On the Security and Privacy of Internet of Things Architectures and Systems

Emmanouil Vasilomanolakis*†, Jörg Daubert*†, Manisha Luthra*,
Vangelis Gazis†, Alex Wiesmaier†‡ and Panayotis Kikiras†
* CASED / Telecooperation Lab,
Technische Universität Darmstadt
{manolis, joerg.daubert}@cased.de, manisha.luthra@stud.tu-darmstadt.de
† AGT International
{vgazis, awiesmaier, pkikiras}@agtinternational.com
‡ Department of Computer Science,
Hochschule Darmstadt

***Abstract*—The Internet of Things (IoT) brings together a multitude of technologies, with a vision of creating an interconnected world. This will benefit both corporations as well as the end-users. However, a plethora of security and privacy challenges need to be addressed for the IoT to be fully realized. In this paper, we identify and discuss the properties that constitute the uniqueness of the IoT in terms of the upcoming security and privacy challenges. Furthermore, we construct requirements induced by the aforementioned properties. We survey the four most dominant IoT architectures and analyze their security and privacy components with respect to the requirements. Our analysis shows a mediocre coverage of security and privacy requirements. Finally, through our survey we identify a number of research gaps that constitute the steps ahead for future research.**

I. Introduction

The Internet of Things (IoT) forms a dynamic global network infrastructure with self configuring capabilities, based on standard and interoperable communication protocols [56]. It represents the interconnection of numerous *things*—smart devices and services. Currently, more than a billion devices are connected to Internet, including PCs, embedded sensors, and mobile phones [32]. This present *Internet of smart devices* is moving towards the *Internet of Things*, and is expected to comprise 16 billion interconnected devices by the year 2020 [49].

IoT applications include domestic scenarios such as smart homes, mobility and transportation, but also industry scenarios such as smart manufacturing processes and smart energy grids. To reach such a level of diffuse and influence, holistic architectures for the IoT are required. Such architectures must cope not only with operational challenges, but also provide security and privacy, e.g., to comply with social acceptance [22]; society must trust that the IoT is handling such scenarios in a secure and privacy-preserving manner.

In this paper, we first identify properties for the IoT, by doing a comprehensive analysis of the related work, that combined make the IoT ecosystem unique compared to previous Information Technology (IT) infrastructures. With respect to these properties, we construct a number of security requirements. Furthermore, we identify the most dominant (in terms of their openness, the amount of research and industrial contributors, their coverage and impact, etc.) IoT architectures and conduct a comprehensive analysis by mapping them with the requirements. Finally, we compare the IoT architectures and highlight the research gaps as well as the necessary steps for a security and privacy complete IoT architecture.

The remainder of this paper is organized as follows: in Section II, we first identify properties that make the IoT unique in terms of the security and privacy challenges. Second, we propose a number of security and privacy requirements that take into account these IoT properties. Section III provides an overview of the IoT architectures as well as a security and privacy analysis of them with respect to our requirements. Section IV compares all IoT architectures with a focus on the fulfillment of the requirements. Finally, Section V concludes the paper and gives insights regarding current research gaps and possible future directions.

II. IoT Properties & Security Requirements

In this section, we identify the properties that constitute the uniqueness of the IoT in terms of the security and privacy challenges. Furthermore, we construct a number of security and privacy requirements, based on the aforementioned properties, and discuss them in detail.

A. IoT Properties

In contrast to traditional IT systems such as enterprise applications, cloud computing, and big data, a combination of a number of properties makes the IoT unique in terms of the challenges that need to be coped with. We identify these properties by analyzing related IoT research [2]–[4], [21], [23], [24], [35], [37], [38], [42], [54]. The identified distinguishing properties are four, namely: the *uncontrolled environment*, the *heterogeneity*, the need for *scalability*, as well as the *constrained resources* utilized in the IoT:

a) Uncontrolled environment: Many *things* will be part of a highly uncontrolled environment; *things* travel to untrustworthy surroundings, possibly without supervision. Sub-properties of the uncontrolled environment are: *mobility, physical accessibility,* and the lack of *trust*.

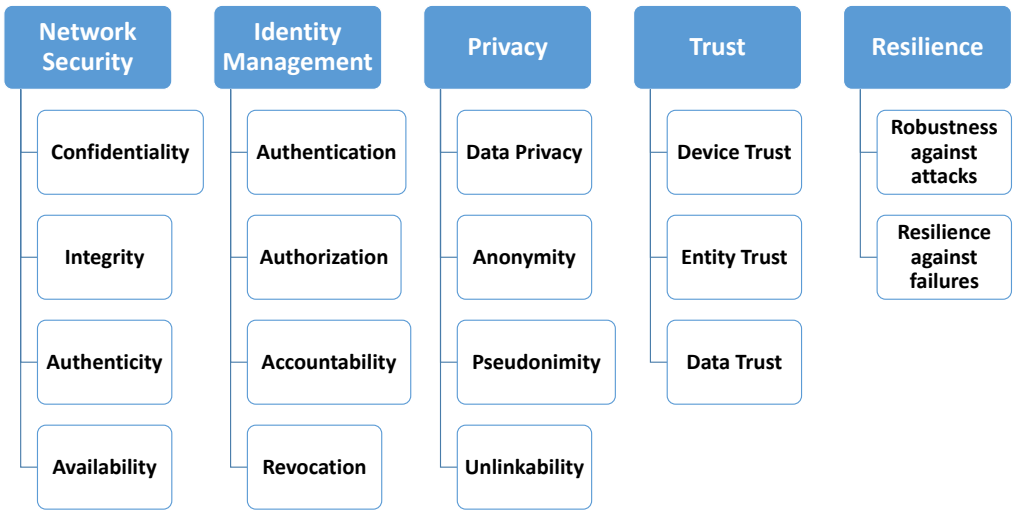

Fig. 1: Main security requirements and their subcomponents

- *Mobility:* Stable network connectivity and constant presence cannot be expected in such an environment.
- *Physical accessibility:* In the IoT, sensors can be publicly accessible, e.g., traffic control cameras, and environmental sensors.
- *Trust:* A priori trusted relationships are unlikely for the large amount of devices interacting with each other and users [42]. Thus, automated mechanisms to measure and manage trust of *things*, services, and users are crucial for the IoT.

b) Heterogeneity: IoT is expected to be a highly heterogeneous ecosystem as it will have to integrate a multitude of *things* from various manufacturers. Therefore, version compatibility, and interoperability have to be considered.

c) Scalability: The vast amount of interconnected things in the IoT demands highly scalable protocols. This also has an influence on security mechanisms. For instance, centralized approaches, e.g., hierarchical Public Key Infrastructures (PKIs), as well as some distributed approaches, e.g., pairwise symmetric key exchange schemes, cannot scale with the IoT.

d) Constrained resources: Things in the IoT will have constraints that need to be considered for security mechanisms. This includes energy limitations, e.g., battery powered devices, as well as low computation power, e.g., micro sensors. Thus, heavy computational cryptographic algorithms cannot be applied to all *things*.

B. Security Requirements for the IoT

Security and privacy are crucial enabling technologies and thus among the biggest challenges [1], [35], [43], [44], [47], [57] for the IoT. Therefore, it is compelling for the IoT architectures to consider and resolve these challenges upfront. Otherwise, applications as well as whole ecosystems building on top of such architectures may repeat the security fallacies of the past decades. For that, a precise understanding of security requirements in the context of the IoT is indispensable.

Prior technology trends, e.g., cloud computing and big data, are likely to share security requirements with the IoT. However, the uniqueness of the IoT introduces new challenges to security requirements, different from previous technology trends. Big data solutions for instance are designed to scale and deal with heterogeneity of data sources. Nevertheless, big data solutions are not required to deal with an uncontrolled environment and constrained resources; big data analytics run in isolated silos with time or resources to spare. Likewise, cloud computing by design is supposed to scale and overcome challenges of constrained resources. However, cloud computing hardly deals with mobility of devices and physical accessibility of sensors.

Related IoT security surveys are incomplete with respect to requirements. For instance, [11] provides a sound review of network security and identity management, but does not consider privacy, trust, and resilience; [53] emphasizes privacy and trust, but hardly tackles network security, identity management, and resilience. The requirement listing in [4] is the most extensive to the best of our knowledge. The analysis however only considers identity management.

To provide a comprehensive overview, we summarize security requirements from the domain of the IoT, but also related areas of IT and elaborate these requirements in the context of the properties of the IoT. For that, we split the requirements into five groups: *Network Security*, *Identity Management*, *Privacy*, *Trust*, and *Resilience*. The five main security requirements along with their subcomponents are shown in Fig. 1. Furthermore, Table I depicts the relationship between the various IoT properties and the security requirements. In a glance, it is shown that with regard to *network security* the constrained resources have the strongest connection, mainly due to the restrictions that they apply to traditional security mechanisms, e.g., cryptography. Moreover, *identity management* is influenced by the heterogeneity of the IoT. *Privacy* is mostly connected with scalability and the constrained resources as restrictions are posed to the technology candidates that can be utilized. Furthermore, the uncontrolled environment and the heterogeneity of the IoT have a serious impact on *trust*.

Lastly, *resilience* is directly connected to the need of the IoT for scalability.

1) Network Security: Network security requirements [46] can be split into *confidentiality*, *authenticity*, *integrity*, and *availability*. These apply to IoT architectures, e.g., by means of *things* connecting to *things* or services. However, properties of the IoT, e.g., constrained resources, must be considered.

The IoT requires architectures to deal with the heterogeneity of things. Interconnecting *things* may require *confidentiality*, e.g., to prevent eavesdropping sensitive information via Internet transmission. Technologies such as IPSec[31] and Transport Layer Security (TLS)[16] exist to fulfill this requirement. However, overhead may exceed the resource constraints of *things* and thus dedicated secure network stacks for the IoT exist [8]. *Authenticity* provides proof that a connection is established with an authenticated entity (cf. the following section). *Integrity* ensures no data is lost or modified undetected. While authenticity includes integrity, integrity alone can be required in the absence of authenticity to detect and recover failures. Existing mechanisms, e.g., TCP and TLS may suffice. However, IoT scenarios may require transactional integrity, e.g., critical infrastructures, and thus this should be considered by the architectures as well. *Availability* ensures that the connectivity of a *thing* or service persists even under link failures. Therefore, IoT architectures should ensure that link handover is possible.

2) Identity Management: Identity management poses a specific challenge in the IoT due to the number of devices, but also due to the complex relationship between devices, services, owners and users [36], [50]. Hence, specific attention has to be payed to *authentication*, *authorization* including *revocation*, and *accountability* or *non-repudiation*.

The mere quantity of devices in the IoT scenarios exceed the capabilities of direct authentication, e.g., a user provisioning many devices with her service credentials. Hence, methods to claim ownership and take control over devices are required.

Within the IoT scenarios, interactions may stretch across multiple domains. Scenarios for existing authorization solutions, e.g., Kerberos [48], assume a single domain that encloses devices, owners, users, and services. Thus, solutions for federated authorization that work with untrusted devices, allow delegation of access across domains, and provide quick revocation, e.g., for broken or rogue devices, are required.

Accountability ensures that every action is clearly bound to an authenticated entity. Accountability is a particular challenge in the IoT due to the magnitude of reuse of devices, services, and also data for many purposes. Thus, accountability must deal with huge amounts of entities, delegation of access, actions that span organizational domains, as well as continuous derivation of data.

3) Privacy: Privacy is considered to be one of the most dominant challenges in the IoT [36] due to the involvement of citizens and increasingly ubiquitous data collection, e.g., in smart home scenarios. A plethora of privacy definitions exist depending on the view of an IT solution. We briefly elaborate on *data privacy*, *anonymity*, *pseudonymity*, and *unlinkability*.

Data privacy complements confidential data transmission in the sense that a stored data record must not expose undesired properties, such as the identity of a person. This requirement is an enormous challenge in the IoT, as so many sensing devices collect personal information. Large amount of such data becomes Personally Identifiable Information (PII) when combined together; the data identifies a person [15]. Models to "anonymize" such data records exist [34], [51], [55], but have constantly proven to be insufficient. Moreover, models to protect this data privacy under data exchange between domains [18] are rather uncharted and complicated to apply.

Anonymity describes the property of a single person not being identifiable as the source of data or an action [41]. Anonymity is desirable in the IoT whenever a persons' identity is not required to comply to data minimization laws (Directive 95/46/EG [19]), as well as to dispel preconceptions that arise with data collection in the IoT. Achieving anonymity is a tough challenge as wearable and mobile devices may leak PII such as IP addresses and location unknowingly. Technologies such as anonymous credentials [9] and onion routing [17] exist, but may not scale well with the IoT.

Pseudonymity trades off anonymity with accountability. With pseudonymity, actions of a person are linked with a pseudonym, a random identifier, rather than an identity. Pseudonyms can serve many purposes [41], e.g., linking multiple actions of the same persons or providing graceful degradation of anonymity in the case of abuse. While pseudonyms may resolve privacy and accountability concerns in the IoT, standardized solutions that accompany multiple domains are required.

Unlinkability qualifies pseudonymity in the sense that specific actions of the same person must not be linked together. Unlinkability protects from profiling in the IoT. While pseudonyms may solve unlinkability, i.e., a different pseudonyms is used for every action, cross-implications with anonymity, in particular unknown meta-data, remain a challenge. Furthermore, some entity can always link every pseudonym to a person, and can thus also link all actions of that person.

4) Trust: Trust is another crucial requirement in the IoT due to the fact that it is highly distributed as well as dependable on qualitative data. Trust can be decomposed into *device trust*, *entity trust*, and *data trust* [15].

Device trust in the IoT is a challenge, as a priori trust in devices cannot always be established, e.g., due to high dynamics and cross domain relations. Hence, approaches such as trusted computing [26] (for standardized devices) as well as computational trust [29] are required to establish device trust. Moreover, every entity may assess trust in a device differently, hence IoT architectures have to deal with non-singular views of trust.

Entity trust in the IoT refers to expected behavior of participants such as persons or services. While device trust can be established via trusted computing, mapping such approaches to device trust, e.g., via behavioral attestation, is more challenging and experimental.

Data trust occurs in the IoT in a twofold manner: first, data originates from many and potentially untrusted devices. Hence, trusted data must be derived from untrusted sources, e.g., by applying data aggregation and machine learning techniques.

	Network Security	Identity Management	Privacy	Trust	Resilience
Uncontrolled Environment	•	•	•	• • •	•
Heterogeneity	•	••	•	••	•
Scalability	•	•	••	•	• • •
Constrained Resources	••	•	••	•	•

TABLE I: IoT properties and security requirements: the "•" symbols represent the level of influence in a scale from one (low) to $three$ (high).

Second, IoT services derivate new data, e.g., by integrating different types of data. For that newly generated data, a new trust assessment is required, e.g., via computational trust.

5) Resilience: The merge of scale of the IoT in terms of devices creates a large surface for attacks and failures. For this reason, *resilience* and *robustness* against attacks and failures apply, as important requirements, to the IoT.

Architectures must provide means to proficiently select *things*, transmission paths, and services according to their robustness (failure/attack *avoidance*). Furthermore, to ensure resilience, fail-over and recovery mechanisms must be provided to maintain operations under failure or attacks, and to return to normal operations (failure/attack *mitigation*).

III. IoT Architectures

The primary concept of the IoT is the pervasive presence of a variety of *things*, e.g., RFID tags, sensors, actuators, mobile phones, that are able to exchange and process information through Internet [33]. This triggers a need of controlling and monitoring of the data. An IoT architecture fulfills this responsibility by creating a bridge between the *things*, and the virtual *entities* (the Internet and associated services) [6], so that the data flow is consistent.

The following sections provide an overview of the existing research projects: *Internet of Things Architecture (IoT-A)* [27], *Building the environment for the Things as a Service (BeTaaS)* [6], *Open source cloud solution for the Internet of Things (OpenIoT)* [40] and *Internet of Things at Work (IoT@Work)* [28].

We selected these architectures as they were constructed during EU FP7 research projects and they are supported by a large number of academic research institutes as well as industrial partners. Thus, we expect these architectures to play a dominant role in future research as well as upcoming IoT solutions. Furthermore, the open nature of the aforementioned architectures suggests that they will be highly reused (either as a whole or partially). In fact, this has already been observed with the case of *IoT-A*. Therefore, a security analysis of them is essential. In addition, the selected IoT architectures provide a holistic surface coverage that includes generic environments (e.g., *IoT-A*), and corporate/industrial areas (e.g., *IoT@Work*).

All the architectures have been surveyed qualitatively, by comprehensively analyzing all the specification documents as well as any related research papers, and by mapping them to our requirements. Unfortunately, due to the fact that most of these proposals lack of available implementations, an extensive quantitative evaluation is not possible. OpenIoT is the only one of the surveyed architectures that provides an implementation. Thus, in this case we also investigated the level of consistency between the specification documents and the provided code.

A. IoT-A

IoT-A[27] is an architecture reference model developed with an EU FP7 project until 2013, with ongoing community development. This architecture uses the concepts of views and perspectives to guide the generation of architecture instances, from business goals via requirements. Such views and perspectives include the *information view* for static structures as well as dynamic information flows, the *performance and scalability perspective*, and the *trust and security perspective* [5]. The requirements are derived from a multitude of coarse-grained requirements (so called *unified requirements*) [14] based upon business goals, and then converted into fine-grained requirements for an architecture instance. The unified requirements are currently 38, addressing the security and privacy perspectives. In addition, IoT-A contains several models that are independent of particular architectures. These models include for instance the *communication model* and the *trust, security and privacy model*.

To address security requirements, IoT-A contains five logical security components [45]. Network security is addressed via the Key Exchange and Management (KEM) component. KEM manages cryptographic keys that are used for confidentiality as well as integrity in combination with authenticity. To handle resource constrained devices, KEM uses IP Security (IPSec) tunnels between (unconstrained) gateways as a well integrated concept to maximize the coverage of network security. However, the connections between constrained devices and the gateway remain unprotected. Furthermore, KEM does not address availability in the context of network connections. KEM also addresses functional requirements, such as lawful interception.

IoT-A contains three modules that address the requirements of identity management. The module Identity Management (IM) places focus on mere management, but does not cover a particular security requirement. The module Authentication (AuthN) covers the authentication requirements for users and services, as well as accountability with non-repudiation. The module Authorization (AuthZ) covers the authorization requirements for services via role-based access control (RBAC) as well as attribute-based access control (ABAC) [45]. Revocation depends on the particular access control model used. The authors are not aware of particular revocation schemes in IoT-A to the best of their knowledge.

A dedicated module called Pseudonymisation (PN) addresses privacy by means of providing pseudonymization for devices, users, and services. Pseudonyms replace real

identities, which are obtained from KEM, but still maintain coupling of identities and pseudonyms to ensure accountability. Pseudonyms can furthermore provide unlinkability, given a new pseudonym for every action is used. Complete anonymity as well as data privacy however are not addressed by PN. Still, AuthZ provides some means of access granularity that may solve data privacy to a certain extend.

The module Trust & Reputation (TRA) tackles the trust requirement for entity and device trust. In particular, the module describes the collection of the user reputation to calculate service trust. However, data trust does not appear to be addressed.

IoT-A describes the *fault handling* model, or functional group respectively. Requirements and measures of this model include *predicting potential failures*, *detecting existing failures*, *reduction of effects of failures* and *repairing the system*. Thus, the first measure addresses avoidance whereas the latter three address a life-cycle for mitigation.

B. BeTaaS

BeTaaS proposes an architecture for the IoT and Machine-to-machine (M2M) communication, to enable running applications over a local cloud of gateways. Each BeTaaS instance builds its own *cloud of gateways* that integrates various heterogeneous M2M systems in a seamless way. BeTaaS is founded on the Things as a Service (TaaS) reference model [7]. Modifying and augmenting the reference models of IoT-A (Sec. III-A), it provides architectural models for domains, information, communication, security, and functions.

The architecture comprises of four layers. First, the *Physical Layer* contains the M2M systems connected to the platform. Second, the *Adaptation Layer* handles the connection to the physical layer, abstracting from peculiarities of the individual M2M systems. The third layer, namely the *TaaS Layer*, relies on the abstraction layer and provides network-wide access to the devices in the M2M layer. Finally, the *Service* layer manages the functionalities and services of BeTaaS applications. At a glance, the BeTaaS architecture is addressing the security requirements by providing individual mechanisms for all of its layers except the physical one (which is implicit to the M2M/IoT systems).

With regard to *Network Security* the *Key Management* component associates entities, performs authentication, manages user sessions, and provides encrypted communication. Since BeTaaS instances consist of multiple gateways, BeTaaS uses a PKI with a Certificate Authority (CA) to manage keys and ensure confidentiality, authenticity and integrity via secure communication channels. BeTaaS also covers cases with multiple involved organizations, e.g., external entities that are not governed by the internal CA. Such cross-organization key management is handled by the BeTaaS directory service. Furthermore, BeTaaS addresses resourced constrained devices by applying computationally efficient cryptographic schemes such as Elliptic Curve Cryptography (ECC) [30].

For *Identity Management*, BeTaaS provides *authentication* via a dedicated architectural component. For that, it distinguishes two cases: gateway level authentication, e.g., when a gateway joins a BeTaaS instance, and application or service level authentication, e.g., when a user utilizes an application. For the first case, the authentication module uses key management, whereas for the latter case OAuth can be adapted for authentication and authorization. *Authorization* is covered by a dedicated component as well [13]. The *accountability* requirement remains unclear.

While *Privacy* is stated as a key aspect of the security mechanisms in BeTaaS [6], there is no evidence of how this requirement is fulfilled. The identity management component is responsible for managing the identities of sensors and gateways, but data anonymity or pseudonymity are not discussed.

Trust is handled by the *trust and reputation* component. The model retrieves input from individual trust aspects: *security mechanisms* (which for instance include information regarding the encryption algorithms, the certificates, etc.), *QoS fulfillment, dependability performance, battery load* and *stability in provided data*. These trust aspects are then aggregated to compute the final trust value.

Lastly, the aspect of *resilience* is handled via four different pillars: fault prevention, removal, tolerance and forecasting. The *Failure Analysis Approach* [13] component is responsible for the identification of potential causes of failures and for providing solutions to properly manage them. A process named *Failure Modes Effects and Critically Analysis* is performed on the functional items of the system. First, the fault modes for each IoT device are identified and corresponding effort on the analysis and operations is computed. Moreover, after assessing the probability of failure occurrence, it assigns the criticality of the failure. At this point, the *Reliability Architectural Approach* component proposes solutions for overcoming the possible system failure, with respect to the aforementioned analysis.

C. OpenIoT

The EU FP7 OpenIoT research project (2012-2014), has introduced an IoT architecture [39], [40]. OpenIoT is based on IoT-A's (cf. Section III-A) defined Architectural Reference Model (ARM). It adopts the main ARM concepts and functional building blocks. However, OpenIoT concentrates on providing a *cloud-based middleware* infrastructure, to deliver an on-request access to the IoT or the IoT services, which could be formulated over multiple infrastructure providers like cloud-based ones [40]. OpenIoT also offers an open source implementation[1] that focuses on structuring principles for the IoT applications with cloud-based characteristics such as *on-demand* or *pay-as-you-go* service delivery. At a glance, the architecture deals with IoT/cloud convergence.

The OpenIoT architecture specification [25] describes two security modules: the security & privacy module as well as the trustworthiness (trust) module. Within the the security module, one submodule addresses secure messaging, another one authentication and authorization. Opposed to the specification, privacy features are not present in the public code. The trustworthiness module evaluates the trustworthiness of input sensor data (data trust).

OpenIoT relies on the HTTP with TLS protocol to ensure secure and encrypted messaging. Resource constrained devices, e.g., IEEE 802.15.4 (ZigBee), are partially addressed

[1] https://github.com/OpenIotOrg/openiot

as well, via IPSec tunnels established by gateways to ensure confidentiality, integrity and authenticity. Availability is not mentioned in the context of network security.

For identity management, OpenIoT uses a centralized *security and privacy* module that provides authentication and authorization based on OAuth. To control authorization, the RBAC model is used. The fulfillment of further requirements, e.g., accountability, remains unclear.

Despite the name "security & privacy", privacy requirements seem not to be addressed.

The *trust* module is an independent module in OpenIoT. The trust module addresses the requirements of both, data trust and device trust. To obtain device trust, OpenIoT uses spatial correlation of sensors, i.e., close sensors in a similar environment should produce similar sensor readings. Once device trust is established, data records can be annotated with trust labels as well. However, entity trust remains unclear.

OpenIoT does not address robustness in terms of failure avoidance, but rather places the focus on resilience in terms of mitigation. For that, OpenIoT maintains an up-to-date inventory of entities and dynamically restructures the dependencies between entities, e.g., reconnects a service to another sensor in case of sensor failure [52]. Thus, fail-over and recovery are integral parts of OpenIoT.

D. IoT@Work

IoT@Work is a European Commission FP 7 project completed in 2013, with the goal of establishing an IoT architecture for the industrial automation domain [28]. The dominant requirements were interoperable and reliable network communication, auto-configuration, as well as security. For that, IoT@Work introduces for instance the concept of network slices, a combination of virtualization, resource management, and security. A network slice is an abstract layer in between the physical view, e.g., network technology and devices, and the application view.

IoT@Work is handling network security via commonly used technologies. Extensible Authentication Protocol (EAP) as an IEEE 802.1X [12] implementation ensures authentication in the low network layer, e.g., for switch ports. EAP-TLS also ensures confidentiality. The concept of network slices allows for network virtualization, and thus fast network link fail-over to protect availability. While device integrity is addressed by IoT@Work, the authors are not aware of network integrity mechanisms.

Authentication is mainly provided by network security in IoT@Work. Furthermore, authorization is realized via Capability-Based Access Control (CBAC) with support for delegation, accountability, and revocation. CBAC works well with many entities as well as under connection failure to the central authorization service. However, alternative schemes such as ABAC, which provide more fine-grained authorization, are not supported.

Privacy is not a driving requirement category for IoT@Work due to the industrial automation focus. However, some data privacy is provided via the modeling of granularities in access capabilities. Furthermore, the access delegation approach can be used for pseudonymous access by delegating capabilities to a pseudonym. Hence, entities can control the desired level of unlinkability by themselves. However, no explicit support for unlinkability is given. Anonymity can be achieved by proving capabilities through Zero Knowledge Proofs (ZKPs). Thus, no identifiers have to be shown to access a device or service.

To the authors' best of knowledge, IoT@Work does not tackle trust-based requirements.

Resilience is a core requirement for IoT@Work, with focus on failure handling. The network slice approach uses virtual network links that are robust against failures. In addition, live-reconfiguration is possible and thus allows for recovery in the sense of resilience. Still, IoT@Work maintains a strong network focus and hardly considers devices and services.

IV. Comparative Analysis of IoT Architectures

In this section, we compare the aforementioned IoT architectures, presented in Section III, with respect to the security requirements (as introduced in Section II). This comparison attempts to provide guidelines for selecting an architecture that fulfills certain requirements. Furthermore, our analysis points out gaps with respect to security requirements for the IoT architectures in general.

A. Network Security

For *network security*, all four architectures address (at least partially) *confidentiality* as well as *integrity* in combination with *authenticity* as dominant requirements. However, *availability* only seems to be considered to a limited extend by the architectures.

IoT-A as well as BeTaaS place strong focus on key and credential management in terms of PKIs and key exchange to ensure confidentiality as well as authenticity. BeTaaS explicitly distinguishes internal and external entities, which is a benefit for open ecosystems. IoT-A as well as OpenIoT favor a combination of established security mechanisms, such as IPSec and TLS. OpenIoT also addresses resource constrained devices via secure ZigBee communication standards. IoT@Work highly deviates from other architectures. It provides strong focus on authenticity, even for low-level network access, as well as availability in terms of network virtualization and link fail-over. However, IoT@Work hardly considers confidentiality as important as the other architectures. Hence, IoT@Work suits restrained and well defined infrastructures best.

B. Identity Management

Identity management is an essential part of the IoT concept. Therefore, all architectures consider this as a requirement and provide various mechanisms to, at least partially, fulfill it.

IoT-A is focusing on mechanisms that provide authentication and accountability for both users and services. In addition, authorization for services is considered. The BeTaaS architecture, in the context of identity management, is concentrating on authentication and identification. Nevertheless, other aspects of identity management, e.g., accountability, are not addressed. OpenIoT works on the Central Authentication Server (CAS) approach to achieve identity management. It is shown [43] that,

via a centralized IoT architecture, this challenge is inherently simpler than with distributed approaches. The access control policies are also easier to manage with a central entity [25]. Finally, IoT@Work provides authenticity (as discussed in IV-A), accountability, via a persistent storage for verifying the credentials, and an interface to the Credential Management Service, for latest access and updates, and authorization.

C. Privacy

Even though *privacy* is one of the main requirements that helps in terms of social acceptance, not all of the investigated IoT architectures provide mechanisms to ensure it.

IoT-A partially deals with privacy by pseudonymizing data. Its pseudonymization coupled with data privacy policies, e.g., access control policies, provides a fair, yet incomplete, fulfillment. BeTaaS inherits from the high level abstraction reference model of IoT-A. Thus, similarly to IoT-A, access control mechanisms enforce data privacy by restricting unauthorized access. The identity management component is responsible for managing the way identities of sensors or gateways are presented in their interaction with BeTaaS instances. Nevertheless, not much information is given regarding this. Apart from data privacy being maintained by centralized access control, data anonymization and pseudonymity is not elaborated in OpenIoT [25], [40], [52]. IoT@Work indirectly deals with privacy by giving control to entities in terms of unlinkability of their data as well as by providing some anonymization capabilities.

D. Trust

Trust is crucial, in particular for assessing the trustworthiness of sources of information. Nonetheless, the surveyed architectures handle this requirement with a plethora of different ways and not always on the same degree.

IoT-A focuses on trust on the application level only. In more details, the TRA component is responsible for establishing the trust to the *things* and compute reputation values based on the recommendations and the feedback received from other *things* and services. BeTaaS is handling trust via a specific component tailored for this purpose. With this it is possible to generate trust values for the various *things*. The centralized nature of the architecture minimizes the need for *device trust* in OpenIoT. *Data trust* is provided by computing the trustworthiness of the data using a spatial correlation algorithm [25]. The IoT scenarios described in IoT@Work do not introduce a need to deal with trust issues, so the model does not provide any mechanisms to cope with trust [20].

E. Resilience

All of the discussed architectures consider resilience as an important block of requirements. All four architectures address robustness as well as resilience, with a slight focus shift towards resilience in the case of IoT@Work.

IoT-A and BeTaaS present the most elaborate approaches for resilience: IoT-A contains a fault handling model that addresses all resilience lifecycle phases explicitly, including the prediction of faults. BeTaaS uses established analytical methods to identify critical components in order to direct efforts on these components. Compared, OpenIoT and IoT@Work contain ready-to-use mechanisms to increase resilience. In the case of OpenIoT, the dynamic adjustment of information flows provides resilience. In the case of IoT@Work, the concepts of network slices and virtualization achieve the same goal closer to the hardware level.

F. Summary

Table II summarizes our findings regarding the four IoT architectures and the security requirements. It becomes evident that each architecture has a specific focus area. For instance, IoT@Work works best for the manufacturing domain whereas OpenIoT works best as an open sensor and service marketplace. IoT-A and BeTaaS seem to fulfill most requirements well. However, these two architectures represent rather architecture frameworks than architectures and the actual implementation remains up to the user.

V. Conclusion

In this paper, we highlighted the specific properties of the IoT compared to other technological trends. We devised a comprehensive list of security and privacy requirements from these properties to establish a standard set of security requirements for the IoT technologies. To the best of our knowledge, this is the most comprehensive list published in the domain so far.

A. IoT Architectural Gaps

Based upon these requirements, we analyzed the security features of the four most dominant IoT architectures, namely the IoT-A, BeTaaS, OpenIoT, and IoT@Work. While all of these architectures seem to fulfill some of the requirements (cf. Table II), several gaps remain: first, network security often only addresses a part of the data transmission, typically from a gateway to cloud infrastructure. However, data transmission within cloud infrastructure as well as with resource constrained devices, e.g., sensor networks connected to a gateway, should be considered as well. Second, most identity management solutions focus on an enclosed domain, but lack inter-domain identity management capabilities. Third, access control models such as CBAC and RBAC become hard to control in very large installations, and grant unnecessary access due to the lack of context. Fourth, the acceptance of privacy and trust mechanisms in the IoT seems to be limited. Today, privacy is mostly based on fine-grained access control rather than privacy enhancing technologies. Lastly, trust seems to be limited to one mechanism per architecture—either hard cryptographic trust, ratings, or spatial correlation. However, modern computational trust mechanisms are capable of integrating an ensemble of different trust technologies altogether.

B. Future Work

With regards to future work, we recommend to address the major gaps that were identified in specific areas of the identity management, privacy, and trust. Within *identity management*, better mechanisms to provide accountability are required. Common mechanisms combine digital signatures (non-repudiation) together with logs. Such mechanisms however provide no privacy protection due to the digital signatures. For a better solution, we recommend to consider mechanisms such

	IoT Architectures			
Requirements	IoT-A	BeTaaS	OpenIoT	IoT@Work
Network security				
...Confidentiality	✓	✓	✓	✓
...Integrity	✓	✓	✓	✗
...Authenticity	✓	✓	✓	✓
...Availability	✗	✗	✗	≈
Identity management				
...Authentication	✓	✓	✓	✓
...Authorization	✓	✓	✓	✓
...Accountability	✗	✗	✗	✓
...Revocation	✓	✗	✗	✓
Privacy				
...Data privacy	≈	✗	✗	≈
...Anonymity	✗	✗	✗	✓
...Pseudonymity	✓	✗	✗	✓
...Unlinkability	✓	✗	✗	✗
Trust				
...Device trust	✓	✓	✓	✗
...Entity trust	✓	✗	✓	✗
...Data trust	✗	✓	✗	✗
Resilience				
...Robustness	✓	✓	✗	≈
...Resilience	✓	✓	✓	≈

TABLE II: IoT architectures and security requirements: "✓" indicates fulfillment, "✗" no fulfillment or missing evidence, and "≈" a partial fulfillment.

as blind signatures in combination with threshold cryptography [10]. With such a mechanism, digital signatures ensure accountability, and revocable pseudonymity is provided as no single entity can link the signature to an identity. Thus, privacy and identity management requirements can indeed be balanced. With respect to *privacy*, we plan to propose a framework for its protection at the device, communication, and cloud level rather than only at one of these levels. Anonymity and pseudonymity for instance should be already addressed at the device-level to prevent the leakage of sensitive information as soon as possible. Finally, with regards to *trust*, the discussed systems only provide crude reputation mechanisms. However, to fully realize the marketplace concept envisioned for the IoT, a real *community of trust* is required. Such a community should encapsulate concepts like *transitive trust*, e.g., in the context of "if a trusted entity of mine trusts another entity, I also trust this entity" rather than relying on one singular view of trust.

REFERENCES

[1] Mohamed Abomhara and Geir M. Koien. Security and Privacy in the Internet of Things : Current Status and Open Issues. In *Privacy and Security in Mobile Systems (PRISMS)*, pages 1–8. IEEE, 2014.

[2] Ahmad W Atamli and Andrew Martin. Threat-Based Security Analysis for the Internet of Things. In *Secure Internet of Things (SIoT)*, pages 35–43. IEEE, 2014.

[3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805, October 2010.

[4] Sachin Babar, Parikshit Mahalle, Antonietta Stango, Neeli Prasad, and Ramjee Prasad. Proposed security model and threat taxonomy for the Internet of Things (IoT). In *International Conference on Network Security & Applications (CNSA)*, volume 89, pages 420–429. Springer Berlin Heidelberg, 2010.

[5] Martin Bauer and Sebastian Lange. *Enabling Things to Talk*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[6] BETaaS Consortium. Building the environment for the things as a service. http://www.betaas.eu/, 2012. [11. Feb. 2014].

[7] BETaaS Consortium. D1.4.2 – TaaS Reference Model. http://www.betaas.eu/docs/deliverables/BETaaS%20-%20D1.4.2%20-%20TaaS%20Reference%20Model%20v1.0.pdf, October 2013. [11 Mar. 2014].

[8] Riccardo Bonetto, Nicola Bui, Vishwas Lakkundi, Alexis Olivereau, Alexandru Serbanati, and Michele Rossi. Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples. *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2012 - Digital Proceedings*, 2012.

[9] Jan Camenisch and Els Van Herreweghen. Design and implementation of the *idemix* anonymous credential system. In Vijayalakshmi Atluri, editor, *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, pages 21–30. ACM, 2002.

[10] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982.*, pages 199–203. Plenum Press, New York, 1982.

[11] Simone Cirani, Gianluigi Ferrari, and Luca Veltri. Enforcing security mechanisms in the IP-based internet of things: An algorithmic overview. *Algorithms*, 6(2):197–226, 2013.

[12] P. Congdon, B. Aboba, A. Smith, G. Zorn, and J. Roese. IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines. RFC 3580 (Informational), September 2003. Updated by RFC 7268.

[13] BETaaS Consortium. BETaaS Building the Environment for the Things as a Service D2 . 2 . 2 – Specification of the extended capabilities of the platform. pages 1–61, 2014.

[14] IoT-A Consortium. Iot-a unified requirements. http://www.iot-a.eu/public/requirements/. [31 Jan. 2014].

[15] Joerg Daubert, Alexander Wiesmaier, and Panayotis Kikiras. A view

on privacy & trust in iot. In *IOT/CPS-Security Workshop, IEEE International Conference on Communications, ICC 2015, London, GB, June 08-12, 2015*, page to appear. IEEE, 2015.

[16] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878, 6176.

[17] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The second-generation onion router. In Matt Blaze, editor, *Proceedings of the 13th USENIX Security Symposium, August 9-13, 2004, San Diego, CA, USA*, pages 303–320. USENIX, 2004.

[18] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 371–380. ACM, 2009.

[19] European Parliament and Council of the European Union. On the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal*, L(281):0031–0050, 1995.

[20] Kai Fischer, Jurgen Gessner, and Steffen Fries. Secure Identifiers and Initial Credential Bootstrapping for IoT@Work. *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 781–786, July 2012.

[21] G A N Gang, L U Zeyong, and Jiang Jun. Internet of Things Security Analysis. In *Internet Technology and Applications (iTAP)*, pages 1–4. IEEE, 2011.

[22] Lingling Gao and Xuesong Bai. A unified perspective on the factors influencing consumer acceptance of internet of things technology. *Asia Pacific Journal of Marketing and Logistics*, 26(2):211–231, 2014.

[23] Vangelis Gazis, Carlos Garcia Cordero, Emmanouil Vasilomanolakis, Panayotis Kikiras, and Alex Wiesmaier. Security Perspectives for Collaborative Data Acquisition in the Internet of Things. In *International Conference on Safety and Security in Internet of Things*. Springer, 2014.

[24] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, September 2013.

[25] Robert Gwadera. D5.2.1 Privacy and Security Framework. 2013.

[26] Alexander Iliev and Sean W. Smith. Protecting client privacy with trusted computing at the server. *IEEE Security & Privacy*, 3(2):20–28, 2005.

[27] IoT-A Consortium. IoT-A – Internet of Things Architecture. http://www.iot-a.eu/. [27 Jan. 2014].

[28] IoT@Work Consortium. D1.2 – Final framework architecture specification. https://www.iot-at-work.eu/data/D1.3_IoT@Work_Architecture_final_v1.0-submitted.pdf, July 2013. [20 Jul. 2014].

[29] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[30] Vivek Kapoor, Vivek Sonny Abraham, and Ramesh Singh. Elliptic curve cryptography. *ACM Ubiquity*, 9(20):20–26, 2008.

[31] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401 (Proposed Standard), November 1998. Obsoleted by RFC 4301, updated by RFC 3168.

[32] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. Future internet: The internet of things architecture, possible applications and key challenges. In *10th International Conference on Frontiers of Information Technology, FIT*, pages 257–260. IEEE, 2012.

[33] Oleg Logvinov, Bruce Kraemer, Chuck Adams, Juergen Heiles, Gary Stuebing, Mary Lynne Nielsen, and Brenda Mancuso. Standard for an Architectural Framework for the Internet of Things (IoT) IEEE P2413 Webinar Panelists. pages 1–12, 2014.

[34] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. *L*-diversity: Privacy beyond *k*-anonymity. *TKDD*, 1(1), 2007.

[35] Anthéa Mayzaud, Rémi Badonnel, and Isabelle Chrisment. Monitoring and Security for the Internet of Things. In *International Conference on Autonomous Infrastructure, Management, and Security*, pages 37–40. Springer, 2013.

[36] Carlo Maria Medaglia and Alexandru Serbanati. An overview of privacy and security issues in the internet of things. In *The Internet of Things*, pages 389–395. Springer, 2010.

[37] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, September 2012.

[38] Huansheng Ning, Hong Liu, and Laurence T Yang. Cyberentity Security in the Internet of Things. *Computer*, 46(4):46–53, 2013.

[39] OpenIoT Consortium. OPENIoT D2.3 Detailed Architecture and Proof-of-Concept Specifications. http://openiot.eu/?q=node/49, 2013.

[40] OpenIoT Consortium. OPENIoT project description. http://www.openiot.eu/, 2013.

[41] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity - A proposal for terminology. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA, July 25-26, 2000, Proceedings*, volume 2009 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 2000.

[42] R Roman, P Najera, and J Lopez. Securing the internet of things. *Computer*, 44(9):51–58, 2011.

[43] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266–2279, July 2013.

[44] Sadeghi, Ahmad-Reza, Christian Wachsmann, and Michael Waidner. Security and privacy challenges in industrial internet of things. In *Annual Design Automation Conference*, page 54. ACM, 2015.

[45] Alexander Salinas, Yosra Ben Saied, and Dissemination Level. Internet-of-Things Architecture Concepts and Solutions for Privacy and Security in the Resolution Infrastructure. (257521), 2013.

[46] Günter Schäfer. *Security in fixed and wireless networks - an introduction to securing data communications*. Wiley, 2003.

[47] Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. Security, privacy and trust in internet of things: The road ahead. *Computer Networks*, 76:146–164, 2015.

[48] Jennifer G. Steiner, B. Clifford Neuman, and Jeffrey I. Schiller. Kerberos: An authentication service for open network systems. In *Proceedings of the USENIX Winter Conference. Dallas, Texas, USA, January 1988*, pages 191–202. USENIX Association, 1988.

[49] H Sundmaeker, P Guillemin, and P Friess. *Vision and challenges for realising the Internet of Things*. Number March. 2010.

[50] Hui Suo, Jiafu Wan, Caifeng Zou, and Jianqi Liu. Security in the internet of things: a review. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, volume 3, pages 648–651. IEEE, 2012.

[51] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[52] Specific Targeted. D2 . 3 OpenIoT Detailed Architecture and Proof-of-Concept Specifications. 2011.

[53] Jari Veijalainen, Denis Kozlov, and Yasir Ali. Security and Privacy Threats in IoT Architectures. In *Proceedings of the 7th International Conference on Body Area Networks*, number International Conference on Body Area Networks, pages 256–262. ACM, 2012.

[54] Rolf H. Weber. Internet of Things – New security and privacy challenges. *Computer Law & Security Review*, 26(1):23–30, January 2010.

[55] Xiaokui Xiao and Yufei Tao. M-invariance: towards privacy preserving re-publication of dynamic datasets. In Chee Yong Chan, Beng Chin Ooi, and Aoying Zhou, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, June 12-14, 2007*, pages 689–700. ACM, 2007.

[56] Andrea Zanella, Nicola Bui, Angelo P Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 2014.

[57] Zhi-kai Zhang, Michael Cheng, Yi Cho, and Shiuhpyng Shieh. Emerging Security Threats and Countermeasures in IoT. In *ACM Symposium on Information, Computer and Communications Security*, pages 1–6. ACM, 2015.