Multi-Tenant IoT Service Management towards an IoT App Economy

Marc-Oliver Pahl
Technische Universität München, pahl@net.in.tum.de

Abstract—**Orchestrating the Internet of Things (IoT) is complex. Service modularization can reduce this complexity. Therefore, service-oriented management is a promising paradigm. For reasons such as privacy and performance, managing services site-locally is very suitable for the IoT. The desired service-oriented site-local IoT management requires a suitable service management. However, due to the lack of globally connected IoT deployments, there is little research regarding service management in our envisioned setting. We identify requirements for a service-oriented site-locally deployed IoT with central global App stores. Such a management architecture will become the basis for an IoT App economy. We focus on a multi-tenant setting on all levels: development, distribution, and running.**

Index Terms—**Internet of Things, IoT, Service Management, Services, Edge-Based, Site-local Management**

I. Introduction

The Internet of Things (IoT) consists of many distributed heterogeneous devices. They are managed by software services. To manage the IoT's complexity, service-orientation is a promising paradigm. Complex applications get modularized into smaller *microservices* that implement reusable building blocks. So-called orchestration services dynamically compose other services for implementing complex scenarios [1], [2]. An example heating control services could connect dynamically to locally available gateway services to the available heaters.

For many reasons most currently deployed IoT systems run in the cloud. Mainly because of lower latency, research is moving IoT management from the cloudy Internet center towards its edges. Edge computing, and fog computing are the corresponding paradigms [3]. For reasons such as privacy and performance, this development can be pushed further towards management directly within local IoT sites [2], [4].

Site-local IoT management research enables the cooperation of so-called microservices [1], [3], [5]. Several (micro-) services run within local IoT smart spaces. These setups are connected to the Internet as a whole. A service-oriented IoT requires the management of many services that mash up dynamically. As this paradigm is on the raise, we consider it a good time to examine service management in a visionary IoT setting with site-local IoT management.

For both, edge-based and site-local IoT, little research is currently happening towards *full life-cycle service management*. Instead, IoT service management research typically focuses on placement in specialized systems [6], [7] and constraint node

Support: German Federal Ministry of Economic Affairs and Energy (BMWI) in DECENT (0350024A); German-French Academy in SCHEIF.

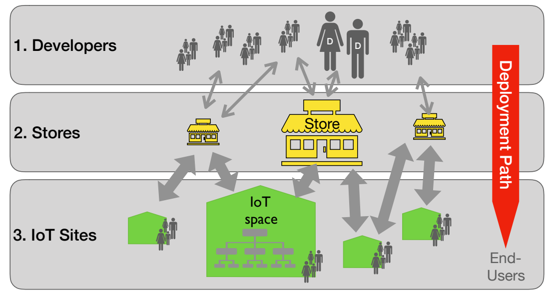

Fig. 1. Three Tier IoT Service Management Setting.

management [8]. A cause is that the middleware research did not converge yet, resulting in the missing of a common runtime environment for IoT services.

The ultimate goal for a service-oriented IoT is establishing an App ecosystem [9]. The smartphone App economy showed how much creative and economic potential can be unlocked by giving developers the right tools [10].

In this work we assume a multi-tenant setting: Developers, store operators, and IoT sites are fully distributed and independent. Figure 1 gives an overview: developers on the top tier 1 deliver services to one or multiple stores on tier 2. The stores distribute services to locally-managed IoT sites (tier 3). A fundamental requirement for deploying a service-oriented, site-local management of the IoT is the availability of a suitable service management through all *service life cycle phases*: from development to update.

To the best of our knowledge, this paper identifies for the first time requirements for a globally-distributed but site-locally managed IoT [3], [11]. Major contributions are:

1) Requirements on a globally-distributed locally-managed IoT service deployment.
2) A site-local IoT management reference architecture.

In section II we present the basic setting of our work: our service management reference architecture. Using it, in section III we identify IoT-specific challenges on implementing such an architecture. In section IV we present related research.

II. Our Reference Architecture using one Smart Space Store

Central for an App economy is the availability of an integrated service management over the whole service life-cycle. It distributes services from distributed developers to distributed sites (fig. 1). For being runnable within different IoT sites, services have to be *portable*, running in diverse target settings without change.

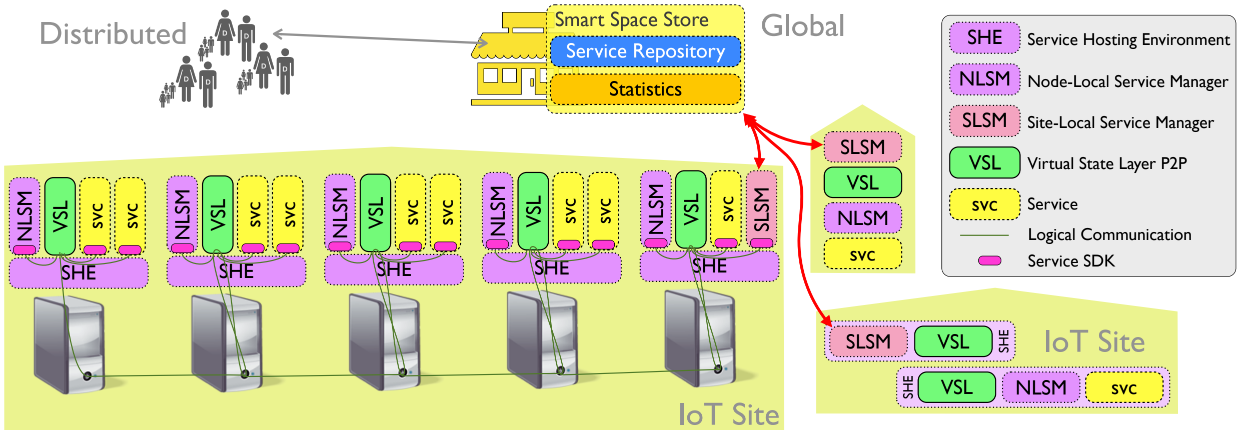

Fig. 2. Hierarchical IoT Service Management Reference Architecture.

With the Virtual State Layer (VSL) [5] we developed a middleware that enables the creation of portable services [2] via dynamic service discovery and composition [12], [13].

On top of middleware, service management functionality is required for obtaining services, installing them on the distributed locally available IoT nodes, and for migrating and updating them to optimize the local setting.

Figure 2 shows our hierarchical service management reference architecture. We assume distributed, independent IoT compute nodes. With the expected continuous increase of computational power, in the near future many IoT devices such as light switches should be able to run services [14], providing a solid base for distributed site local service management.

The service management is hierarchically organized. Node-Local Service Managers (NLSM) autonomously manage all services on a compute node. They report service states and node resources to Site-Local Service Managers (SLSM). Each site's SLSMs optimize the service placement and migration. The SLSMs in each site act as interface towards global stores. A Smart Space Store acts as repository, can collect data from the sites, and provides feedback to developers.

Our assumptions are in-line with other researchers' architectural visions (section IV). Therefore, we use it as reference for identifying important IoT-specific design challenges next.

III. Requirements on IoT Service Management

Our requirements result from a long-term review of research and commercial activities in the IoT, Mobile, Pervasive, and Ubiquitous Computing, and –especially for its systematic research– Network Management. Using our reference architecture (fig. 2), service management can be divided into:

- Development
- Distribution
- Configuration Management
- Dependency Management
- Deployment
- Placement Optimization, e.g. via service migration
- Update Management
- Security Management

Following this structure, we identify requirements for establishing an IoT App economy. See Table I. For each of our identified requirements, a source for further reading is given. For an overview on the IoT application domain from our different viewpoints, we recommend [15]–[18].

Similar to the App economy for smartphones we assume *distributed developers* (see fig. 1). For *service distribution* we assume one or multiple stores. *Configuration* happens partly during development, in the store, at initial deployment within a site, and during operation. *Deployment*, *Placement Optimization*, and *Update* happen within an IoT site.

Services enter the reference architecture at the **Store Back-end**. A major challenge is making it *simple-to-use* (Dev.1): simple, understandable, and transparent [10].

Softwarizing physical environments with IoT installations requires high security standards. Each IoT site is different, making comprehensive service testing before deployment difficult. This circumstance requires continuous *real world testing* (Dev.2): Developers need *access to error statistics* from th distributed IoT sites that are connected to a store [19]. Finally, for fostering crowdsourced development [9], the store has to provide an *accepted business model* (Dev.3), e.g. leaving developers 70% of the revenue [20].

The **Store Front-end** requires *good accessibility* (Sto.1), e.g Browsing through services, and getting recommendations [21]. Software quality can increase through *User Feedback* (Sto.2) [22], [23], e.g. via ratings that help developers and other users.

Site-local service management handles all local management aspects. Dynamic service coupling requires *Software Dependency Management* (Man.1) [24], e.g. towards supporting services [25]. The lack of administrators, and the complexity require *automatic service deployment* (Man.2), e.g. for resolving dependencies. Similarly, *Hardware Dependency Management* (Man.3) is required, e.g. for spawning gateway services on hardware nodes with the required communication interfaces, e.g. Bluetooth [26].

Enabling the composing of services requires *Service Interface Directories* (Man.4) to look up interfaces [12], [24]. Smart spaces differ significantly from smartphones as IoT devices are ambient and as such typically unattended. Having mainly

ID	Required Features	Adv. Pkg. Tool	App/ Play Store	MS HomeStore
Store Back-end				
Dev.1	Simplicity-to-Use [10]	-	✓	-
Dev.2	Support for Real World Testing [19]	-	(✓)	-
Dev.3	Accepted Business Model [**?**], [20]	-	✓	-
Store Front-end				
Sto.1	Good Accessibility, e.g Browsing [21], [22]	(✓)	✓	-
Sto.2	User Feedback [23]	-	✓	-
Site-Local Service Management				
Man.1	SW Dependency Management [24]	✓	-	✓
Man.2	Automatic Service Deployment [10]	✓	(✓)	-
Man.3	HW Dependency Management [26]	(✓)	(✓)	-
Man.4	Service Interface Directories [12], [24], [32]	-	-	-
Man.5	Unattended Provisioning/ Update [27]	-	(✓)	-
Man.6	Management at Run Time [2]	-	(✓)	-
Security				
Sec.1	Verification of Service Package [4]	✓	✓	-
Sec.2	Access Rights Management [25]	(✓)	✓	✓
Sec.3	Functional SW Validation [29]	(✓)	(✓)	-
Monitoring				
Mon.1	Usage Statistics [19], [24], [30]	-	(✓)	-
Mon.2	Continuous Monitoring [19], [24], [30]	-	(✓)	✓
Mon.3	Autom. Feedback Collection [19], [30], [31]	-	(✓)	✓
Convergence				
Con.1	Convergence Support [24], [28]	-	-	-

TABLE I
PROPOSED FEATURES COMPARED TO THE CLOSEST STATE OF THE ART.

unattended computing nodes requires *Unattended Provisioning/ Update* (Man.5) [27]. The IoT's "always-on" requires *Management at Run Time* (Man.6).

The IoT processes personalized data, e.g. by monitoring a person's presence. With its inherent threats to user privacy [28], *security is mission-critical in the IoT*. Mechanisms for protecting the integrity-of and access-to services is essential. This includes the *Verification of Service Packages* (Sec.1) [4], suitable *Access Rights Management* (Sec.2) [25], and ideally *Functional SW Validation* (Sec.3) [29].

Service *Usage Statistics* (Mon.1) from the store, *Continuous Monitoring* (Mon.2) on site, and *Automated Feedback Collection* (Mon.3) of non-privacy critical metadata can help identifying and mitigating problems [19], [30], [31].

Finally, crowdsourced development requires a scaling equivalent for the standardization processes we have in the networking community. Service interface *Convergence Support* (Con.1) is required for service portability and overall usability of the service-oriented approach [24].

IV. STATE OF THE ART

Plenty of research regarding service management in other domains such as general IT management, Mobile Computing, Pervasive and Cloud Computing exists. However, as seen in section III, distribution and heterogeneity of the IoT impose additional requirements on suitable, full life-cycle service management. Table I shows that most of our identified requirements go beyond the closest state of the art: Advanced Package Tool [33], App/ Play store [10], [34], and MS HomeStore [35].

The software distribution infrastructure of operating systems is close to our work as it provides node-local service management. *Packet managers* emerged to manage the growing complexity of the software installed on computers [33]. They are less focused on ad-hoc, online service management than on keeping local installations up-to-date. Software packets contain executables and metadata, including dependencies (Man.1, Man.3) [36]. Packet managers typically offer an interface for browsing and searching applications (Sto.1), manage and resolve dependencies on other software packages (Man.1) [33], and verify downloaded packets before installing them (Sec.1). Repository maintainers test and sign the packets in the repository (Sec.1). Access rights are typically configured on a machine's operating system, e.g. by an install script (Sec.2). Some functional software validation happens by the community (Sec.3).

The Mobile Computing App economies, App Store (2008) and the Play Store (2009) [10], [34], manage more homogeneous devices. The missing distribution of compute nodes within smartphones makes them less complex. Besides core packet management functionality, App stores have significantly increased usability compared to packet managers (Dev.1). Additionally, Apple created a value chain that fosters innovation and crowdsourced development (Dev.2, Dev.3, Sto.1, Sto.2) [37]. The App Store supports the entire App life cycle including payment, shipping, and updates [38], [39]. Operators typically keep 30% of the revenue of an App (Dev.3) [34], [40]. With user consent, crash reports from the participating devices can be collected and evaluated (Dev.2, Mon.2, Mon.3). The access statistics are partly publicly available (Mon.1). Apple manually checks Apps for various criteria before admitting them (Sec.3) [41]. Google runs automated tests, and reacts to user security reports (Sec.3) [42]. App Stores form a holistic App management ecosystem (Man.2, Man.5, Man.6, Sec.1, Sec.2) [43].

The scenario-wise closest related work is from the Pervasive Computing community. Microsoft research introduced the centralized HomeOS concept with its HomeStore in [35]. Apps in the HomeStore have manifests that express dependencies to other Apps (Man.1). When installing an App the user is asked to set rules that define what the App may access (Sec.2). In [19], using real world installations as a testbed (HomeLab) is proposed with a focus on doing real-world user studies (Mon.2, Mon.3).

Little work towards establishing a non-cloud-based IoT App economy exists. IoT site-locally, existing solutions either focus on service placement within IoT node sets [6], [7], or on deploying software on distributed resource-constraint nodes [8]. In [44] an overview on different IoT management architectures is given. The authors of [11] identify the need for service management research in the IoT.

Regarding edge-computing, there are approaches towards content management, load distribution, and offloading between resource weak devices and the cloud [3], [11].

[45] use docker containers with Kubernetes and OpenStack. Their focus is on deploying services between the cloud, the fog, and site-locally. Their solution solves the unattended service management (Man.5, Man.6). In [46] the solution gets generalized for a site-local management.

V. Conclusion

Mobile Computing showed how an App economy can significantly push the establishment of a computing paradigm [10], [34], [37]. Based on the life-cycle of an IoT service, we identified IoT-specific challenges that emerge when designing a global App distribution/ local management architecture.

Having an App Store strongly influenced the smartphone buying decision [34]. A similar impact can be expected for a Smart Spaces Store in the future, making the presented architecture not only technically but also economically relevant.

With this work we want to direct IoT service management research to points that may not be directly at hand, but are highly relevant. We are working on the presented concepts and look forward to other researchers joining us.

References

[1] D. Lu, D. Huang, A. Walenstein, and D. Medhi, "A Secure Microservice Framework for IoT," in *IEEE Symposium on Service-Oriented System Engineering (SOSE)*. IEEE, 2017.

[2] M.-O. Pahl, G. Carle, and G. Klinker, "Distributed Smart Space Orchestration," in *Network Operations and Management Symposium 2016 (NOMS 2016) - Dissertation Digest*, 2016.

[3] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Communications Surveys and Tutorials*, 2018.

[4] M.-O. Pahl and L. Donini, "Securing IoT Microservices with Certificates," in *Netw. Operations and Managem. Symp. (NOMS)*, 2018.

[5] M.-O. Pahl and S. Liebald, "Designing a Data-Centric internet of things: Vsl," in *Int. Conference on Networked Systems (NetSys)*, 2019.

[6] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards QoS-Aware Fog Service Placement," *Proceedings - 1st Int. Conference on Fog and Edge Computing, ICFEC*, 2017.

[7] S. Ali, M. G. Kibria, and I. Chong, "WoO enabled IoT service provisioning based on learning user preferences and situation," *Int. Conference on Information Networking*, 2017.

[8] L. Belli, S. Cirani, L. Davoli, A. Gorrieri, M. Mancin, M. Picone, and G. Ferrari, "Design and Deployment of an IoT Application-Oriented Testbed," *Computer*, 2015.

[9] M.-O. Pahl and G. Carle, "Taking Smart Space Users Into the Development Loop," in *HomeSys 2013 (UbiComp Adjunct)*, 2013.

[10] V. Gonçalves, N. Walravens, and P. Ballon, "How about an App Store? Enablers and Constraints in Platform Strategies for Mobile Network Operators," in *9th Conference on Mobile Business and Global Mobility Roundtable (ICMB-GMR)*. IEEE, 2010.

[11] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions," 2016.

[12] M.-O. Pahl and S. Liebald, "A modular distributed iot service discovery," in *Int. Symposium on Integrated Network Management (IM)*, 2019.

[13] M.-O. Pahl, "Data-Centric Service-Oriented Management of Things," in *Integrated Network Management (IM)*, 2015.

[14] R. Want, "When Cell Phones Become Computers," *Pervasive Computing, IEEE*, 2009.

[15] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things - A Survey." *IEEE Internet of Things Journal*, 2016.

[16] M. Knappmeyer, S. L. Kiani, E. S. Reetz, N. Baker, and R. Tonjes, "Survey of Context Provisioning Middleware," *Communications Surveys & Tutorials, IEEE*, 2013.

[17] V. Raychoudhury, J. Cao, M. Kumar, and D. Zhang, "Middleware for pervasive computing: A survey," *Perv. and Mobile Computing*, 2012.

[18] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Personal Communications*, 2001.

[19] A. J. B. Brush, J. Jung, R. Mahajan, and J. Scott, "HomeLab: shared infrastructure for home technology field studies," in *UbiComp '12*, 2012.

[20] B. Bergvall-Kåreborn and D. Howcroft, "The Apple business model: Crowdsourcing mobile applications," *Accounting Forum*, 2013.

[21] F. Cuadrado and J. C. Duenas, "Mobile Application Stores: Success Factors, Existing Approaches, and Future Developments," *IEEE Communications Magazine*, 2012.

[22] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," *21st IEEE Int. Requirements Engineering Conference (RE)*, 2013.

[23] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, "Why people hate your app," *19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, 2013.

[24] M.-O. Pahl and G. Carle, "Crowdsourced Context-Modeling as Key to Future Smart Spaces," in *Network Operations and Management Symposium (NOMS)*, 2014.

[25] M.-O. Pahl and L. Donini, "Giving iot edge services an identity and changeable attributes," in *Int. Symposium on Integrated Network Management (IM)*, 2019.

[26] M.-O. Pahl and G. Carle, "The Missing Layer - Virtualizing Smart Spaces," in *MUCS 2013*, 2013.

[27] Apple Incorporated, "App Distribution Guide," Apple Inc., 2013.

[28] M.-O. Pahl and F.-X. Aubet, "All eyes on you: Distributed Multi-Dimensional IoT microservice anomaly detection," in *14th Int. Conference on Network and Service Management (CNSM)*, 2018.

[29] M. Mohsin, Z. Anwar, G. Husari, E. Al-Shaer, and M. A. Rahman, "IoTSAT: A formal framework for security analysis of the internet of things (IoT)," *IEEE Conference on Communications and Network Security, CNS*, 2017.

[30] D. Bloom, "Selection Criterion and Implementation of a Trouble Tracking System: What's in a Paradigm?"

[31] H. M. Tran, C. Lange, G. Chulkov, J. Schönwälder, and M. Kohlhase, "Applying semantic techniques to search and analyze bug tracking data," *Journal of Network and Systems Management*, 2009.

[32] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, "Web Services: Concepts, Architectures and Applications, 1st edition," 2004.

[33] F. Mancinelli, J. Boender, R. di Cosmo, J. Vouillon, B. Durak, X. Leroy, and R. Treinen, "Managing the Complexity of Large Free and Open Source Package-Based Software Distributions," in *Automated Software Engineering, 2006. ASE '06*, 2006.

[34] K. Kimbler, "App store strategies for service providers," in *Intelligence in Next Generation Networks (ICIN) 2010*, 2010.

[35] C. Dixon, R. Mahajan, S. Agarwal, A. J. Brush, B. Lee, S. Saroiu, and P. Bahl, "An operating system for the home," in *NSDI'12*, 2012.

[36] C. C. Lee. (2002) Debian Binary Package Building HOWTO.

[37] H. Kim and I. Kim, "The Success Factors for App Store-Like Platform Businesses from the Perspective of Third-Party Developers," in *Pacific Asia Conference on Information Systems 2010*, Taipei, Taiwan, 2010.

[38] M. N. Cortimiglia, A. Ghezzi, and F. Renga, "Mobile Applications and Their Delivery Platforms," *IT Professional*, 2011.

[39] V. K. Tuunainen, T. Tuunanen, and J. Piispanen, "Mobile Service Platforms: Comparing Nokia OVI and Apple App Store with the IISIn Model," in *Mobile Business (ICMB), 2011*, 2011.

[40] Y. S. Yoon, J. Yoo, and M. Choi, "Revenue sharing is the optimal contractual form for emerging app economy?" in *Information and Communication Technology Convergence (ICTC), 2010*, 2010.

[41] Apple Incorporated, "App Store Review Guidelines," Apple Inc., 2010.

[42] Android Open Source Project. Android Security FAQ — Android Developers. [Online]. Available: http://developer.android.com/guide/faq/security.html

[43] M. H. Goadrich and M. P. Rogers, "Smart Smartphone Development: iOS versus Android ," in *the 42nd ACM technical symposium*, 2011.

[44] C. K. Dehury and P. K. Sahoo, "Design and implementation of a novel service management framework for IoT devices in cloud," *Journal of Systems and Software*, 2016.

[45] D. Santoro, D. Zozin, D. Pizzolli, F. D. Pellegrini, and S. Cretti, "Foggy: A Platform for Workload Orchestration in a Fog Computing Environment," *Proceedings of the Int. Conference on Cloud Computing Technology and Science, CloudCom*, 2017.

[46] Y. C. Hsieh, H. J. Hong, P. H. Tsai, Y. R. Wang, Q. Zhu, M. Y. S. Uddin, N. Venkatasubramanian, and C. H. Hsu, "Managed edge computing on Internet-of-Things devices for smart city applications," 2018.