soo and Y. Yang. (2018) Apple's China headache worsens as court bans iPhone sales in Qualcomm patent dispute. [Online]. Available: https://www.scmp.com/tech/gear/article/2177406/apple-hit-qualcomm-patent-dispute-china-court-bans-iphone-sales-upping

[17] A. Bittau, M. Handley, and J. Lackey, "The final nail in WEP's coffin," in *2006 IEEE Symposium on Security and Privacy (S P'06)*, May 2006, pp. 15 pp.–400.

[18] B. den Boer and A. Bosselaers, "Collisions for the compression function of MD5," in *Advances in Cryptology — EUROCRYPT '93*, T. Helleseth, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 293–304.

[19] X. Wang, D. Feng, X. Lai, and H. Yu, "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD," 2004, no lai-xj@cs.sjtu.edu.cn 12647 received 16 Aug 2004, last revised 17 Aug 2004. [Online]. Available: http://eprint.iacr.org/2004/199