Abstract

The Internet of Things (IoT), which describes the equipment of physical objects with sensors, actuators, computing logic, and connectivity, has attracted substantial attention in recent years. Although many facets of the IoT such as technical foundations, business models, and use cases have already been explored, little is known about smart things, the nucleus of the IoT. In most cases, they are treated as a black box. However, an understanding of smart things beyond technical foundations is essential to tap the potential of the IoT in research and practice. Hence, we developed and validated a multi-layer taxonomy of smart things based on the latest literature and a sample of 200 smart things. On this foundation, we also identified five smart thing clusters via cluster analysis. While the taxonomy supports the analysis and design of smart things, the clusters reflect typical combinations of characteristics. Our results advance the understanding of the IoT and facilitate research on the affordances, adoption, and design of smart things as well as on the role of smart things in broader contexts such as IoT ecosystems. From a managerial viewpoint, our results assist practitioners throughout the product development process as they structure the design space of smart things.

Keywords: Internet of Things, Digitalization, Digital Technologies, Smart Thing, Taxonomy, Cluster Analysis
Managerial Relevance Statement

Although smart things are the nucleus of the Internet of Things (IoT), they are typically treated as a black box in research and practice. In reality, however, smart things have different characteristics – ranging from smart fitness trackers over smart vacuum cleaners to smart cars. Hence, an in-depth understanding of the characteristics of smart things beyond technical foundations is in high need when product developers, engineers, and managers set out to explore the design space of smart products, to reason about viable use cases, and to assess the business value of IoT-based solutions. Our research addresses this need through a multi-layer taxonomy of smart things and related clusters. Based on the layers of established IoT stacks and a sample of 200 smart things from relevant IoT application domains, the taxonomy supports the analysis and design of smart things through eleven dimensions whose scope ranges from the physical thing to digital services. Virtually, the taxonomy allows for distinguishing 272,160 smart things. The clusters reflect five typical combinations of characteristics in line with a smart thing’s value proposition and smartness level. They provide a high-level overview of the smart thing universe that enables managers to reason about the role of smart things in broader contexts such as IoT ecosystems or smart service systems.
1 Introduction

The emergence of digital technologies such as Social, Mobile, Analytics, and Cloud leads to changes for individuals, organizations, and society [1–3]. Among the technologies that have attracted considerable attention in recent years is the Internet of Things (IoT), which is defined by the equipment of physical objects with sensors, actuators, computing logic, and connectivity [4–7]. These technology-equipped physical objects, commonly referred to as smart things, link the physical with the digital world and are the nucleus of the IoT [2].

Consulting and market research organizations attribute huge economic potential to the IoT. Market spend related to the IoT amounted to USD 690 billion in 2015, and is expected to reach USD 1.46 trillion in 2020 with a compound annual growth rate of 16 percent [8]. It is reported that, every single second, about 130 new devices are connected to the Internet [9] and a total of about 31 billion devices will be installed in 2020 [10]. For five years in a row, the IoT has been presented as a dominant trend in the Gartner Hype Cycle [11, 12]. Moreover, the smartness of physical products has shifted from an optional feature to a prerequisite for market entry in many business areas [13]. The International Telecommunication Union even expressed the vision that “a new dimension has been added to the world of information and communication technologies […]: from anytime, any place connectivity for anyone, we will now have connectivity for anything” [14, p. 2]. The ubiquity of the IoT is reflected in the diversity of its application domains, e.g., Smart City, Smart Mobility, Smart Health, Smart Home, or Smart Factory [2]. Moreover, the IoT is not restricted to smart things. Rather, it enables product systems, which encompass closely related smart things, as well as systems of systems or IoT ecosystems, which coordinate product systems [5].

Multiple facets of the IoT have already been examined, including technical foundations [15–17]. For example, LaBuda and Gillespie [18] focus on challenges pertaining to security, interoperability, and data processing. The IoT has also been explored from a business-to-business
(B2B) perspective, mainly with a focus on IoT-enabled innovation in the logistics and supply chain sector [19, 20]. Witkowski [21], for example, presents solutions for transport and logistics, showing that almost 90% of companies in this sector will soon implement IoT-based solutions. In addition, the IoT has been investigated from a business-to-consumer (B2C) perspective [22–24]. Porter and Heppelmann [5] and Rosemann [6], for example, provide insights into the challenges and opportunities of IoT-enabled business models. Furthermore, Ju et al. [25] propose a business model framework for IoT-based businesses. In this context, Oberländer et al. [4] argue that smart things will become autonomous actors and they propose business-to-thing (B2T) interactions as substitutes for human-intensive B2C interactions, an idea that has been complemented by Beverungen et al. [26], who reason that smart things serve as boundary objects among businesses and consumers.

Despite their valuable contribution, all these works treat smart things as a black box. Thereby, they disregard that, in reality, smart things feature diverse characteristics – ranging from smart fitness trackers over smart vacuum cleaners to smart cars [27]. However, an understanding of the characteristics of smart things beyond technical foundations is vital for tapping the potential of the IoT in research (e.g., understanding smart things in broader contexts such as IoT ecosystems) and practice (e.g., by structuring the design space of smart products). Against this background, we investigate the following research question: *What characteristics can be used to distinguish smart things?*

To answer this question, we analysed smart things in two steps. First, we developed a taxonomy that enables classifying individual smart things, in line with Nickerson et al.’s [28] taxonomy development method and drawing on existing IoT architectures. To build and validate the taxonomy, we used a sample of 200 smart things. Based on the classification of these smart things, we applied cluster analysis to infer five smart things clusters each covering characteristics that typically occur together. We confirmed the validity and reliability of these clusters via the Q-
sort method within the author team and other researchers. Our study advances our understanding of the IoT by means of a conceptually well-founded and empirically validated taxonomy of smart things and related smart things clusters.

Our study is structured as follows: In section 2, we provide domain background on the IoT. In section 3, we outline the research method. In section 4, we present our developed taxonomy and the evaluation results. In section 5, we present the clusters and the related evaluation results. Our study concludes in section 6 with theoretical and managerial implications, followed by a discussion of limitations and an outlook on future research.

2 Domain Background

Smart things relate to the third wave of IT to transform competition and strategy [5]. The first wave brought process automation, which increased productivity through the automated collection, processing, and analysis of data [29]. The second wave, characterized by the uptake of the Internet, enabled new levels of connectivity with free data exchange. Despite their impact on value chains, the first two waves of IT did not change the nature of products offered by companies. In the third wave, IT is embedded into products in the form of sensors, actuators, computing components, and connectivity [30]. The embedding of such digital capabilities are constitutive of pervasive digital technologies in general and of the IoT in particular [5, 7]. The technical foundations enabling smart things are miniaturization, powerful microprocessors, affordable and reliable memory as well as broadband communication [31].

For some time, there was no common understanding of the term Internet of Things [2, 15, 32, 33]. One reason is that it has been used to convey different conceptualisations of two dimensions, namely the communication dimension and the dimension pertaining to the thing itself. As for the communication dimension, it was not clear whether wired networks should be included or not [34]. The same holds for the thing dimension regarding the inclusion of devices.
such as personal computers (PCs), smartphones, or tablets or whether smart things may only have a virtual representation [2, 4, 15, 34–38]. In recent works, the IoT is consistently defined as the connectivity of physical objects, equipped with sensors, actuators, and computing logic, with the Internet through communication technology [4, 27]. Thereby, the communication dimension covers wired and wireless technologies [34, 35]. As for the thing dimension, smart things can exist independently from IT [38, 39], which is why PCs, smartphones, or tablets are treated as intermediaries between humans and smart things [34]. Both physical materiality and a representation in the digital world are constitutive of smart things [26, 40].

Smart things can act ever more independently from human agency [5, 41]. Oberländer et al. [4] found that the IoT enables and requires a new perspective on material agency, which has so far been defined as the way an object acts when humans provoke it [41, 42]. This restrictive understanding of material agency had to be updated, as the IoT empowers smart things to act and make decisions independently of human agency. This capability, which is also referred to as autonomy or self-dependency, builds on self-x capabilities (e.g., self-controlling, self-monitoring, self-diagnosis, self-configuration, self-learning) and advanced data analysis capabilities (e.g., predictive and prescriptive analytics) [5, 43–46].

When it comes to the implementation of IoT-based solutions, architecture models referred to as IoT (technology) stacks have attracted widespread attention. Prominent examples are the works of Porter and Heppelmann [5], Fleisch et al. [47], and Yoo et al. [31]. Despite differences in their structuring and labelling, the layers of IoT stacks largely overlap. For our purposes, we considered the following layers: thing, interaction, data, and service layer. The thing layer, which is the bottom-most layer, relates to the physical thing equipped with sensors, actuators, and computing logic. The interaction layer covers the ability of physical things to remotely interact with and to be remotely accessed by other entities through their representation in the digital world. In addition, the data layer refers to data collection and usage. Finally, the service
layer, which is the top-most layer, investigates the value proposition of smart things and their role in broader contexts [47, 48]. In line with these layers, the embedding of digital capabilities in physical objects drives the fusion of the digital and the physical world, a development that entails an integrated product/service perspective when analysing smart things [5, 7, 49].

3 Research Method

3.1 Development of a Taxonomy of Smart Things

To answer the research question, we first developed a taxonomy of smart things in line with the iterative taxonomy development method of Nickerson et al. [28]. Taxonomies, a term often used interchangeably with framework or typology, are classification schemes composed of dimensions and characteristics, which help understand, describe, analyse, and classify objects in emerging domains such as the IoT [28, 50]. When developing and validating our taxonomy, we drew on the latest IoT literature and a sample of 200 smart things from the B2C domain. We used smart things from the B2C domain as much more information was publicly available [4].

The taxonomy development method by Nickerson et al. [28] includes seven steps: determination of a meta-characteristic, determination of ending conditions, iterative choice of approach, conceptualization of characteristics and categories, examination of real-life objects, design or revision of the taxonomy, and testing of ending conditions. The meta-characteristic is specified at the beginning of the taxonomy development process. It represents the basis from which all other characteristics are derived. As for the choice of approach, Nickerson et al. [28] propose an empirical-to-conceptual and a conceptual-to-empirical approach. In the empirical-to-conceptual approach, real-world objects (i.e., smart things) are selected and characteristics are induced. Then, characteristics are grouped, given conceptual labels, and assigned to dimensions [51]. The conceptual-to-empirical approach starts without real-life objects. In a deductive process, researchers first conceptualize dimensions and characteristics by leveraging extant domain
knowledge. After that, dimensions and characteristics are examined empirically by classifying real-life objects. This leads to an initial or revised taxonomy. Both approaches can be selected as required and are iterated until all ending conditions are met.

For our purposes, we built on the work of Püschel et al. [27], who proposed an initial taxonomy of smart things with ten dimensions structured according to the layers of IoT stacks. This taxonomy was evaluated using a small sample of 50 smart things. With the IoT domain having matured over the last years, we deemed it reasonable to update and extend the taxonomy of Püschel et al. [27]. Methodologically, the extension of taxonomies complies with Nickerson et al. ’s [28] recommendations. Nevertheless, our research does not only strive for understanding individual smart things, but also to identify clusters of smart things based on characteristics that typically occur together. The identification of smart thing clusters required an up-to-date taxonomy, which accounts for the latest conceptual developments, as well as a broad sample of currently available smart things. Prior to the identification of smart thing clusters, we conducted three iterations. To ensure an incremental and convergent research process, we used the ending conditions and the meta-characteristic of Püschel et al. [27], namely: characteristics of smart things structured according to the layers of IoT stacks.

Our taxonomy development process ended after all objective ending conditions were met: (1) every characteristic is unique in its dimension, (2) every dimension is unique and not repeated, (3) at least one object is classified under each characteristic of each dimension (i.e., no empty characteristics), and (4) no new dimensions or characteristics have been added in the last iteration. Another condition is that characteristics should be mutually exclusive and exhaustive. However, the classification of smart things revealed that, in many cases, multiple characteristics per dimension apply. Hence, in line with other works, we allowed for non-exclusive characteristics [27, 52, 53]. We also specified which dimensions are nominally or ordinally scaled, an information relevant for the application of clustering techniques [54]. As for subjective ending
conditions, we terminated the taxonomy development process after all co-authors agreed that
the taxonomy was concise, robust, comprehensive, explanatory, and extendible. Below, we
show how we compiled our sample of 200 smart things and provide an overview of the taxon-
omy development process in Table 1.

Iteration 1: In the first iteration, we chose the conceptual-to-empirical approach to conceptual-
ise characteristics based on the latest IoT literature (e.g., product smartness, autonomy, ecosys-
tem integration). We used the latest versions of those 50 smart things collected by Püschel et al. [27], which have been identified through a literature review [2, 5, 6, 15, 44] and a search in
the Crunch Base database, which is considered “the most influential technology blog in the
USA” [55, p. 244].

Iteration 2 and 3: In the second and third iteration, we followed the empirical-to-conceptual
approach. This enabled us to consider recent developments within the IoT market. In both iter-
ations, we selected an additional set of 75 smart things based on recent best lists included in
press releases of technology news web sources (i.e., CNET, TechCrunch, and ECT News Net-
work’s TechNewsWorld). We also added smart things from publicly updated IoT lists (e.g.,
iotlist.co and smarthomedb.com). Like Püschel et al. [27], our goal was not to gather an ex-
haustive sample of smart things, but to ensure that our sample covered relevant application
domains in the B2C context [2], namely Smart Home (49%), Smart Health (17%), Smart En-
ergy (9%), Individual Well-Being (21%), Smart Mobility (3%), and Smart City (2%).

After all ending conditions had been met, two co-authors independently classified the sample
of 200 smart things (Appendix 2) to validate the final taxonomy [28]. To do so, we used only
publicly available information such as that provided on homepages or in press releases (Appen-
dix 1). To measure agreement, we calculated dimension- and object-specific hit ratios, where
agreement and disagreement were counted as 1 and 0 [56, 57]. For non-exclusive characteris-
tics, we also considered partial agreement such that all dimensions were weighted equally.
<table>
<thead>
<tr>
<th>#</th>
<th>It</th>
<th>Approach</th>
<th>Changes</th>
<th>Ending conditions</th>
<th>Real-life objects</th>
</tr>
</thead>
</table>
| 1 |    | Conceptual-to-empirical      | - Addition of the dimension autonomy to the thing layer (based on the literature)  
- Addition of the human-computer interaction to the interaction layer (based on the literature)  
- Removal of the dimension human-computer interaction based on object classification  
- Update of the definitions of the sensing capabilities dimension in line with Daft and Lengel [58]  
- Addition of the dimension ecosystem integration to the service layer and removal of the dimension thing compatibility by Püschel et al. [27] | x Objective conditions violated:  
- Empty characteristic (i.e., human-computer interaction) as the type of interaction (e.g., mechanical or acoustic) described the smart thing for the intended users in too much detail  
x Subjective conditions violated:  
- Not explanatory as the former definition described that smart things collect little or huge amounts of data which did not allow for to comparing smart things  
- Overall sample: 125 smart things | Further 75 smart things identified from recent best lists in technology news web sources (e.g., CNET and TechCrunch) and from publicly updated IoT lists (e.g., iotlist.co and smartthomedb.com). Overall sample: 200 smart things |
| 2 |    | Empirical-to-conceptual      | - Removal of the characteristic many-to-many of the dimension multiplicity  
- Switch of the two characteristics limited and none of the dimension offline functionality  
- Renaming of the dimension main purpose to value proposition and emphasize the differentiation of the characteristics thing-centric and service-centric | x Objective conditions violated:  
- Empty characteristic as the taxonomy focuses on single smart thing  
- Subjective conditions violated:  
- Not concise/useful as smart things could not be classified intuitively as we classify smart things along the dimensions’ scale (i.e., ordinal scale)  
- Overall sample: 125 smart things | Further 75 smart things identified from recent best lists in technology news web sources (e.g., CNET and TechCrunch) and from publicly updated IoT lists (e.g., iotlist.co and smartthomedb.com). Overall sample: 200 smart things |
| 3 |    | Empirical-to-conceptual      | - Final finetuning of labels of characteristics to enable unambiguous and intuitive classification | ✓ All objective and subjective conditions met:  
Every dimension is unique and not repeated, at least one object can be classified under each characteristic of each dimension. All authors agreed that the taxonomy was concise, robust, comprehensive, extendible, and explanatory | Further 75 smart things identified from recent best lists in technology news web sources (e.g., CNET and TechCrunch) and from publicly updated IoT lists (e.g., iotlist.co and smartthomedb.com). Overall sample: 200 smart things |
3.2 **Identification of Smart Thing Clusters**

Based on our taxonomy, we aimed at understanding which characteristics of smart things occur together, as the taxonomy offers 272,160 possibilities for classifying smart things\(^1\). While this is appropriate for product engineering and technical research (e.g., when exploring the design space smart things), it is too detailed for managerial purposes and research (e.g., when investigating the role of smart things in broader contexts or adoption drivers of smart things).

Hence, we set out to identify smart things clusters. We applied cluster analysis, a statistical data analysis technique that groups objects according to their similarity [59, 60]. A key objective of cluster analysis is to group homogeneous objects into clusters [61, 62], whereas heterogeneous objects should be assigned to different clusters [63, 64]. The literature offers various clustering techniques, with the separation into hierarchical and partitioning algorithms being most common [65–67]. Partitioning algorithms determine the number of clusters ex-ante, followed by an iterative re-assignment of objects. Hierarchical algorithms either group objects according to their similarity in an agglomerative way starting with \(n\) clusters (where \(n\) refers to the number of objects), or they divide clusters starting with one cluster of \(n\) objects [63]. We used Ward’s \([68]\) agglomerative hierarchical algorithm [69–72] and the Manhattan-metric, which have proven useful in combination [73]. We also chose this setup due as it fits our data set.

Cluster algorithms require comparable input variables regarding unit and scale level. In our setting, three variable types had to be considered: (1) dimensions with nominal and mutually exclusive characteristics, (2) dimensions with nominal and non-exclusive characteristics, and (3) dimensions with ordinal and mutually exclusive characteristics. To not attribute too much importance to specific dimensions, we ensured that the distance between two smart things can

---

\(^1\) The number of possible realizations can be calculated as follows: For dimensions with mutually exclusive characteristics, the number of characteristics is considered. For dimensions with non-exclusive characteristics, the cardinality of the characteristics’ power set minus 1 is considered. Finally, all dimension-specific parameters must be multiplied, leading to the following possible number of realizations: \(3 \cdot 2 \cdot 2 \cdot 3 \cdot (2^2-1) \cdot (2^3-1) \cdot 2 \cdot 2 \cdot 3 \cdot (2^2-1) \cdot 2 = 272,160\)
take a maximum of 1 per dimension. As for the first variable type, we followed Bacher et al.’s [54] approach to encoding nominal binary variables. This led us to assign the values 0 or 1, since nominal and mutually exclusive dimensions can be split in two characteristics. As for dimensions with nominal and non-exclusive characteristics, we found no best practice in the literature. Thus, we followed Bacher et al.’s [54] approach for encoding nominal binary variables by splitting variables into dummy variables, one per characteristic. This implies that, if a dimension consists of two characteristics, two dummy variables are used. To ensure a maximum distance of 1 between two objects, Bacher et al. [54] propose a multiplication with 0.5. We extended this idea to dimensions with three or four characteristics, and thus multiplied by 0.33 or 0.25, respectively. Finally, as for dimensions with ordinal characteristics, we applied a quasi-metric scaling to maintain the ordering information [74, 75]. Dimensions were coded with a scale of numbers ranging from 0 to 1 to ensure equal weighting.

To conclude the cluster analysis, we had to determine a suitable number of clusters. Generally, there is a trade-off between the manageability of the cluster solution and the similarity of the objects per cluster [76, 77]. In hierarchical clustering, the techniques used to determine an appropriate number of clusters are referred to as stopping rules [76]. Despite substantial research in this field, clear recommendations are missing [78]. Thus, the appropriate number of clusters needs to be justified case-wise both quantitatively and qualitatively [79].

To determine an appropriate number of clusters, we started with a quantitative approach based on the following criteria: Error Sum of Squares (ESS) [68, 80], the majority rule for twenty quantitative indices [81], and two further graphical indices [81]. The ESS, where a significant knee in a graphical plot points to the optimal number of clusters, suggested three clusters. After that, we calculated the number of clusters according to the CH [82] and the Duda index [83], those two stopping rules that performed best in a simulation study by Milligan and Cooper [76]. The suggested numbers of clusters were two and 12, respectively, indicating no unambiguous
solution. Thus, we calculated another 18 indices, leading to range from one to 15 clusters. By applying the majority rule (i.e., the most frequently occurring cluster solution is selected), we obtained a suggested number of three clusters [81]. The Hubert index and Dindex are two examples for graphical indices following the ideas of the ESS. While the Hubert index proposed to use five clusters, the Dindex suggested three and five clusters. Based on these indications, we concluded that the optimal solution ranged between three to five clusters.

On this foundation, we continued with a qualitative approach based on our knowledge of the sample and the IoT domain. We interpreted the solutions with three, four, and five clusters – and that with six clusters to offset potential bias. In the three- and four-cluster solutions, valuable information about the characteristics of smart things were lost (e.g., autonomy and ecosystem integration). In the six-cluster solution, a cluster with 15 smart things has been separated from a cluster with 51 smart things. The new cluster could only be distinguished by one characteristic, all other characteristics were identical to the five-cluster solution. As the six-cluster solution was very similar to the five-cluster solution and no additional information could be derived, we excluded the six-cluster solution. Finally, we chose the five-cluster solution, as each cluster could be reasonably interpreted stand-alone and compared to all other clusters.

To evaluate the clusters’ reliability and validity, we applied the Q-sort method, a statistical approach that can be used to classify items (i.e., smart things as our Q-set) in accordance with predefined constructs (i.e., smart things clusters) by two or more judges (P-set). Developed to examine people’s attitudes and opinions [84], the Q-sort has been applied in marketing, psychology, and sociology [85] as well as to test taxonomies [4, 86, 87]. The degree of agreement between the judges forms the basis for assessing reliability and validity [57]. Reliability can be measured via the Kappa Coefficient, defined as “the proportion of joint judgment in which there is agreement after chance agreement is excluded” [57, p. 115]. While Cohen’s Kappa [88] only accounts for the opinions of two judges, Fleiss’ Kappa covers the agreement of more than two
judges [89]. Validity is measured in accordance with Moore and Benbasat’s [56] hit ratio, defined as the frequency of with which items are correctly assigned to the predefined constructs. When applying the Q-Sort, judges require a detailed understanding of the subject in focus, and should not be selected randomly [86]. Hence, we only included participants with an IoT background. As for the Q-set, we prepared a sample of 20 smart things, selecting at least three smart things from each cluster as determined by the cluster algorithm to ensure a representative selection. We also selected at least two smart things from each domain covered by our sample (i.e., Smart Home, Smart Health, Smart Energy, and Individual Well-Being constitute 95% of our sample) [2]. We applied the Q-sort in two scenarios: first, two co-authors who were yet unfamiliar with the cluster solution classified 20 smart things. Second, 15 academics with an IoT background did the same. For both scenarios, we calculated reliability and validity. A summary of our Q-sort application is shown in Table 2 and detailed information on the Q-set (i.e., the 20 smart things selected for the Q-Sort) can be found in Appendix 3.

Table 2: Evaluation Criteria for Two Rounds of the Q-Sort

<table>
<thead>
<tr>
<th></th>
<th>Scenario 1</th>
<th>Scenario 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-set</td>
<td>2 co-authors</td>
<td>15 academics with a background in IS</td>
</tr>
<tr>
<td>Q-set</td>
<td>20 real-life objects</td>
<td>20 real-life objects</td>
</tr>
<tr>
<td>Construct validity measure</td>
<td>Hit ratio(s) [56]</td>
<td>Hit ratio(s) [56]</td>
</tr>
<tr>
<td>Reliability measure</td>
<td>Cohen’s Kappa coefficient [88]</td>
<td>Fleiss’ Kappa coefficient [89]</td>
</tr>
</tbody>
</table>

4 A Multi-Layer Taxonomy of Smart Things

We now present our taxonomy for classifying individual smart things. The taxonomy comprises four layers structured according to existing IoT stacks, eleven dimensions structured according to these layers, and 28 characteristics. Figure 1 shows the taxonomy, whereas Table 3 compiles definitions of all characteristics along with selected justificatory references. Below, we elaborate on each layer. Starting with a short description per layer, we present all related dimensions and characteristics using justificatory references and real-world examples from our sample.
Figure 1: Multi-layer Taxonomy of Smart Things

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Characteristics</th>
<th>Scale</th>
<th>Exclusivity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Service</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ecosystem Integration</td>
<td>None</td>
<td>Proprietary</td>
<td>Ordinal</td>
</tr>
<tr>
<td>Value Proposition</td>
<td>Thing-centric</td>
<td>Service-centric</td>
<td>Nominal</td>
</tr>
<tr>
<td>Offline Functionality</td>
<td>None</td>
<td>Limited</td>
<td>Nominal</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Usage</td>
<td>Transactional</td>
<td>Analytical (basic)</td>
<td>Ordinal</td>
</tr>
<tr>
<td>Data Source</td>
<td>Thing State</td>
<td>Thing Context</td>
<td>Thing Usage</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partner</td>
<td>User(s)</td>
<td>Business(es)</td>
<td>Thing(s)</td>
</tr>
<tr>
<td>Multiplicity</td>
<td>One-to-one</td>
<td>One-to-many</td>
<td>Nominal</td>
</tr>
<tr>
<td>Direction</td>
<td>Unidirectional</td>
<td>Bi-directional</td>
<td>Nominal</td>
</tr>
<tr>
<td><strong>Thing</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Autonomy</td>
<td>None</td>
<td>Self-Controlled</td>
<td>Self-Learning</td>
</tr>
<tr>
<td>Acting Capabilities</td>
<td>Own</td>
<td>Intermediary</td>
<td>Nominal</td>
</tr>
<tr>
<td>Sensing Capabilities</td>
<td>Lean</td>
<td>Rich</td>
<td>Ordinal</td>
</tr>
</tbody>
</table>

ME: Mutually exclusive NE: Non-exclusive

4.1 Thing Layer

As the bottom-most layer of IoT stacks, the thing layer is the foundation for all other layers [5]. Here, physical things are transformed into smart things by being equipped with *sensing* and *acting capabilities* [2]. The thing layer also covers the *autonomy* of smart things [45].

Sensing Capabilities

The collection of data about the physical environment (e.g. temperature, humidity, or brightness) is a constitutive characteristic of smart things [2]. Different technologies are available to implement sensing capabilities, for example, sensor nodes integrated in wireless networks [15]. Appelboom *et al.* [90] provide an extensive overview of sensor types and related technologies.

For our purposes, we take an information processing perspective in line with media richness theory [58]. With information richness being defined as the ability of information to change understanding, we focus on what smart things can collect, deliberately abstracting from technical details. We also abstract from the quantity of sensors. Rather, we focus on whether a smart
thing has lean or rich sensing capabilities. Smart things feature lean sensing capabilities, if they collect simple data, for example, the smart shower Eva Drop with sensors recording position and temperature. Rich sensing capabilities are found in smart things that collect complex data, such as the smart security camera Nest Cam IQ that uses 4K-colour sensors and sub-sound sensor technologies.

**Table 3: Definitions and Justificatory References**

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Definitions</th>
<th>Justificatory References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ecosystem Integration</td>
<td>None: There is no possibility to integrate the smart thing into ecosystems.</td>
<td>[5, 34, 44, 47, 49, 91]</td>
</tr>
<tr>
<td></td>
<td>Proprietary: The smart thing can be integrated into ecosystems, but is only compatible with smart things of the same provider or manufacturer.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Open: The smart thing can be integrated into different ecosystems, as it is compatible with smart things of other providers and manufacturers.</td>
<td></td>
</tr>
<tr>
<td>Value Proposition</td>
<td>Thing-centric: The smart thing primarily serves a thing-related purpose.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Service-centric: The smart thing serves as distribution mechanism for digital services.</td>
<td></td>
</tr>
<tr>
<td>Offline Functionality</td>
<td>None: The smart thing provides no functionality without Internet connection.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Limited: The smart thing provides parts of its functionality without Internet connection.</td>
<td></td>
</tr>
<tr>
<td>Data Usage</td>
<td>Transactional: The smart thing processes data of transactions or interactions.</td>
<td>[2, 16, 20, 44–46, 92]</td>
</tr>
<tr>
<td></td>
<td>Analytical (basic): The smart thing processes data for descriptive purposes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Analytical (extended): The smart thing processes data for diagnostic, predictive, or prescriptive purposes.</td>
<td></td>
</tr>
<tr>
<td>Data Source</td>
<td>Thing state: The smart thing processes data about its internal condition.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thing context: The smart thing processes data about its physical environment.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thing usage: The smart thing processes data about its usage.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cloud: The smart thing processes external data primarily from the cloud.</td>
<td></td>
</tr>
<tr>
<td>Partner</td>
<td>User(s): The smart thing interacts with human users.</td>
<td>[4, 5, 22, 44, 47, 93]</td>
</tr>
<tr>
<td></td>
<td>Businesses(s): The smart thing interacts with businesses.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thing(s): The smart thing interacts with other smart things.</td>
<td></td>
</tr>
<tr>
<td>Multiplicity</td>
<td>One-to-one: The smart thing interacts with a single interaction partner.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>One-to-many: The smart thing interacts with many interaction partners.</td>
<td></td>
</tr>
<tr>
<td>Direction</td>
<td>Unidirectional: Data flows in one direction.</td>
<td>[2, 5, 26, 31, 34, 43, 45, 58]</td>
</tr>
<tr>
<td></td>
<td>Bidirectional: Data flows in all directions.</td>
<td></td>
</tr>
<tr>
<td>Autonomy</td>
<td>None: The smart thing cannot learn and needs an external trigger for any action.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Self-controlled: The smart thing cannot learn, but operates in an independent manner without external intervention to fulfil specific tasks. It needs external triggers.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Self-learning: The smart thing can act and decide in line with goals and improves over time by learning. It takes over decision-making tasks and adapt to a user’s needs and preferences. The smart thing acts without external intervention and does not require external triggers.</td>
<td></td>
</tr>
<tr>
<td>Acting Capabilities</td>
<td>Own: The smart thing directly interacts with the environment.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Intermediary: The smart thing relies on intermediaries to interact with the environment.</td>
<td></td>
</tr>
<tr>
<td>Sensing Capabilities</td>
<td>Lean: The smart thing collects simple data.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rich: The smart thing collects complex data.</td>
<td></td>
</tr>
</tbody>
</table>
Acting Capabilities

In addition to collecting data from the environment via sensors, smart things can also influence their environment through actuators [26]. We conceptualize acting capabilities by examining how smart things influence their environment (e.g., via audible signals, text, or voice messages). In contrast to the interaction layer, the acting dimension primarily focuses on a smart thing’s local level. As for the characteristics of this dimension, we follow Püschel et al. [27] who distinguish own and intermediary acting capabilities. The suitability of this distinction has been confirmed by the classification of our 200 real-life examples. Hence, we distinguish between smart things with integrated acting capabilities (e.g., the activity tracker vivosmart HR+ displays diverse fitness data on its screen) and smart things that rely on intermediary devices such as smartphones, tablets, or tools (e.g., the smart lock Lockitron sends notifications via smartphone) [34]. In line with the definition of the interaction layer, a smart thing can also act via intermediaries, for example, by using Bluetooth, local Wi-Fi, or Internet. Since smart things can have own and intermediary acting capabilities, both characteristics are non-exclusive.

Autonomy

While smart things can act autonomously in ever more scenarios, many neither learn nor act without external triggers [5]. We assign the autonomy level none to such smart things. For instance, the Situ food scale weighs food, displays the weight on its display, and sends information to the user’s smartphone. However, at its local level, the scale needs to be operated in a conventional way. In contrast, other smart things operate in a self-controlled manner such as the smart vacuum cleaner Roomba. This robot is equipped with sensors and software to autonomously scan and clean floors, operating in an independent way without external interference to fulfil a task unless exceptions occur [45]. More sophisticated smart things are even able to learn about their environment as well as adapt to their users’ needs and preferences [5]. Such self-learning smart things can act and decide in line with goals, without external interference
and in distinct cases even without external triggers. They refine internal model of themselves and their environment [45]. Self-learning smart things take over decision-making tasks from their users [4, 26, 43]. As an example, the smart security camera Butterflye learns the individual rhythm of a home and provides increasingly accurate alerts when something unexpected happens.

4.2 Interaction Layer

Also referred to as connectivity [47] or transmission [2] layer, the interaction layer focuses on a physical thing’s embedding into the digital world. For example, an Internet connection enables remote access to a smart thing’s sensing, acting, and computing capabilities [5, 47] as well as interactions among smart things [22]. Hence, the interaction layer covers the direction and multiplicity of interactions, and the partners with which smart things may interact.

Direction

Regarding the direction of interactions, a smart thing can participate in two mutually exclusive forms of interaction: unidirectional and bidirectional [4, 93]. Unidirectional interactions involve one-way flows of data. For instance, Babolat’s smart tennis racket collects data during a game, which is then forwarded to a user’s smartphone. In bidirectional interactions, two or more partners are actively involved, and data can flow in all directions between them. For example, the smart bracelet Olive, designed to reduce stress, uses haptic feedback (i.e., taps on the wrist) to alert the wearer when it detects elevated stress levels. Users can also tap the bracelet to register that they are in a good mood.

Multiplicity

Smart things can also be classified in terms of the number of interactions in which they engage – as one-to-one or one-to-many [5]. In one-to-one interactions, a smart thing connects to one interaction partner at a time [4]. For instance, the Lumo Run fitness tracker which attaches onto
sports clothes and displays diverse fitness parameters on the user’s smartphone, is designed for use by one person at a time. One-to-many interactions, in contrast, take place between multiple interaction partners. For instance, the EverSense Location-Sensing Thermostat can track the location of all members of a family via their smartphones, and adjusts the temperature of their house after the last person leaves and before the first person returns. Porter and Heppelmann [5] included many-to-many interactions in their analysis, yet, as our analysis takes the perspective of individual smart things, we excluded this characteristic from our taxonomy.

**Partner**

Smart things can interact with various partners. Bucherer and Uckelmann [22] identify consumers, things, businesses, and (information) service providers as actors. We do not account for service providers here, but consider them as businesses. Furthermore, Beverungen et al. [26] conceptualise smart things as boundary objects among businesses and consumers. Thus, we distinguish three types of interaction partners: users, things, and businesses. The concept of users as interaction partners is straightforward: the use of any smart thing by an individual serves as an example. Interactions among things are exemplified by the smart doorbell system Skybell: if the Nest smoke detector inside the house gives a smoke or carbon monoxide warning, the LED lights on the Skybell outside the house change to red, signalling that it is unsafe to enter. If Skybell detects motion, it triggers the Nest indoor camera, which starts monitoring from inside the home. Finally, the smart bracelet QMedic is an example of a smart thing’s interaction with a business: as anomalies in mobility or sleep are detected, a call centre is notified and, if required, in-home care can be provided. As a smart thing can interact with multiple partners, this dimension is non-exclusive [4].
4.3 Data Layer

Data is a key asset in the digital economy [1]. In the IoT context, data may even constitute a smart thing’s main value proposition [22, 26, 44, 94]. This is because smart things can locally process internal and external data through their computing logic and hence provide access to real-time data [44]. Accordingly, the data layer focuses on which data are used by smart things (source) – be it collected through local sensors or exchanged with external sources – as well as on how smart things use data (usage).

Source

As for the data source, we distinguish between thing state, thing context, thing usage, and cloud. Thing state refers to the local internal condition of a smart thing, covering data related to the thing such as identity, charging, or operating status [2]. The thing context refers to a thing’s physical environment (e.g., temperature or humidity), which is monitored via its sensors [2]. Thing usage describes the data generated and processed during a smart thing’s interactions with partners or usage. Ultimately, smart things may have access to external data (e.g., weather forecast or energy prices) or additional enterprise data (e.g., warranty status) stored in the cloud. This also holds for other external data sources, for example, if a smart thing can receive data directly from other smart things [44]. As a smart thing can process the input of different data sources, this dimension is non-exclusive [44]. The smart vacuum cleaner Roomba, for example, uses context data for coordination during the cleaning process. Furthermore, its cleaning settings can be adjusted by a user.

Usage

Data usage can be split into transactional and analytical categories. Transactional usage refers to the processing of individual transactions or interactions. For instance, the smart shower Eva Drop collects data about the water temperature and the person’s position in the shower and uses
these data to adjust the water flow. This way of data usage can be understood as reactive, referring to the smart thing’s ability to react to changes in its environment [45, 92]. In addition, data can be used for analytical purposes, which is commonly split into descriptive, diagnostic, predictive and prescriptive [5, 44, 46]. We refer to descriptive data usage as analytical basic, while diagnostic, predictive, and prescriptive data usage are categorised as analytical extended. Smart things with basic analytical data usage enable descriptive analytics. The smart tracking device Fitbit Charge tracks and displays the distance walked and calories burned by the user. Analytical extended data usage provides more sophisticated analytics, as in the case of the smart camera Nest Cam IQ, which can identify via facial recognition technology whether a person entering the house is a family member or a stranger.

4.4 Service Layer

With smart things driving the fusion of the physical and the digital world, the variety of enabled services is said to be “limited only by imagination” [44, p. 114]. Against this backdrop and in line with existing IoT stacks, the top-most layer of our taxonomy relates to the services associated with smart things [47]. This layer relates to whether a smart thing can provide offline functionality and examines central value proposition of smart things. Finally, this layer analyses services enabled by a smart thing’s integration into ecosystems [5].

Offline Functionality

Connectivity is vital for the functionality of smart things. In the context of the IoT, the literature defines connectivity rather broadly, i.e., with a focus on a smart thing’s general communication ability [34]. Rather than investigating specific communication technologies, we consider whether a smart thing’s functionality depends on a working Internet connection, as this influences in which services a smart thing can be involved. If a smart thing cannot provide any functionality without a working connection, its offline functionality is classified as none. The
WeMo insight switch, for example, requires a working connection to perform its two functions: to remotely control home appliances and track their energy use. If a smart thing provides parts of its functionality without a working Internet connection, it is classified as limited. For example, the smart light bulb LIFX works like an ordinary light if it is turned on or off at the switch. To access colours and other features, however, it needs an active Wi-Fi connection.

Value Proposition

From a product/service perspective and in line with the idea of smart things bridging the physical and the digital world, smart things consist of a physical underlying and a digital representation [95, 96]. That way, smart things serve as a distribution mechanism for digital services [91]. Against this background, smart things with a thing-centric and a service-centric value proposition must be distinguished. Smart things with a thing-centric value proposition primarily serve a thing-related purpose in the physical world extended by a digital representation and services [5]. An example is Babolat’s tennis racket that helps players improve their game via an app, analysing sensor data from the racket. Despite this valued-added service, the racket still retains its traditional functionality. Smart things with a service-centric value proposition also have a physical underlying, which primarily serves as a distribution mechanism. Related smart things cannot be used independently from related digital services. For example, Amazon’s Echo has no physical functionality but serves purely the provision of digital services.

Ecosystem Integration

A central priority in the development of smart things is to ensure that they not only capture data about their own performance and environment, but also capture data about other smart things [5]. This enables the rise of broader product systems, wherein related smart things collaborate, as well as the formation of systems of systems for linking and coordinating product systems.
We refer to a smart thing’s integration into such systems as ecosystem integration and classify this dimension in terms of three characteristics: none, proprietary, and open. The June oven, for example, related to the first characteristic, as it is a standalone smart thing not integrated into any broader contexts. The characteristic proprietary refers to smart things that can be integrated into an ecosystem, but which are only compatible with smart things from the same provider or manufacturer. For example, the smart camera Pivot is part of the Zmodo smart home ecosystem and can rotate to focus on an opened door or window when a Zmodo sensor is triggered. However, Pivot cannot interact with things from other providers. It exclusively works with Zmodo smart things and only can be controlled via the Zmodo app. Finally, open ecosystem integration enables interoperability, covering smart things that are compatible with other smart things across their own product family [44]. The Nest Thermostat, for instance, is equipped with an application programming interface (API) that allows information to be exchanged with products from different providers. For example, it can adjust the temperature when the smart Kevo lock reports that the door has been opened.

4.5 Classification Results and Examples

To evaluate the applicability of the taxonomy and prepare a dataset as input for the identification of smart thing clusters, two co-authors independently classified the sample of 200 smart things. We also validated the taxonomy by calculating hit ratios based on the classification results of both co-authors. Table 4 provides an overview of the dimension-specific hit ratios, whereby, 91% of the object-specific hit ratios exceed 75%. The classification results and therefore the agreement of the both co-authors confirm that the taxonomy is clear and meaningful regarding its dimensions and characteristics [57]. The results also show that smart things can be classified independently from their application domain, as our sample includes smart things from all application domains relevant in the B2C context [2].
Table 4: Dimension-specific Hit Ratios

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Hit Ratio</th>
<th>Dimension</th>
<th>Hit Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensing capabilities</td>
<td>96%</td>
<td>Data Source</td>
<td>93%</td>
</tr>
<tr>
<td>Acting capabilities</td>
<td>94%</td>
<td>Data Usage</td>
<td>89%</td>
</tr>
<tr>
<td>Autonomy</td>
<td>88%</td>
<td>Offline Functionality</td>
<td>95%</td>
</tr>
<tr>
<td>Direction</td>
<td>92%</td>
<td>Main Value Proposition</td>
<td>90%</td>
</tr>
<tr>
<td>Multiplicity</td>
<td>95%</td>
<td>Ecosystem Integration</td>
<td>86%</td>
</tr>
<tr>
<td>Partner</td>
<td>92%</td>
<td>Overall Hit Ratio</td>
<td>91%</td>
</tr>
</tbody>
</table>

Below, we outline the classification of two smart things from our sample to demonstrate the tangibility of our results and to illustrate the broad spectrum of smart things (Figure 2 and Figure 3). As a first example, the Situ food scale has lean sensing capabilities for weighing food. Using its own and intermediary acting capabilities, it shows the weight on its display and via an app. The Situ food scale is a non-autonomous device that needs a trigger for any action and that interacts with one user at a time. It provides analytical basic data usage featuring, for instance, the calculation of nutrients and calories. Without an Internet connection which enables transferring data to the app, the Situ Food scale can be used as a conventional scale. Thus, it has limited offline functionality. This complies with the scale’s thing-centric purpose. The scale is used in isolation from other smart things and not integrated in any ecosystem. As a second example, the indoor security camera Nest Cam IQ gathers vast amounts of data via rich sensing capabilities. It uses own acting capabilities (e.g., speaker) and intermediaries (via smartphone or tablet), and is a self-learning smart thing which can, over time, differentiate between family members and strangers and enable customized alerts. The Nest Cam IQ interacts with its users via microphone and has extensive capabilities to interact with other smart things. Apart from gathering data during usage and by monitoring its environment, it can access cloud data, including its video history. This enables extended analytical data usage, for example, facial recognition. Yet the Nest Cam IQ has no offline functionality and relies on its supervision service. It is compatible with smart things from the Nest ecosystem and from third providers.
Based on the cluster analysis, we identified five clusters that indicate which characteristics of smart things typically occur together. Below, we introduce each cluster along with constitutive characteristics and some examples. The clusters are also visualised in Figure 4. We also highlight the most frequent characteristics per dimension. For the non-exclusive dimensions,
we included all characteristics that cover more than one third of the sample. Appendix 3 shows to which cluster the smart things from our sample were assigned.

Figure 4: Composition of the Five Smart Thing Clusters

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Standalone Thing-Centric Executive</th>
<th>Connected Thing-Centric Performer</th>
<th>Standalone Service-Centric Monitor</th>
<th>Connected Service-Centric Partner</th>
<th>Self-Learning Service-Centric All-rounder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>38</td>
<td>40</td>
<td>51</td>
<td>32</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>19%</td>
<td>20%</td>
<td>26%</td>
<td>16%</td>
<td>20%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service</th>
<th>Ecosystem Integration</th>
<th>Value Proposition</th>
<th>Offline Functionality</th>
<th>Data Usage</th>
<th>Data Source</th>
<th>Partner</th>
<th>Multiplicity</th>
<th>Direction</th>
<th>Autonomy</th>
<th>Acting Capabilities</th>
<th>Sensing Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>None [34] (90%)</td>
<td>Thing-centric [35] (92%)</td>
<td>Limited [36] (95%)</td>
<td>Analytical (basic) [27] (71%)</td>
<td>Thing context [19] (50%)</td>
<td>User(s) [38] (100%)</td>
<td>One-to-one [37] (97%)</td>
<td>Unidirectional [22] (58%)</td>
<td>None [23] (61%)</td>
<td>Own [26] (68%)</td>
<td>Lean [33] (87%)</td>
</tr>
<tr>
<td></td>
<td>Open [20] (50%)</td>
<td>Thing-centric [39] (98%)</td>
<td>Limited [38] (95%)</td>
<td>Transactional [18] (45%)</td>
<td>Thing State [16] (40%)</td>
<td>Thing(s) [17] (43%)</td>
<td>One-to-many [38] (95%)</td>
<td>Bi-directional [23] (58%)</td>
<td>Self-Controlled [21] (53%)</td>
<td>Intermediary [37] (93%)</td>
<td>Lean [33] (83%)</td>
</tr>
<tr>
<td></td>
<td>None [47] (92%)</td>
<td>Service-centric [41] (80%)</td>
<td>None [47] (92%)</td>
<td>Analytical (basic) [35] (69%)</td>
<td>Thing context [24] (47%)</td>
<td>User(s) [50] (98%)</td>
<td>One-to-one [30] (59%)</td>
<td>Unidirectional [35] (69%)</td>
<td>None [28] (55%)</td>
<td>Intermediary [51] (100%)</td>
<td>Lean [36] (71%)</td>
</tr>
<tr>
<td></td>
<td>Open [23] (72%)</td>
<td>Service-centric [32] (100%)</td>
<td>None [32] (100%)</td>
<td>Transactional [17] (53%)</td>
<td>Thing Usage [39] (77%)</td>
<td>Thing(s) [23] (72%)</td>
<td>One-to-many [28] (88%)</td>
<td>Unidirectional [22] (69%)</td>
<td>Self-Controlled [17] (53%)</td>
<td>Intermediary [30] (94%)</td>
<td>Lean [31] (97%)</td>
</tr>
<tr>
<td></td>
<td>Open [21] (54%)</td>
<td>Service-centric [38] (97%)</td>
<td>None [38] (97%)</td>
<td>Analytical (extended) [22] (56%)</td>
<td>Thing context [22] (69%)</td>
<td>User(s) [31] (97%)</td>
<td>One-to-many [33] (85%)</td>
<td>Bi-directional [34] (87%)</td>
<td>Self-Learning [16] (41%)</td>
<td>Cloud [15] (39%)</td>
<td>Rich [35] (90%)</td>
</tr>
</tbody>
</table>

[...]: total number of smart things (…): relative number of smart things
To structure the clusters, we split them according to their value proposition, which was the first division performed by the cluster algorithm. We found that, in subsequent divisions, the clusters remained either almost exclusively thing- or service-centric. As such, we divided the clusters into two groups. The first group included Standalone Thing-Centric Executants and Connected Thing-Centric Performers. The second group contained Standalone Service-Centric Monitors, Connected Service-Centric Performers, and Self-Learning Service-Centric All-rounders. In both groups, we also found an increasing level of smartness as reflected by the dimensions data usage, autonomy, and ecosystem integration.

5.1 **Cluster 1: Standalone Thing-Centric Executant**

Standalone Thing-Centric Executants are everyday items with an immanent thing-centric purpose enhanced by digital services. Most related smart services are equipped with lean sensing capabilities (87%). They often provide own (68%) and always intermediary (100%) acting capabilities. The majority of Standalone Thing-Centric Executants are not autonomous (61%) and require an external trigger for any action. Interactions occur with one partner (97%) who is the smart thing’s user (100%). As for data usage, only basic data analytics such as descriptive statistics are provided (71%). In accordance with their thing-related purpose and functionality (92%), related smart things perform at least some of their functionality without an Internet connection (95%). Standalone Thing-Centric Executants are not integrated into ecosystems (90%).

Two examples are the Oral-B smart toothbrush and the smart fork Hapifork, which observe brushing or eating habits and offer real-time feedback via a smartphone. Independent from their digital services, they provide thing-related functionality as a toothbrush or a fork. Further, the analytical capabilities of both smart things are restricted to analytical basic data usage in the form of brushing statistics or eating habits. Finally, both smart things only interact with their users and are not integrated into any ecosystems.
5.2 Cluster 2: Connected Thing-Centric Performer

Connected Thing-Centric Performers are everyday items integrated in broader IoT contexts. Most have lean sensing capabilities (83%) as well as own (93%) and intermediary (93%) acting capabilities. They also have an immanent thing-centric purpose and functionality (98%). After an initial trigger, a small majority can perform in a self-controlled way without user interference (53%). The remaining 47% are not autonomous. Interactions typically occur with more than one partner (95%), mostly with users (98%) and other things (43%). Connected Thing-Centric Performers implement transactional (45%) or basic analytical (45%) data usage capabilities. Without an Internet connection, most related smart things can perform parts of their functionality (95%), a finding that corresponds with their thing-centric value proposition. Typically, Thing-Centric Performers are well-connected and can be integrated into ecosystems (68%). While some are only compatible with products from the same provider (18%), half (50%) of them can also interact with products from other providers.

Examples are Bosch’s I-Dos smart washing machine, with its automatic dosing system, and the smart refrigerator Smart Instaview, which includes features such as voice control. Both smart things have a traditional functionality bound to the physical product, and they exist independently from additional services. Both I-Dos and Smart Instaview can be controlled via an app, and may be integrated into ecosystems established by multiple associated partners such as Bosch Home Connect or Google Home.

5.3 Cluster 3: Standalone Service-Centric Monitor

Standalone Service-Centric Monitors are smart things that monitor their environment or users without being integrated into ecosystems. Most of these smart things possess lean sensing capabilities (71%) and intermediary acting capabilities (100%, only 22% also have own acting
capabilities). This implies that the thing itself is only the distribution mechanism of digital services and completely reliant on intermediaries such as smartphones to interact with users. The majority of Standalone Service-Centric monitors are not autonomous (55%) and need human triggers for any action. Others also function independently without user interference (43%). Most Standalone Service Centric Monitors interact with only one partner (59%), who is nearly always their user (98%). Data is collected while the thing is in use (77%) or gathered via the thing’s monitoring of the physical environment (47%). Data is mostly used in the basic analytical way (69%). Without an Internet connection, almost no functionality is available (92%). This is in line with the service-centric value proposition of Standalone Service-Centric Monitors (80%). Standalone Service-Centric monitors are not integrated into ecosystems (92%). Examples include the RunScribe smart fitness tracker and the Mon Baby device which monitors a baby’s sleep. Both devices consist of a clip, which is without own acting capabilities, that is attached to a running shoe or baby’s shirt, respectively. The clip monitors running performance or sleep, and enables analyses through the related app. Consequently, the smart thing itself serves as a distribution mechanism for digital services and is unusable without the app, which is provided on an intermediary device.

5.4 Cluster 4: Connected Service-Centric Partner

Connected Service-Centric Partners are smart things with notably well-developed connectivity and compatibility. Almost all have lean sensing capabilities (97%). More than half have own acting capabilities (53%), and the clear majority have intermediary (94%) acting capabilities. They are either self-controlled (53%) or not autonomous (44%). Most can interact with more than one partner (88%), and Connected Service-Centric partners do not only interact with users (97%) but also with other things (72%). Data is usually drawn from the thing’s context – i.e., its physical environment (69%), or is collected while the thing is in use (53%). More than half of the Connected Service-Centric Partners have transactional data-usage (53%) and thus are
restricted to processing data related to distinct transactions or interactions, whereas basic analytical data usage only occurs in just over one-third (34%). Without an Internet connection, no functionality is available (100%) and related smart things only serve as a vehicle for digital services (100%). Another feature of many related smart things is that they support an open ecosystem integration (72%).

Examples include the Wi-Fi connected garage control Garageio, which consists of a black box installed in a garage, and the Amazon Dash Wand, which is a connected Alexa-enabled home barcode scanner. Both devices solely serve as distribution mechanisms for services and are integrated into ecosystems, working with other smart things and services such as Amazon Echo.

5.5 **Cluster 5: Self-Learning Service-Centric All-rounder**

Self-Learning Service-Centric All-rounders are sophisticated smart things which feature high autonomy and are integrated into broader IoT contexts. They have rich sensing capabilities (90%), and usually have own (92%) and intermediary (97%) acting capabilities. Many Self-Learning Service-Centric All-rounders can take over decision-making tasks from users, at which they improve over time (41%). Despite the share of only 41% self-learning devices, this cluster includes 89% of all self-learning devices included in our sample of 200 smart things. Self-Learning Service-Centric All-rounders interact with users (100%) and other things (49%), and the majority can interact with more than one partner (85%). Data is collected from the thing’s environment (54%) or its usage and, in some cases, from the cloud (39%), which implies that Self-Learning Service-Centric All-rounders can processes further external or enterprise data. Most related smart things offer analytical extended data usage (56%) and thus support diagnostic, predictive, and/or prescriptive analytical purposes. Apart from this, their value proposition is service-centric (97%). Most related smart things can be integrated into different ecosystems (69%) and many are compatible with products from other providers (54%).
One example is the Nest smart thermostat, which learns and automatically adapts to its users’ behaviour by re-configuring itself. It also works with lights, locks, and other products, and is compatible with different providers. A second example is the smart camera, Cocoon, which learns the unique patterns of a home and improves over time to avoid false alarms.

5.6 Q-Sort Results

Finally, we used the Q-sort to evaluate the smart thing clusters. To yield comprehensive results, we split our analysis in two scenarios. In the first scenario, two co-authors who were not familiar with the results of the cluster analysis classified 20 smart things from the sample. They achieved an overall hit ratio of 90% [56] and a Cohen’s Kappa of 75% [88]. These results reflect substantial agreement [97]. In the second scenario, we asked 15 participants with an IoT background to accomplish the same task. They obtained an overall hit ratio of 78% and a Fleiss’ Kappa of 62% [89], which also mirrors substantial agreement [97]. The high hit ratios and Kappa coefficients corroborate the clusters’ validity and reliability.

In Figure 5, we present the cluster-specific hit ratios. The values on the diagonal represent the share of smart things per cluster that the Q-sort judges classified as proposed by the clustering algorithm. Four out of five hit ratios exceed 70%, whereas the fifth cluster (Connected Service-Centric Partners) is just below this level (69%). One classification error that caught our attention was the share of Connected Thing-Centric Performers which were classified as Connected Service-Centric Partners (27%). When tracing back the results, we noticed that this error was due to the Holmes Smart Heater, which is a Connected Thing-Centric Performer but, in seven cases, has been wrongly assigned to the Connected Service-Centric Partners. On further investigation, we found that the participants’ perception of the Holmes Smart Heater was wrong: as in each case, we had given the participants a short text based on public information and no pictures. Hence, they did not realise that the Holmes Smart Heater is a smart thing with an electric fan
and control buttons. Rather, they inferred that it is a connected device which serves as a temperature control and triggers other devices. As a result, they assigned it to the group of service-centric devices and classified it as a Connected Service-Centric Partner. In sum, however, most smart things were classified correctly.

Figure 5: Cluster-specific Hit Ratios

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Classification according to cluster analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standalone Thing-Centric Executant</td>
</tr>
<tr>
<td>Standalone Thing-Centric Executant</td>
<td>91%</td>
</tr>
<tr>
<td>Connected Thing-Centric Performer</td>
<td>0%</td>
</tr>
<tr>
<td>Standalone Service-Centric Monitor</td>
<td>9%</td>
</tr>
<tr>
<td>Connected Service-Centric Partner</td>
<td>0%</td>
</tr>
<tr>
<td>Self-Learning Service-Centric All-rounder</td>
<td>0%</td>
</tr>
</tbody>
</table>

6 Discussion and Outlook

The IoT has attracted considerable attention in recent years, as it affects our private and business lives. Although great potential is attributed to the IoT and research has already investigated topics such as technical foundations, business models, and use cases, smart things are treated as a black box in research and practice. However, understanding the characteristics of smart things beyond technical foundations is a prerequisite for tapping the potential of the IoT.

Aiming to take the lid of the smart thing black box, the theoretical contribution of our work is twofold: First, we developed a taxonomy of smart things that helps understand, describe, and analyse individual smart things. The taxonomy builds on and extends Püschel et al.’s [27] taxonomy in light of the latest developments in the IoT domain. In line with established IoT stacks [5, 31, 47], the taxonomy includes eleven dimensions structured according to four layers (i.e.,
thing, interaction, data, and service layer). The taxonomy has also been developed and validated empirically based on a sample of 200 smart things selected from all relevant IoT application domains in the B2C context [2]. Second, we identified and validated five smart thing clusters based on our sample. Each cluster represents a typical combination of characteristics. By evaluating the clusters with the Q-sort, we confirmed their robustness, clarity, and meaningfulness. Based on these results, we show that smart things available on the market need to be distinguished according to their characteristics. Two examples are the value proposition and smartness level of smart things. As for value proposition, we found that smart things can serve as foundation for digital services, some of which exceed the thing-related functionality by far and only require the physical thing as distribution mechanism. As for smartness, which is covered by dimensions such as autonomy or data usage, we found that smart things have different levels of smartness. This finding is supported by our cluster analysis where we found that clusters can be split in two groups according to their value proposition (i.e., thing-centric and service-centric) and that sub-groups can be built based on the smartness level. Both the taxonomy and the clusters corroborate that smart things should not be treated as a black box and that the set of feasible use cases depends on their characteristics. To the best of our knowledge, our study is the first to advance our understanding of the IoT by means of a well-founded and empirically validated taxonomy and related clusters. By enabling an in-depth classification of smart things and including several dimensions per layer, the taxonomy also operationalizes established IoT stacks [5, 31, 47].

Our findings not only advance our understanding of the IoT, but also stimulate future research. For example, the taxonomy and the clusters may serve as foundation for investigating the adoption of smart things, their affordances in different settings, and the role of smart things in broader contexts such as IoT ecosystems, systems of systems, and smart service systems. Our
results also inspire research on methods for the effective analysis and development of smart things and IoT solutions as well as for the quantification of the business value of such solutions. Our findings also have implications for practitioners (e.g., product developers, engineers, and managers) throughout the product development process. First, the clusters provide practitioners with a first high-level overview of different existing smart things and related possibilities. Then, the taxonomy can be used to operationalize which customer needs a smart product should address and which characteristics are required [98]. Second, practitioners can use the clusters and the taxonomy to compare already existing (smart) products (i.e., own ones or those of competitors) and to systematically explore potentials for further development. During the product development process, the taxonomy structures the design space that can be used to define the core features and potential use cases [98]. While the taxonomy and the clusters support the analysis and development of smart things and IoT solutions, they do not provide methodological guidance. As mentioned above, related methods must be developed in the future.

As any research, our work is beset with limitations. First, despite carefully selected 200 smart things from different application domains, our sample is restricted to a certain period and focuses on the B2C context. Given the fast-moving nature of the IoT, it is most likely that underrepresented characteristics will become common and that new characteristics will emerge. For example, other technologies such as Blockchain or Artificial Intelligence will influence the taxonomy and the clusters [99, 100]. Hence, our taxonomy should be reassessed in the future to account for new developments. Our sample is also restricted to the B2C context as much more information was publicly available here. Future research should challenge how the taxonomy needs to be extended to fit other contexts. A suitable sample should include smart things from the B2B context, e.g., related to smart factories or cyber-physical systems. Second, our taxonomy and the clusters take the perspective of individual smart things. On the one hand, an understanding of smart things is in high need and a necessary foundation. On the other hand, the
IoT will not unfold its full potential as long as individual smart things remain the centre of interest. Therefore, smart things should also to be investigated in broader contexts referred to as IoT ecosystems, systems of systems, or smart service systems in the future [5, 26, 44, 101].

1 References


