# GovOps: The Missing Link for Governance in Software-Defined IoT Cloud Systems

Stefan Nastic(✉), Christian Inzinger, Hong-Linh Truong, and Schahram Dustdar

Distributed Systems Group, Vienna University of Technology, Vienna, Austria
{nastic,inzinger,truong,dustdar}@dsg.tuwien.ac.at

**Abstract.** Cloud computing and the IoT are converging ever stronger, enabling the proliferation of diverse large-scale IoT cloud systems. Such novel IoT cloud systems offer numerous advantages for the variety of involved stakeholders. However, due to scale, complexity, and inherent geographical distribution of IoT cloud systems, governing new IoT cloud resources and capabilities poses numerous challenges. In this paper, we introduce GovOps – a novel approach and a conceptual model for cloud-based, dynamic governance of software-defined IoT cloud systems. By introducing a suitable *GovOps reference model* and a dedicated *GovOps manager*, it simplifies realizing governance processes and enables performing custom governance tasks more efficiently in practice. We introduce real-world case studies in the building automation and vehicle management domains, to illustrate the main aspects and principles of our approach to governance of large-scale software-defined IoT cloud systems.

## 1 Introduction

To date, cloud computing models and techniques, such as infrastructure virtualization and management, Compute-, Storage- and Network-as-a-Service, etc., have been intensively exploited for large-scale Internet of Things (IoT) systems [7,14,18]. Recently, software-defined IoT cloud systems have been introduced [10] in order to enable easier provisioning and management of IoT cloud resources and capabilities. Generally, software-defined denotes a principle of abstracting low-level components (e.g., hardware) and enabling their management, programmatically through well-defined APIs [8]. This enables refactoring the underlying infrastructure into finer-grained resource components whose functionality can be (re)defined after they have been deployed. While IoT cloud systems introduce numerous possibilities, a plethora of challenges to govern and operate these new IoT cloud resources and capabilities emerge.

Various domains, such as smart building and vehicle management, increasingly rely on IoT cloud resources and capabilities. Consequently, governance issues such as security, safety, legal boundaries, compliance, and data privacy concerns are ever stronger being addressed [4,5,17], mainly due to their potential impact on the variety of involved stakeholders. However, existing approaches are mostly


F. Toumani et al. (Eds.): ICSOC 2014, LNCS 8954, pp. 20–31, 2015.
DOI: 10.1007/978-3-319-22885-3_3

intended for high-level business stakeholders, neglecting support, e.g., tools and frameworks, to realize governance strategies in large-scale, geographically distributed IoT cloud systems. Approaching IoT cloud from the operations management perspective, different approaches have been presented, e.g. [2,14,15,18]. Such approaches deal with IoT cloud infrastructure virtualization and its management, enabling utilization of cloud computation resources and operating cloud storage resources for big IoT data. However, most of these approaches do not consider high-level governance objectives such as legal issues and compliance. This increases the risk of lost requirements or causes over-regulated systems, potentially increasing costs and limiting business opportunities.

Currently, IoT governance mostly addresses the *Internet* part of the IoT, e.g., in the context of the Future Internet services[1], while IoT operations processes mostly deal with *Things* (e.g., in [3]) as additional resources that need to be operated. Therefore, governance objectives (law, compliance, etc.) are not easily mapped to operations processes (e.g., querying sensory data streams or adding/removing devices). Contemporary models, which assume that business stakeholders define governance objectives, and operations managers implement and enforce them, are hardly feasible in IoT cloud systems. In practice, bridging the gap between governance and operations management of IoT cloud systems poses significant challenges, because traditional management and governance approaches are hardly applicable for IoT cloud systems, mainly due to the large number of involved stakeholders, novel requirements for shared resources and capabilities, dynamicity, geographical distribution, and the sheer scale of IoT cloud systems.

This calls for a systematic approach to govern and operate IoT cloud resources and capabilities. Extending the previously developed concepts [10], in this paper we introduce GovOps – a novel approach for cloud-based dynamic governance and operations management in software-defined IoT cloud systems. The main objectives of GovOps are twofold. On the one side, it aims to enable seamless integration of high-level governance objectives with concrete operations processes. On the other side, it enables performing operational governance processes for IoT cloud systems in such manner that they are feasible in practice. We present a GovOps reference model that defines required roles, concepts, and techniques to reduce the complexity of realizing IoT cloud governance processes. GovOps enables performing custom governance tasks more efficiently, thus reduces time, costs, and potential consequences of insufficient or ineffective governance.

The remainder of this paper is structured as follows: Sect. 2 presents motivating scenarios that will be used throughout the paper. In Sect. 3, we present the GovOps approach to governance and operations management in software-defined IoT cloud systems; Sect. 4 outlines the GovOps reference model; Sect. 5 discusses the related work; Finally, Sect. 6 concludes the paper and gives an outlook of our future work.

[1] http://ec.europa.eu/digital-agenda/en/internet-things.

## 2 Scenarios: Governing Software-Defined IoT Systems

Consider the following scenarios in the Building Automation and Vehicle Management domains that we will refer to throughout the rest of this paper. The scenarios are derived from our work conducted in the P3CL lab[2].

### 2.1 Scenario 1 – Fleet Management System

**General Description.** Fleet Management System (FMS) is responsible for managing electric vehicles deployed worldwide, e.g., on different golf courses. We have identified three stakeholders who rely on the FMS to optimize their business tasks: vehicle manufacturer, distributors and golf course managers. The stakeholders have different business models. For example, as the manufacturer only leases the vehicles, he is interested in the complete fleet, e.g., regular maintenance, crash reports and battery health. On the other side, golf course managers are mostly interested in vehicles security (e.g., geofencing features), preventing misuse, and safety on the golf course.

**Infrastructure Setup.** The FMS is an IoT cloud system comprising vehicles' proprietary on-board gateways, network and cloud infrastructure. The on-board gateway is capable to host lightweight applications for: vehicle maintenance, tracking, monitoring and club set-up. Vehicles communicate with the cloud via 3G, GPRS or Wi-Fi network to exchange telematic and diagnostic data. On the cloud we host different FMS subsystems and services to manage and analyze this data, e.g., determine vehicle status, perform remote diagnostics, batch configuration and software updates. Legacy vehicles that are not capable to host applications are integrated using a CAN-IP bridge, and any custom business logic needs to be executed in the cloud.

### 2.2 Scenario 2 – Building Automation System

**General Description.** Building Automation System (BAS) is responsible to monitor and control various building assets, such as HVAC, lighting, elevators and humidity control systems, as well as to handle fault events and alarms (e.g., fire or gas leakage). For safety-critical services (e.g., alarm handling), timely processing of the events and the availability of the BAS play a crucial role and need to be ascertained.

**Infrastructure Setup.** Generally, BAS comprises a set of cloud-based services, gateways and various sensors and actuators integrated with the building's assets. Gateways which support typical BAS device protocols (ModBus, BACnet, LonWorks and Fox), e.g., Niagra or Sedona[3], are used to communicate with sensors

[2] http://pcccl.infosys.tuwien.ac.at/.
[3] http://www.tridium.com/.

and actuators. For local processing, the gateways usually allow executing custom triggers, rules and some form of complex event processing (CEP) queries. For permanent storage and more resource-demanding processing, the gateways send streams of data to the remote cloud services.

### 2.3 System Characteristics

We notice that both the FMS and the BAS have large-scale, geographically distributed infrastructure. Additionally, the FMS utilizes virtualized IoT cloud infrastructure, such as virtual gateways (VGW), to support integrating legacy vehicles. Depending on stakeholder and task-at-hand our systems have different customization requirements and non-functional requirements (e.g., regarding fault-tolerance and availability). For example in BAS, while for safety-critical services, real-time delivery and processing is essential, for services such as HVAC controller, cost reduction is more important. Due to the multiplicity of the involved stakeholders, the FMS needs to allow for flexible runtime customizations in order to exactly meet the stakeholder's functional requirements, depending on the problem-at-hand and availability or accessibility of the vehicles, as well as desired system's non-functional properties.

## 3 GovOps – A Novel Approach to Governance and Operations Management in IoT Cloud

The main objective of our GovOps approach (Governance and Operations) is twofold. On the one side it aims to enable seamless integration of high-level governance objectives and strategies with concrete operations processes. On the other side, it enables performing operational governance processes for IoT cloud systems in such manner that they are feasible in practice.

Figure 1 illustrates how GovOps relates to IoT cloud governance and operations. It depicts the main idea of GovOps to bring governance and operations closer together and bridge the gap between governance objectives and operations processes, by incorporating the main aspects of both IoT cloud governance and operations management. To this end, we define *GovOps principles and design process* of GovOps strategies (Sect. 4) that support determining what can and needs to be governed, based on the current functionality and features of an IoT cloud system, and that allow for aligning system's capabilities with regulations and standards. Additionally, we introduce a novel role, *GovOps manager* (Sect. 3.3) responsible to guide and manage designing GovOps strategies,

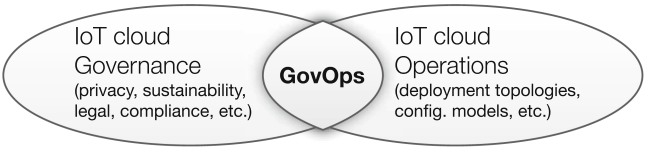


**Fig. 1.** GovOps in relation to IoT cloud governance and operations.

because in practice it is very difficult, risky, and ultimately very costly to adhere to traditional organizational silos, separating business stakeholders from operations managers. Therefore, GovOps integrates business rules and compliance constraints with operations capacities and best-practices, from early stages of designing governance strategies in order to counteract system over-regulation and lost governance requirements.

It is worth noting that GovOps does not attempt to define a general methodology for IoT cloud governance. There are many approaches (Sect. 5), which define governance models and accountability frameworks for managing governance objectives and coordinating decision making processes. Most of these approaches can be applied within GovOps without substantial modifications.

### 3.1 Governance Aspects

From our case studies, we have identified various business stakeholders such as building residents, building managers, governments, vehicle manufacturers and golf course managers. Typically, they are interested in energy efficient and greener buildings, sustainability of building assets, legal and privacy issues regarding sensory data, compliance (e.g., regulatory or social), health of the fleet, as well as security and safety issues related to the environments under their jurisdiction.

Depending on the concrete (sub)system and the involved stakeholders, governance objectives are realized via different governance strategies. Generally, we identify the following governance aspects: (i) *environment-centric*, (ii) *data-centric* and (iii) *infrastructure-centric governance*.

*Environment-centric governance* deals with issues of overlapping jurisdictions in IoT cloud managed environments. For example, in our BAS, we have residents, building managers and the government that can provide governance objectives, which directly or indirectly affect an environment, e.g., a residential apartment. In this context, we need to simultaneously articulate multiple governance objectives related to comfort of living, energy efficiency, safety, health and sustainability.

*Data-centric governance* mostly deals with implementing the governance strategies related to the privacy, quality, and provenance of sensory data. Examples include addressing legal issues, compliance, and user preferences regarding the sensory data.

*Infrastructure-centric governance* addresses issues about designing, installing, and deploying IoT cloud infrastructure. This mostly affects the early stages of introducing an IoT cloud system and involves feasibility studies, cost analysis, and risk management. For example, it supports deciding between introducing new hardware or virtualizing the IoT cloud infrastructure.

### 3.2 Operations Management Aspects

Operations managers implement various processes to manage BAS and FMS at runtime. Generally, we distinguish following operational governance aspects: (i) *configuration-centric*, (ii) *topology-centric*, and (iii) *stream-centric governance*.

*Configuration-centric governance* includes dynamic changes to the configuration models of deployed software-defined IoT cloud systems at runtime. Example processes include (a) enabling/disabling an IoT resource or capability (e.g., start/stop a unit), (b) changing an IoT capability at runtime (e.g., communication protocol), and (c) configuring an IoT resource (e.g., setting sensor poll rate).

*Topology-centric governance* addresses structural changes that can be performed on software-defined IoT systems at runtime. For example, (a) Pushing processing logic from the application space towards the edge of the infrastructure; (b) Introducing a second gateway and an elastic load balancer to optimize resource utilization; (c) Replicating a gateway, e.g., for fault-tolerance or data-source history preservation.

*Stream-centric governance* addresses runtime operation of sensor data streams and continuous queries, e.g., to perform custom filtering, aggregation, and querying of the available data streams. For example, to perform local filtering the processing logic is executed on physical gateways, while complex queries, spanning multiple data streams are usually executed on VGWs. Therefore, operations managers perform processes like: (a) Placement of custom filters (e.g., near the data source to reduce network traffic); (b) Allocation of queries to VGWs; and (c) Stream splitting, i.e., sending events to multiple VGWs.

### 3.3 Integrating Governance Objectives with Operations Processes

The examples presented in Sects. 3.1 and 3.2 are by no means a comprehensive list of IoT cloud governance processes. However, due to dynamicity, heterogeneity, geographical distribution and the sheer scale of IoT cloud, traditional approaches to realize these processes are hardly feasible in practice. This is mostly because such approaches implicitly make assumptions such as physical on-site presence, manually logging into gateways, understanding device specifics, etc., which are difficult, if not impossible, to meet in IoT cloud systems. Therefore, due to a lack of a systematic approach for operational governance in IoT systems, currently operations managers have to rely on ad hoc solutions to deal with the characteristics and complexity of IoT cloud systems when performing governance processes.

Further, Table 1 lists examples of governance objectives and according operations management processes to enforce these objectives. The first example comes from the FMS, since many of the golf courses are situated in countries with specific data regulations, e.g., the US or Australia. In order to enable monitoring of the whole fleet (as required by the manufacturer) the operations managers need to understand the legal boundaries regarding data privacy. For example, in Australia, the Office of the Australian Information Commissioner (OAIC) has issued an extensive guidance[4] as to what reasonable steps to protect personal information might include, that in practice need to be interpreted by operations managers. The second example contains potentially conflicting objectives supplied by stakeholders, e.g., building manager, end user, and the government,

[4] http://www.oaic.gov.au/privacy/applying-privacy-law/app-guidelines/.

**Table 1.** Example governance objectives and operations processes.

| | Governance objectives | Operations processes |
|---|---|---|
| 1 | Fulfill legal requirements w.r.t. sensory data in country X. Guarantee history preservation | Spin-up an aggregator gateway. Replicate VGW, e.g., across different availability zones. |
| 2 | Reduce GHG emission. User preferences regarding living comfort. Consider health regulations | Provide configuration directives for an IoT cloud resource (e.g., HVAC). |
| 3 | Data quality compliance regarding location tracking services | Choose among available services, e.g., GPS vs. GNSS (Global Navigation Satellite System) platform. |

leaving it to the operations team to solve the conflicts at runtime. The third example hints that GNSS is usually better-suited to simultaneously work in both northern and southern high latitudes. Therefore, even for these basic processes, an operations team faces numerous difficulties, since in practice there is no one-size-fits-all solution to map governance objectives to operations processes.

To address these issues, GovOps proposes a novel role, *GovOps manager*, as a dedicated stakeholder responsible to bridge the gap between governance strategies and operations processes in IoT cloud systems. The main rationale behind introducing a GovOps manager is that in practice designing governance strategies needs to involve operations knowledge about the technical features of the system, e.g., physical location of devices, configuration models, placement of queries and component replication strategies. Reciprocally, defining systems configurations and deployment topologies should incorporate standards, compliance, and legal boundaries at early stages of designing operations processes. To achieve this, the GovOps manager is positioned in the middle, in the sense that he/she continuously interacts with both business stakeholders (to identify high-level governance issues) and operations team (to determine operations capacities).

The main task of a GovOps manager is to determine suitable tradeoffs between satisfying the governance objectives and the system's capabilities, as well as to continuously analyze and refine how high-level objectives are articulated through operations processes. In this context, a key success factor is to ensure effective and continuous communication among the involved parties during the decision making process, facilitating (i) openness, (ii) collaboration, (iii) establishment of a dedicated GovOps communication channel, along with (iv) early adoption of standards and regulations. This ensures that no critical governance requirements are lost and counteracts over-regulation of IoT cloud systems. On the other side, in order to support performing runtime operations processes in IoT cloud systems, while considering system characteristics (e.g., large-scale, geographical distribution and dynamicity), GovOps proposes a set of concepts that includes: (i) central point of operation, (ii) automation, (iii) fine-grained control, (iv) late-bound policies, and (v) resource autonomy.

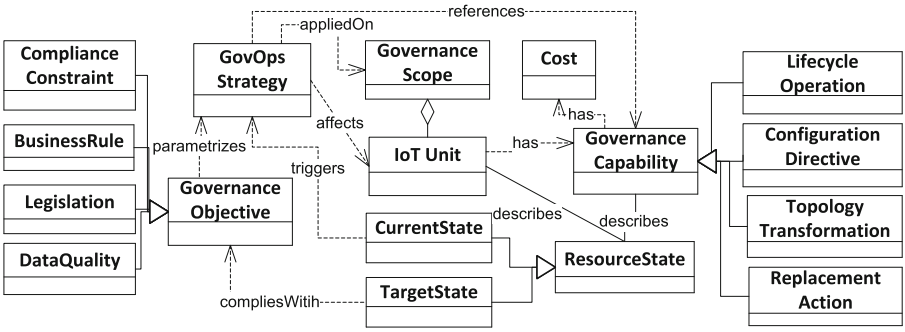


**Fig. 2.** Simplified UML diagram of GovOps model for IoT cloud governance.

## 4 A Reference Model for GovOps in IoT Cloud

### 4.1 Overview of GovOps Model for Software-Defined IoT Cloud Systems

To realize the GovOps approach we need suitable abstractions to describe IoT cloud resources that allow IoT cloud infrastructure to be (re)defined after it has been deployed. We show in [10] how this can be done with *software-defined IoT units*. The GovOps model (Fig. 2) builds on this premise and extends our previous work with fundamental aspects of operational governance processes: (i) describing states of deployed IoT resources, (ii) providing capabilities to manipulate these states at runtime, and (iii) defining governance scopes.

Within our model, the main building blocks of GovOpsStrategies are *GovernanceCapabilities*. They represent operations which can be applied on IoT cloud resources, e.g., query current version of a software, change communication protocol, and spin-up a virtual gateway. These operations manipulate IoT cloud resources in order to put an IoT cloud system into a specific (target) state. Governance capabilities are described via software-defined APIs and they can be dynamically added to the system, e.g., to a software-defined gateway. From a technical perspective, they behave like add-ons, in the sense that they extend resources with additional operational functionality. Generally, by adopting the notion of governance capabilities, we allow for processes to be automated to a great extent, and also give a degree of autonomy to IoT cloud resources.

Since the meaning of a resource state is highly task specific, we do not impose many constraints to define it. Generally, any useful information about an IoT cloud resource is considered to describe the *ResourceState*, e.g., a configuration model or monitoring data such as CPU load. Technically, there are many frameworks (e.g., Ganglia or Nagios) that can be used to (partly) describe resource states. Also configuration management solutions, such as OpsCode Chef, can be used to maintain and inspect configuration states. Finally, design best practices and reference architectures (e.g., AWS Reference Architectures[5]) provide a higher-level description of the desired target states of an IoT cloud system.

[5] http://aws.amazon.com/architecture/.

The *GovernanceScope* is an abstract resource, which represents a group of IoT cloud resources (e.g., gateways) that share some common properties. Therefore, our governance scopes are used to dynamically delimit IoT cloud resources on which a *GovernanceCapability* will have an effect. This enables writing the governance strategies in a scalable manner, since the IoT cloud resources do not have to be individually addressed. It also allows for backwards compatible GovOps strategies, which do not directly depend on the current resource capabilities. This means that we can move a part of the problem, e.g., fault and exception handling, inside the governance scope. For example, if a gateway loses a capability the scope simply will not invoke it i.e., the strategy will not fail.

### 4.2 Design Process of GovOps Strategies

As described in Sect. 3, the GovOps manager is responsible to oversee and guide the GovOps design process and to design concrete GovOps strategies. The design process is structured along three main phases: (i) identifying governance objectives and capabilities, (ii) formalizing strategy, and (iii) executing strategy.

Generally, the initial phase of the design process involves eliciting and formalizing governance objectives and constraints, as well as identifying required fine-grained governance capabilities to realize the governance strategy in the underlying IoT cloud system. GovOps does not make any assumptions or impose constraints on formalizing governance objectives. To support specifying governance objectives the GovOps manager can utilize various governance models and frameworks, such as 3P [13] or COBIT [6]. However, it requires tight integration of the GovOps manager into the design process and encourages collaboration among the involved stakeholders to clearly determine risks and tradeoffs, in terms of what should and can be governed in the IoT cloud system, e.g., which capabilities are required to balance building emission regulations and residents temperature preferences. To this end, the GovOps manager gathers available governance capabilities in collaboration with the operations team, identifies missing capabilities, and determines if further action is necessary. Generally, governance capabilities are exposed via well-defined APIs. They can be built-in capabilities exposed by IoT units (e.g., start/stop), obtained from third-parties (e.g., from public repositories or in a market-like fashion), or developed in-house to exactly reflect custom governance objectives. By promoting collaboration and early integration of governance objectives with operations capabilities, GovOps reduces the risks of lost requirements and over-regulated systems.

After the required governance capabilities and relevant governance objectives have been identified, the GovOps manager relies on the aforementioned concepts and abstractions (Sect. 4.1) to formally define the GovOps strategy and articulate the artifacts defined in the first phase of the design process. Governance capabilities are the main building blocks of the GovOps strategies. They are directly referenced in GovOps strategies to specify the concrete steps which need to be enforced on the underlying IoT cloud resources, e.g., defining a desired communication protocol or disabling a data stream for a specific region. Also in this context, the GovOps reference model does not make assumptions about

the implementation of governance strategies, e.g., they can be realized as business processes, policies, applications, or domain specific languages. Individual steps, defined in the generic strategy, invoke governance capabilities that put the IoT cloud resources into desired target state, e.g., which satisfies a set of properties. Subsequently, the generic GovOps strategy needs to be parameterized, based on the concrete constraints and rules defined by the governance objectives. Depending on the strategy implementation these can be realized as process parameters, language constraints (e.g., Object Constraint Language), or application configuration directives. By formalizing the governance strategy, GovOps enables reusability of strategies, promotes consistent implementation of established standards and best practices, and ensures operation within the system's regulatory framework.

The last phase involves identifying the system resources, i.e. the governance scopes that will be affected by the GovOps strategy and executing the strategy in the IoT cloud system. It is worth mentioning that the scopes are not directly referenced in the GovOps strategies, rather the GovOps manager applies the strategies on the resource scopes. Introducing scopes at the strategy-level shields the operations team from directly referencing IoT cloud resources, thus enables designing declarative, late-bound strategies in a scalable manner. Furthermore, at this point additional capabilities identified in the previous phase will be acquired and/or provisioned, whereas unused capabilities will be decommissioned in order to optimize resource consumption.

## 5 Related Work

The IoT governance has been receiving a lot of attention recently. For example, in [17] the author evaluates various aspects of the IoT governance, such as privacy, security and safety, ethics, etc., and defines main principles of IoT governance, e.g., legitimacy and representation, transparency and openness, and accountability. In [16], the authors deal with issues of data quality management and governance. They define a responsibility assignment matrix that comprises roles, decision areas and responsibilities and can be used to define custom governance models and strategies. Traditional IT governance approaches, such as SOA governance [1,12] and governance frameworks like CMMI [9], the 3P model [13], and COBIT [6], provide a valuable insights and models which can be applied in GovOps processes, usually without substantial modifications. Compared to these approaches, GovOps does not attempt to define a general methodology for IoT cloud governance. Therefore, such approaches conceptually do not conflict with GovOps and can rather be seen as complementary to our approach.

Also approaches addressing operations management in IoT cloud system have recently emerged. For example, in [14,18] the authors deal with IoT infrastructure virtualization and its management on cloud, whereas [2] utilizes the cloud for additional computation resources. In [15] the authors focus on operating cloud storage resources for IoT data, and [11] present approaches for monitoring IoT systems and enforcing QoS aspects. Such approaches provide useful concepts and techniques, which can be used to support the GovOps processes in

IoT cloud systems. In [7] the authors develop an infrastructure virtualization framework, based on a content-based pub/sub model for asynchronous event exchange. In [18] the authors propose virtualizing physical sensors on the cloud and provide management and monitoring mechanisms for the virtual sensors. Such approaches provide various governance capabilities, e.g., template-based controlling of sensor groups, registering and decommissioning sensors and monitoring the QoS that can seamlessly be integrated with our GovOps approach.

The GovOps model builds on these approaches and addresses the issue of bridging the gap between governance objectives and operations processes, by introducing the GovOps manager as a dedicated stakeholder, as well as defining the suitable GovOps reference model to support early integration of governance objectives and operations processes.

## 6 Conclusion and Future Work

In this paper, we introduced the GovOps approach to governance of software-defined IoT cloud systems. We presented the GovOps reference model that defines suitable concepts and a flexible process to design IoT cloud governance strategies. We introduced the GovOps manager, a dedicated stakeholder responsible to determine tradeoffs between satisfying governance objectives and IoT cloud system capabilities, and ensure early integration of these objectives with operations processes, by continuously refining how the high-level objectives are articulated through operations processes. We showed how GovOps enables systematically approaching IoT cloud governance to counteract system over-regulation and lost requirements. Further, it allows for IoT cloud governance processes to be easily and flexibly realized in practice, without worrying about the complexity and scale of the underlying IoT cloud and diversities of various legal and compliance issues. In the future, in order to support GovOps managers, we will develop a comprehensive framework for GovOps that implements the presented concepts and required toolset.

**Acknowledgments.** This work is sponsored by Pacific Controls Cloud Computing Lab (PC3L), as well as the Austrian Science Fund under grant P23313-N23 (Audit 4 SOAs).

## References

1. Charfi, A., Mezini, M.: Hybrid web service composition: business processes meet business rules. In: Second International Conference on Service-Oriented Computing - ICSOC 2004 Proceedings, pp. 30–38, New York, 15–19 November 2004
2. Chun, B., Ihm, S., Maniatis, P., Naik, M., Patti, A.: Clonecloud: elastic execution between mobile device and cloud. In: Proceedings of the Sixth European Conference on Computer systems (EuroSys 2011), pp. 301–314, Salzburg, 10–13 April 2011

3. Copie, A., Fortis, T., Munteanu, V.I., Negru, V.: From cloud governance to iot governance. In: 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA 2013), pp. 1229–1234, Barcelona, 25–28 March 2013
4. DeLoach, D.: Internet of Things: Critical issues around governance for the Internet of Things. http://tinyurl.com/mxnq3ma. Accessed on July 2014
5. European Commission: Report on the public consultation on IoT governance. http://tinyurl.com/mx24d9o. Accessed on August 2014
6. Hardy, G.: Using IT governance and COBIT to deliver value with IT and respond to legal, regulatory and compliance challenges. Inf. Sec. Techn. Report **11**(1), 55–61 (2006)
7. Hassan, M.M., Song, B., Huh, E.: A framework of sensor-cloud integration opportunities and challenges. In: Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication (ICUIMC 2009), pp. 618–626, Suwon, 15–16 January 2009
8. Lantz, B., Heller, B., McKeown, N.: A network in a laptop: rapid prototyping for software-defined networks. In: Proceedings of the 9th ACM Workshop on Hot Topics in Networks. HotNets 2010, p. 19, Monterey, 20–21 October 2010
9. Lawler, B.: Review of cmmi distilled: a practical introduction to integrated process improvement. ACM SIGSOFT Softw. Eng. Notes **30**(1), 37–38 (2005). Addison Wesley, 2004, paperback. ISBN 0-321-18613-3
10. Nastic, S., Sehic, S., Le, D., Truong, H.L., Dustdar, S.: Provisioning software-defined iot cloud systems. In: 2014 International Conference on Future Internet of Things and Cloud, FiCloud 2014, pp. 288–295, Barcelona, 27–29 August 2014
11. Nef, M.A., Perlepes, L., Karagiorgou, S., Stamoulis, G.I., Kikiras, P.K.: Enabling qos in the internet of things. In: CTRQ 2012, The Fifth International Conference on Communication Theory, Reliability, and Quality of Service, pp. 33–38 (2012)
12. Niemann, M., Miede, A., Johannsen, W., Repp, N., Steinmetz, R.: Structuring SOA governance. IJITBAG **1**(1), 58–75 (2010)
13. Sandrino-Arndt, B.: People, portfolios and processes: the 3p model of it governance. Inf. Sys. Control J. **2**, 1–5 (2008)
14. Soldatos, J., Serrano, M., Hauswirth, M.: Convergence of utility computing with the internet-of-things. In: Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012), pp. 874–879, Palermo, 4–6 July 2012
15. Stuedi, P., Mohomed, I., Terry, D.: Wherestore: Location-based data storage for mobile devices interacting with the cloud. In: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond (MCS 2010), pp. 1:1–1:8. ACM, New York (2010)
16. Weber, K., Otto, B., Österle, H.: One size does not fit all-a contingency approach to data governance. J. Data Inf. Q. **1**(1), 4 (2009)
17. Weber, R.H.: Internet of things governance quo vadis? Comput. Law Secur. Rev. **29**(4), 341–347 (2013)
18. Yuriyama, M., Kushida, T.: Sensor-cloud infrastructure - physical sensor management with virtualized sensors on cloud computing. In: The 13th International Conference on Network-Based Information Systems (NBiS 2010), pp. 1–8, Takayama, September 14–16 2010