

COLT Collaborative Delivery of Lightweight IoT Applications

Michael Vögler^(✉), Fei Li, Markus Claeßens, Johannes M. Schleicher,
Sanjin Sehic, Stefan Nastic, and Schahram Dustdar

Distributed Systems Group, Vienna University of Technology,
Argentinerstraße 8/184-1, 1040 Vienna, Austria
{voegler,li,claessens,schleicher,sehic,nastic,dustdar}@dsg.tuwien.ac.at
<http://dsg.tuwien.ac.at/>

Abstract. Today's IoT solutions are typically delivered by domain-specific solution providers. In these solutions, components are highly customized for specific project requirements, limiting their users to the offerings of their providers. To overcome these limitations, we propose a novel mechanism that opens the market for OEMs and third-party developers. This paper introduces the IoT Application Market, where a range of stakeholders can distribute, sell, share and contribute features for lightweight device-specific IoT applications. Based on the offerings in the IoT Application Market, users can buy applications and deploy them in their environment through self-service.

Keywords: Internet of things · IoT applications · IoT market · IoT PaaS

1 Introduction

The overall process of delivering an IoT solution is typically orchestrated by solution providers for specific projects [1, 8]. Since this delivery process is mostly vendor-specific, a big challenge in the Internet of Things domain, is to establish an environment that allows various stakeholders to share, contribute and distribute applications collaboratively. The scale of IoT solutions can range from a simple application that monitors and reacts to a temperature sensor, to a full-blown enterprise application that manages and controls a set of buildings. In this paper we focus on lightweight applications that can be executed on gateways [9–11], which are used to connect heterogeneous devices and handle resources efficiently. These lightweight applications are software services that carry out various tasks autonomously by executing application code directly on the embedded runtime environment of the device, resulting in a small footprint in terms of resource and memory consumption. The drawback of this approach is that the actual development of these gateway-dependent applications requires deep knowledge of the runtime environment, which significantly increases the efforts for development, and limits the involvement of more developers. As a result providers are

interested in distributing and selling their applications to a broad audience to cover their efforts. To save time and costs as well as expand their market coverage, providers are eager to collaborate with other stakeholders to reuse their solutions, which will only work in an ecosystem that allows not only distribution but also a model to collaborate.

In this paper we present a novel ecosystem, where application providers can share, contribute to and distribute IoT applications. The core idea is to realize an IoT Application Market, that opens the application delivery process to OEMs and third-party developers. To facilitate and integrate the offerings of the mentioned stakeholders, the market supports application providers to form collaborations with other stakeholders to create and distribute applications. Furthermore the market allows users to search for suitable applications, purchase and deploy them in their local environment through self-service.

The paper is structured as follows: Sect. 2 presents a motivating industrial use case, Sect. 3 illustrates the overall approach, Sect. 4 describes the implementation details, in Sect. 5 the related work is compared, and finally the paper concludes in Sect. 6.

2 Motivation

This work is motivated by a real-world scenario in Building Management Systems (BMS). The scenario is based on a case study we conducted in our lab in collaboration with an industrial partner. In the case study we have identified the following: generally a BMS provides efficient management of building facilities to save energy and reduce operational costs. A BMS consists of a vast number of sensors, other smart devices and a platform that connects and handles these resources and is developed by an *Application Provider* for a specific *Customer*. During the process of delivering a BMS, the provider gathers information about the target building, installs and integrates suitable hardware into an infrastructure and finally connects it to a platform that provides specific control and management applications. This approach has the following limitations: (i) the customer is limited to the applications and services offered by his provider, and (ii) providers have to develop tailored applications based on customer demands and their respective installed solution, which is both time- and cost-ineffective. To address these limitations we want to open the process of developing customer specific applications to the following contributing stakeholders. *Third-party developers* develop and offer lightweight applications that run on specific IoT devices. Based on the installed device-set, a customer can choose suitable applications. This gives the customer more flexibility and the provider the opportunity to outsource some of the application development. *OEMs* provide complete applications or just contribute features that are tailored for specific hardware devices. These offerings can be reused by both third-party developers and providers for new requirements or different physical environments. Based on this scenario we identified the following requirements: (i) a market-based collaboration and distribution environment, and (ii) a solution to manage the deployment and execution of lightweight IoT Applications.

3 IoT Application Market

In order to address the previously mentioned requirements we propose an approach that consists of the following key components: *IoT Application*, *IoT Application Market* and *IoT PaaS*.

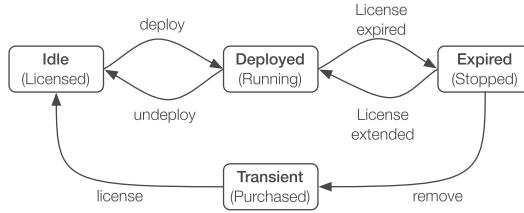


Fig. 1. IoT Application lifecycle

In this paper we focus on lightweight IoT Applications, which are used to automatically tune building facilities, acquire data from different sensors, monitor devices, and automatically report a certain device status. In our approach, an IoT Application is represented as a self-contained archive with corresponding metadata, containing the following information: (i) a *Name* that uniquely identifies the application, (ii) a natural language *Description* that can be read by humans and a structural one that can be understood by software (i.e. the market) in order to provide a reliable search, (iii) *Provider* name and id, who created and uploaded the application, (iv) a list of *Suitable Devices* the application can be deployed and executed on. This list can always be updated by the provider, to react on new emerging compatible devices, and (v) a *Version* number that uniquely identifies the current release of the application.

The lifecycle of an application is depicted in Fig. 1 and has the following states: *Transient*, *Idle*, *Deployed* and *Expired*. An application that is purchased is in state *Transient*. Once the application is licensed by the market it is in state *Idle*. As soon as the user deploys the licensed application on the gateway device, the gateway checks if the license is valid, initializes the startup process and sets the state to *Deployed*. If the gateway detects that the license is expired, it interrupts the execution and sets the application to *Expired*.

3.1 IoT Application Market

The IoT Application Market represents an ecosystem, where *Application Providers* offer and users browse and retrieve applications. The overall process, involved components and stakeholders are depicted in Fig. 2. The market stores the uploaded applications in the *IoT Application Repository*, which is an open service repository that hosts applications as self-contained archives with corresponding metadata. The market provides the following core components.

The *Stakeholder Management* is responsible for handling the management of users, their authentication and authorization within the market. We identified

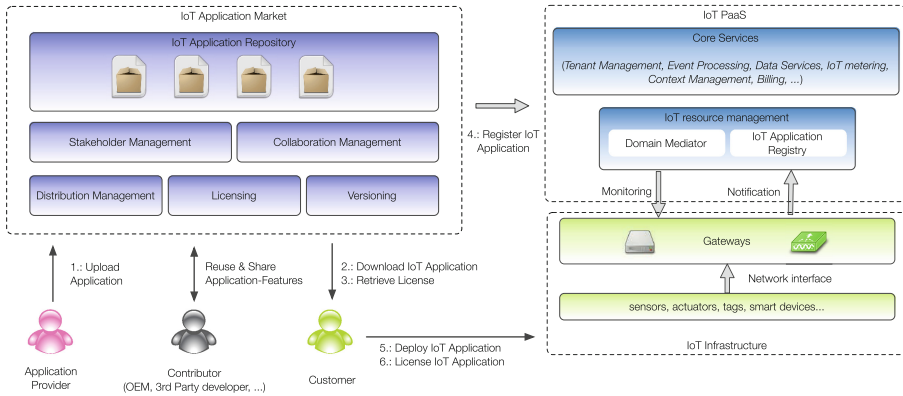


Fig. 2. IoT Application Market and IoT PaaS

the following groups of users based on [3]: *Application providers* develop applications and upload them to the market. Additionally the provider can decide to share features with other users of the market and form a collaboration. Users invited to collaborations are called contributors. *Contributors* can reuse shared features for other or new applications and provide new or improved code for the collaboration. Based on this collaboration, involved parties can decide if and how they want to distribute the outcome and decide on an appropriate revenue sharing model. Users of the market, which are not involved in the development process are called customers. *Customers* browse the market and search for interesting and suitable applications e.g. if the release of an application can be deployed and executed on the owned IoT devices. The *Collaboration Management* supports and manages the collaboration among the application provider and contributors. For each group of users that formed a collaboration, the market provides a specific private repository, where the group can store their application code and other required artifacts. This repository can only be accessed by the members of the collaboration and provides the basic features of a conventional source code management (SCM) and revision control system. Additionally it supports different pluggable metrics to track the contribution of each member of the collaboration, e.g. one of the easiest models, calculates the contribution based on the number of committed code-lines. The Collaboration Management uses the data provided by the respective metric to calculate the revenue share. The *Distribution Management* monitors and controls the distribution of applications. A monitoring component keeps track of all downloads and a control component regulates if a user is allowed to download a specific application, based on historic data e.g. punctual payment of bills. The *Licensing* component governs the use or redistribution of an application. A typical license in our model grants the user permission to use an application on a defined set of field devices.

After downloading the application from the market and deploying it on the gateway, the user needs the corresponding license file that triggers the actual instantiation and startup process of the application.

Based on the user's inputs (e.g. used device group, subscription schema) the market generates a license file that can now be deployed on the gateway. The gateway detects the deployed license and triggers the startup process of the corresponding application. During the execution of the application the gateway may detect that the license is expired and passivates the execution. The user can now decide if he wants to extend the license or undeploy the application from the device. In case the user wants to prolong the license he can retrieve an extended license in the market. To keep track of multiple releases of the same application the market uses a *Versioning* component. Since in the market we need to handle a vast amount of applications, we need an efficient mechanism to deal with different releases of the same application. Therefore, whenever a provider uploads a new release to the market, he has to decide if it is either a major (significant improvement) or minor (bug fixes) release. This approach has the following advantages: (i) developers can decide for which release they want to provide updates and which are no longer supported, (ii) users can easily decide which release is the most suitable for their environment, and (iii) the market can notify the user about updates or known issues for a certain release.

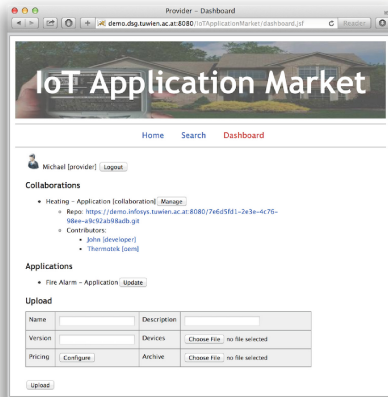
3.2 IoT PaaS

IoT PaaS [7] is a domain-independent Platform as a Service framework. In general, IoT solutions are highly domain-specific, so the IoT PaaS framework is build to be generic and extendable enough to be used in different IoT domains. The framework provides essential platform services on cloud that can be used and extended by IoT solution providers. Additionally IoT PaaS offers a registration point for devices, gateways and control applications, to handle the vast number of IoT resources. This registration point is the *IoT resource management* and uses *Domain Mediators* to mediate between different gateway interfaces. To handle events and notifications that can be pushed by IoT applications, we extended the resource management of IoT PaaS and added the *IoT Application Registry*. The registry is a constantly evolving catalog of information about the available and deployed applications, which guarantees that the platform always knows what kind of application and how many of them are deployed. The necessary information about the current status of an application is published by two components: the IoT Application Market and connected gateways, respectively the deployed application.

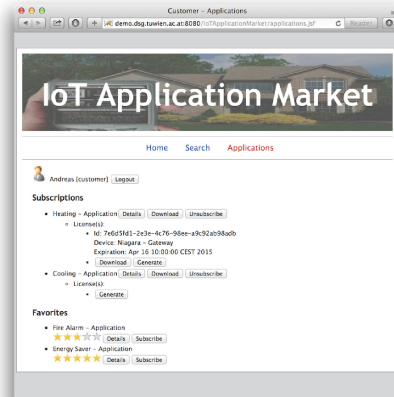
4 Implementation

The prototype of the presented approach is implemented in Java. We utilize several open-source products from WSO2¹. We used the Application Server for hosting the service-based components. To provide a flexible and scalable solution the prototype uses the ESB to connect and integrate the components. For the

¹ <http://wso2.com>.



(a) Provider Perspective



(b) Customer Perspective

Fig. 3. IoT Application Market

IoT Application Repository we implemented a custom repository, based on the Governance Registry and Repository. The *Stakeholder Management* is based on an Identity Server that handles users and their respective roles and additionally provides sophisticated security management of services and APIs. For the *Collaboration management* we utilized Gitorious² and integrated the statistical analysis tool gitinspector³. The communication of IoT Applications with IoT PaaS is based on push notifications implemented by JMS Queues, Publishers and Subscribers supported by Apache ActiveMQ⁴. As our Gateway environment we used the Niagara Framework⁵. The implementation of the IoT Application Lifecycle facilitates Niagara's Component Lifecycle⁶ mechanism. Furthermore we facilitated the Niagara Licensing⁷ methodology for custom applications, for our *Licensing* implementation.

To demonstrate our prototype we will discuss two stakeholder perspectives of the market. The Application Providers' perspective is depicted in Fig. 3a and displays the provider's collaborations and for each collaboration the repository provided by the market and the contributors with their respective role are shown. Additionally the screen displays applications that got uploaded by the provider and a form to add new offerings to the market. Figure 3b depicts the customers' perspective. The screen displays subscriptions, which the customer received for purchasing an application. For each subscription the customer can generate a

² <https://gitorious.org>.

³ <https://code.google.com/p/gitinspector/>.

⁴ <http://activemq.apache.org>.

⁵ http://www.niagaraax.com/cs/products/niagara_framework.

⁶ <https://community.niagara-central.com/ord?portal:/blog/BlogEntry/254>.

⁷ <https://community.niagara-central.com/ord?portal:/blog/BlogEntry/269>.

new or download a license file that is required for deploying the application. Applications that are currently not purchased can be searched and added as favorites.

5 Related Work

The key concept, the overall terminology and used components of the Internet of Things is well-defined in the literature. In comparison the definition of applications in IoT is not that clear. Recent approaches [4, 6] define IoT applications, as applications that are developed on top of a resource abstraction layer. Zhu et al. [11] define two different types of IoT applications. An application that is embedded in and an enterprise application that is build on the Gateway. In contrast to the aforementioned approaches, this paper proposes lightweight IoT applications that are directly deployed and executed on IoT devices (gateways), which brings advantages in terms of scaling and resource consumption.

Bohli et al. [2] describes a market that offers a platform to collect and trade all kinds of sensor information in a WSN. They identified several entities that play an important role in the market and discuss possible pricing schemes and strategies. As the current development of IoT solutions is mostly driven by big industrial players, [5] proposes a set of connected marketplaces where users and developers can create and share innovative IoT artifacts. Although, they share similarities with our approach regarding opening the IoT service delivery process the authors solely concentrate on including users to enable innovation. Contrary to them, our market provides an ecosystem that empowers providers to collaborate with various stakeholders to reduce the costs of the delivery process. Furthermore [5] is mainly focused on providing concepts for the overall approach, whereas we implemented a prototype to demonstrate our market's capabilities.

6 Conclusion and Future Work

This paper proposes the IoT Application Market – a novel open market model for managing the distribution of IoT applications. In the IoT Application Market stakeholders can share, contribute and distribute their lightweight and device-specific IoT Applications. Based on the offerings users can browse the market for interesting and suitable applications. Then they can buy, deploy and execute them in their own local environment through self-service. To efficiently handle the deployment and execution of the vast amount of applications this paper utilizes the IoT PaaS framework. This domain-independent framework provides a set of essential platform services and eases the overall management of IoT Applications. As future work we plan to extend our approach beyond the domain of building management systems. We will also try to identify further participants in the overall market and distribution model. Furthermore, we intend to integrate pricing and revenue sharing models that allow more stakeholders that are involved in the development process, to collaborate.

Acknowledgment. This work is sponsored by Pacific Controls Cloud Computing Lab (PC³L), a joint lab between Pacific Controls L.L.C., Scheikh Zayed Road, Dubai, United Arab Emirates and the Distributed Systems Group of the Vienna University of Technology. (<http://pc3l.infosys.tuwien.ac.at/>)

References

1. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. *Comput. Netw.* **54**(15), 2787–2805 (2010)
2. Bohli, J.M., Sorge, C., Westhoff, D.: Initial observations on economics, pricing, and penetration of the Internet of Things market. *SIGCOMM Comput. Commun. Rev.* **39**(2), 50–55 (2009)
3. Dustdar, S., Li, F., Truong, H.L., Sehic, S., Nastic, S., Qanbari, S., Vögler, M., Claeßens, M.: Green software services: from requirements to business models. In: 2013 2nd International Workshop on Green and Sustainable Software (GREENS), pp. 1–7. IEEE (2013)
4. Guinard, D., Ion, I., Mayer, S.: In search of an Internet of Things service architecture: REST or WS-*? A developers' perspective. In: Puiatti, A., Gu, T. (eds.) *MobiQuitous 2011*. LNICTS, vol. 104, pp. 326–337. Springer, Heidelberg (2012)
5. Kortuem, G., Kawsar, F.: Market-based user innovation in the Internet of Things. In: 2010 Internet of Things (IoT), pp. 1–8 (2010)
6. Kovatsch, M.: Firm firmware and apps for the Internet of Things. In: Proceedings of the 2nd Workshop on Software Engineering for Sensor Network Applications, SESENA 2011, pp. 61–62. ACM, New York (2011)
7. Li, F., Vögler, M., Claeßens, M., Dustdar, S.: Efficient and scalable IoT service delivery on cloud. In: 2013 IEEE 6th International Conference on Cloud Computing (CLOUD) (2013)
8. Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I.: Internet of Things: vision, applications and research challenges. *Ad Hoc Netw.* **10**(7), 1497–1516 (2012)
9. ThereCorporation: ThereGate. <http://therecorporation.com/en/platform>
10. Tridium: JACE Controller. http://www.tridium.com/cs/products/_services/jace
11. Zhu, Q., Wang, R., Chen, Q., Liu, Y., Qin, W.: IoT gateway: bridging wireless sensor networks into Internet of Things. In: 2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC), pp. 347–352 (2010)