Ann. Telecommun. (2016) 71:87–92
DOI 10.1007/s12243-016-0493-z

EDITORIAL

Middleware for Internet distribution in the context of cloud computing and the Internet of Things

Editorial Introduction

Gordon Blair[1] · Douglas Schmidt[2] · Chantal Taconet[3]

Received: 9 February 2016 / Accepted: 9 February 2016 / Published online: 26 February 2016

Middleware is software that resides between applications, services, and their underlying distributed architecture and platforms. Middleware provides several types of capabilities to developers, including providing higher-level programming abstractions to support the development of applications and services; supporting end-to-end quality attributes, such as scalability, persistence, and security; and masking problems such as system failure and heterogeneity of languages, operating systems, and networks.

Middleware has provided a key set of layers in distributed architectures since the early 90s, starting with the basic client/server model of distribution. Over the past several decades, there have been extensive innovations in middleware capabilities, with new paradigms and underlying systems technologies emerging. Significant developments during this time frame include the programming models associated with distributed objects, components, and, most recently, service-oriented architecture (and associated areas like service composition); new communication architectures focused on group-based communication, publish-subscribe, tuple-space approaches, and message queues; and underlying architectural patterns and software engineering techniques, including reflection and model-driven engineering for distributed systems (including models at runtime).

These innovations have provided developers with an abundance of tools to develop distributed systems and, until recently, middleware has largely delivered on its promise of enhancing portability, interoperability, and integrability. Over the last few years, however, there have been significant trends in distributed systems that motivate new advances in middleware to overcome limitations with conventional middleware technologies. These trends include the emergence of the Internet of Things (IoT) and cloud computing, which add new and unprecedented challenges for scale, dependability, and security to the underlying middleware technology. When taken together, the challenges are even greater when we move toward *complex distributed systems* that are very large (Internet scale), highly heterogeneous, and dynamic. There is also a corresponding change in the deployment of distributed systems, moving from individual systems to *systems of systems*.

To address the challenges for middleware emanating from the advent of the Internet of Things and cloud computing, new research is required, particularly with respect to the types of middleware needed to support key domains, such as cyber-physical systems, smart cities, the smart grid, big data analytics, digital earth, and so on. This special issue addresses these challenges by (i) considering the state-

✉ Chantal Taconet
chantal.taconet@telecom-sudparis.eu

Gordon Blair
g.blair@lancaster.ac.uk

Douglas Schmidt
schmidt@dre.vanderbilt.edu

[1] Lancaster University, Lancaster, UK

[2] Vanderbilt University, Nashville, TN, USA

[3] SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, Évry, France

of-the-art of middleware in these areas, (ii) documenting emerging ideas and concepts that help meet key challenges, (iii) increasing awareness of a grand challenge for distributed system, and (iv) galvanizing the community of researchers around this challenge.

The remainder of this editorial is organized as follows: We first examine the challenges from IoT and cloud computing individually and then consider the additional challenges emanating from emerging systems of systems; an industrial perspective on the potential and challenges of such systems is provided by *Stan Schneider, Real-Time Innovations*; then, we summarize the papers appearing in this special issue, focusing on how they address the challenges we present; finally, we conclude with a short reflection on progress that has been made in these areas and other key areas of ongoing work.

1 Middleware challenge 1: the Internet of Things

The IoT is a key step toward the ubiquitous and disappearing computing world envisioned by Marc Weiser [1] in the nineties. The IoT represents a significant extension of the current Internet to no longer be just a collection of conventional smart phones, laptops, desktops, and servers, but to greatly expand the Internet to embrace a wide range of physical objects and devices that exhibit pervasive sensing, computational, and actuation capabilities. Through the Internet, such connected Things become globally accessible—and in some cases controllable. More importantly, IoT creates opportunities to develop new types of distributed services and mass market applications in multiple domains, including smart cities, health care, transport, energy, and new forms of eCommerce.

Given its role as the universal integrator of distributed computational elements, middleware plays a key role in supporting the development of such IoT enhanced applications and services [2, 3]. However, IoT introduces significant new challenges for middleware [4] stemming from the vast number of connected objects, the volume and richness of the data produced, the rich patterns of communication required, the heterogeneity of communicating components, and new challenges in terms of quality-of-service, privacy, and security. Those challenges must be tackled by researchers and practitioners in the middleware community before society can fully benefit from this potential. We examine these middleware challenges in more detail below:

- One of the striking feature of IoT is its massive *scale*. In particular, billions of devices will be connected to the Internet, producing continuous vast flows of data. Potentially, millions of IoT services may consume data extracted from these flows, and seek to extract *actionable* information and knowledge. From the middleware perspective, this unprecedented scale brings in itself numerous issues, including naming, discovery, multi-scale distribution, routing, filtering, stability, dependability, and real-time big data analytics.
- In following the path from connected devices to end users, IoT systems involve many intermediary computers. Indeed, IoT systems include connected devices, gateways, proximity servers (Cloudlets [5]), end user (often mobile) computers, and also cloud services supporting the analysis of the resultant big data (discussed more below). IoT systems are thus complex distributed systems-of-systems. Interactions between entities may involve new forms of data-centric dissemination with complex event processing that may require new forms of event routing. For example, routing in IoT systems may be based on logical criteria, such as organization of entities, communities of interest, or geographical proximity. Moreover, these complex systems are composed of heterogeneous entities often communicating in heterogeneous ways, so portability, interoperability, and integrability are crucial challenges for middleware.
- The complexity and dynamism of IoT systems necessitates new types of deployment and management frameworks, with significant interest in autonomous styles of management. In addition, the management of IoT systems must take into account energy concerns and the potential for disconnection of devices, many of which may be limited in their power, memory, and processing capabilities.
- One often desired characteristic of IoT systems is their ability to provide open access to a wide array of raw data sources, such as sensor readings or signal detections. To achieve this, middleware must address concerns such as the privacy of the underlying data, the quality and provenance of data produced by the entities, and the safety and the trustworthiness of the IoT entities and their compositions. Finally, IoT entities are shared by many IoT services and applications, which may lead to resource conflicts and transient overloads that must be resolved by the IoT middleware.

As mentioned above, middleware should provide high level programming abstractions, support end-to-end quality attributes, and mask the inherent complexity of IoT systems, as well as the heterogeneity of their underlying components. In practice, however, additional R&D is required to achieve these basic aims in IoT domains. A number of experimental IoT middleware platforms have been developed, and some are considered in this special issue, though these platforms are still in their infancy. Moreover, applications and services are still largely developed in an ad hoc manner, often by-passing available middleware deemed not to meet IoT

requirements. The end result is that interoperability is often sacrificed and the benefits from middleware are not realized, which is not sustainable as the scale and criticality of IoT systems increase. On the positive side, there are a number of promising developments, including model-driven engineering [6], communication protocols [7], and data-centricity [8] to manage the complexity of IoT system and thus provide the desired level of abstraction to IoT service designers.

2 Middleware challenge 2: cloud computing

Cloud computing represents a second major transformational development in distributed systems. With cloud computing, there is a move toward computing as a utility, making computational resources from (physical or virtualized) infrastructure through platforms to software applications available as services in the Internet. As with IoT, there has been a corresponding burst of innovation in middleware for cloud computing [9].

A wide range of commercial cloud computing platforms are now available, offering access to resources, platforms, and services on a vast scale. Important offerings include Amazon Web Services, Microsoft Azure, and the Google App Engine. A range of open source solutions are also available, including Hadoop and OpenStack. A number of specific innovation also underpin this work, including important work on computational frameworks, such as MapReduce, in areas of eventual consistency in large-scale systems, and in novel data storage architectures [10].

In general, middleware for cloud computing is more mature than middleware for IoT, but important challenges remain. One of the key challenges is heterogeneity, which is actually increasing over time. There are few standards in cloud computing, and different vendors offer distinct, proprietary interfaces underpinned by often quiet distinct programming paradigms. A number of researchers are consequently interested in interoperability across cloud platforms. More generally, there is a growing community examining cross-cloud brokers and the transparent management of heterogeneous cloud infrastructure [11].

A number of other research challenges remain, including around multi-tenancy and its management, energy reduction, managing service level agreements, enhanced computational models to meet a range of application requirements, addressing security and privacy concerns, and so on. The vast scale of such systems also remains an important challenge underpinning much on this work. IoT and cloud integration is also a key challenge—particularly in the context of the Industrial Internet and computing clouds for cyber-physical systems—as discussed further in the section below.

The Industrial Internet of Things: Integrating Industrial Systems with the Cloud - An industrial perspective - Stan Schneider, Real-Time Innovations (RTI)

The Industrial Internet of Things (IIoT) is an emerging paradigm that applies Internet of Things (IoT) technologies in manufacturing and industrial control systems. Today, most IIoT systems do not envision control in the cloud since they are restricted to monitoring and analysis. However, this view is overly limited since future IIoT systems will combine control, monitoring, and analysis to enable three types of flows: between devices and applications deployed at the *edge*, between the edge and cloud (traditional machine-to-machine communication), and between applications in the cloud.

Traditional industrial control systems like robots, trains, or planes require data interactions on the order of milliseconds. It is not easy for conventional public clouds to participate directly in this type of transaction because network connections are too slow. However, private clouds can and should handle industrial grade transactions. Consider an ultrasound imaging system that combines real-time sensors, cloud-based image analytics, and connection with back-office IT. To be useful, the entire information trip from real-time sensor to cloud analytics and to the end-user (doctor) must be no more than a few milliseconds. As systems become faster, the entire network and cloud infrastructure will be required to support real-time data flow.

As industrial systems become increasingly complex, it is vital to build flexible architectures that are easy to update so that system integrators can compose complex systems from subsystems. Achieving this goal requires a bridging capability that provides two key functions: (1) it allows subsystems to restrict which data is "exported," or visible, to higher-level systems and (2) it includes the ability to transform the data into the likely different structures used by higher-level systems. Given the critical nature of most IIoT systems, security is a paramount concern. Traditionally, security has been a challenge because Operational Technology (OT) systems at the edge and IT system in the cloud used different connectivity technologies. Addressing this challenge requires an edge-through-cloud platform that provides an end-to-end security solution encompassing authentication, access control, and privacy.

A promising middleware technology for meeting the needs of IIoT systems described above is the Data Distribution System (DDS). DDS is an open standard that provides extensive fine control of real-time QoS parameters, including reliability, bandwidth control, delivery deadlines, liveliness status, resource limits and security. Its data-centric middleware mechanisms explicitly manage the communications data model, or types used to communicate between endpoints that can be integrated to compose complex systems from subsystems. Its security model supports fine-grained access control for data in motion without requiring a centralized message broker or Enterprise Service Bus (ESB). This design prevents intentional sabotage and hacking, as well as unintended human error that could also compromise industrial system and cause safety issues. DDS is available from multiple suppliers and is used in thousands of mission-critical IoT and IIoT systems around the world.

3 Middleware challenge 3: systems of systems

From the discussion above, it is clear that middleware faces significant challenges from the two major areas of innovation: IoT and cloud computing. Arguably, however, the biggest challenge is not in dealing with these new environments, but in managing the resultant complex distributed system. Distributed systems of the future will no longer be individual systems, but instead will be systems of systems and ultra-large-scale systems [12], which themselves will be dynamically composed. For example, in a smart city, an IoT system may be deployed to manage patterns of traffic mobility and to control traffic signalling; another IoT system may examine air quality and other key environmental factors. Moreover, a cloud environment may be used to bring together such data streams, seek correlations, and take appropriate actions. A distinct system may also be set up to provide health alerts to citizens through their mobile phones. The end result is that the distributed system very quickly becomes a large-scale system-of-systems.

Work in this area is in its infancy. A number of European projects have been established to look at systems of systems, including COMPASS [13], DANSE [14], Road2SoS [15], and T-AREA-SoS [16] but the distributed systems perspectives is not fully represented in these projects. More generally, there is insufficient interaction between the systems-of-systems community, on the one hand, and the distributed systems and middleware communities, on the other. One goal of this special issue is to help build bridges between these communities. One interesting project is Dionasys [17], which is looking explicitly at middleware for systems-of-systems. Dionasys explicitly recognizes that future distributed systems will be systems of systems and seeks programming abstractions and underlying systems principles to support the development of rich applications and services on this base.

4 This special issue

This special issue contains five articles that describe results from recent R&D attempts to provide middleware to facilitate the production of distributed applications in the context of cloud computing and the Internet of Things. The selected papers represent many of the important issues and directions for the future of middleware:

- Two of the papers provide possible solutions to the heterogeneity issue, which is particularly relevant in the context of the billions of connected objects covered by the future IoT.
- Two papers highlight the necessity of frameworks supporting the dynamic deployment of complex distributed systems with many software components geographically distributed.
- Most of the papers illustrate that middleware has evolved significantly from consideration of basic "plumbing"—how to transport requests and messages between components—to composition of services and hence how to build high level services and applications through this approach.
- All papers bring out new concerns that need to be transparently taken into account by a new generation of middleware: elasticity concerns for services executed in the cloud, and privacy and quality of service concerns more generally in the context of the IoT.

Below, we introduce the five selected papers.

The cloud computing paradigm offers different levels of service models (IaaS, PaaS, SaaS). In the article "A Simulation as a Service Cloud Middleware," Shashank Shekhar, Hamzah Abdel-Aziz, Michael Walker, Faruk Caglar, Aniruddha Gokhale, and Xenofon Koutsoukosand focus on Software as a Service (SaaS) and introduce a new style of Saas, namely a high level cloud-based Simulation-as-a-Service (SIMaaS) middleware. SIMaas is dedicated to executing simulations that may be distributed across several instances (e.g., stochastic simulations). This middleware transparently manages elasticity, with the number of simulation instances varying significantly over time to meet the required deadlines and minimize cost. The solution is based on lightweight Linux containers.

In many industries (e.g., electrical, water, and oil) control and data acquisition are realized through many thousands of control systems spread geographically. Many issues arise in the operation of these control systems: they are heterogeneous in terms of operating systems, the underlying protocols used, and the storage capabilities. Moreover, specific protocols are needed to access those systems from mobile devices, each control system has specific characteristics to be discovered on the fly, and controlling applications have to be updated dynamically when new controlling operations are needed. In the article "NERD—Middleware for IoT Human Machine Interfaces," Thaddeus Czauski, Jules White, Yu Sun, Hamilton Turner, and Sean Eade propose a middleware for the management of such control systems from users with mobile devices. The authors propose a domain specific language to define the various control interfaces. These concise interfaces are stored on the control system and discovered on the fly. The paper describes the interactions between the mobile devices and the control systems for discovering the equipment and its control interface, as well as during the control process.

The IoT offer opportunities for building new mass market application in several domains such as smart cities, smart homes, and smart transportation. For these services, millions of communicating objects produce relevant data to be distributed to millions of end users. In the article "Enhancing Context Data Distribution for the Internet of Things using QoC-Awareness and Attribute-Based Access Control," Léon Lim, Pierrick Marie, Denis Conan, Sophie Chabridon, Thierry Desprats, and Atif Manzoor consider the problem of distributing and filtering context data for applications built upon the Internet of Things. They propose middleware that enhances the publish/subscribe paradigm with properties of quality of context data and privacy preservation. This paper explores a vital area—incorporation of security and privacy concepts into data distribution services, in their case supported by negotiation of Quality of Context (QoC) as required by such applications.

In the article "CIRUS: An elastic Cloud-based framework for Ubilytics," Linh Manh Pham, Ahmed El-Rheddane, Didier Donsez, and Noel de Palma propose an abstract architecture to guide the designers of ubiquitous real-time big data analytics applications (Ubilylists) to choose and deploy appropriate services. The components in the architecture include IoT gateways, message brokers, and Big Data Analytics components. The design of such an overall architecture is a hard task: how to choose the most suitable components, how to provide an architecture that scales smoothly to the volume of data that must be handled, how to deploy the components on various levels of the infrastructure. The authors propose the deployment of a generic and elastic cloud-based PaaS platform that scales elastically to deal with the volume of data to be analyzed. The approach is demonstrated and evaluated by a use case in the area of energy management for homes.

In the article "Model-Driven Interoperability: Engineering Heterogeneous IoT and Cloud Systems," Paul Grace, Brian Pickering, and Mike Surridge present a framework to detect interoperability errors in running complex distributed systems. This framework aims to answer one key issue in distributed systems where the systems are composed of spatially and semantically decoupled services, that is semantic interoperability. This issue is essential for future applications written upon the IoT, where the systems are highly heterogeneous and dynamic, where sensor data is consumed by services totally unknown to sensor providers, and where applications are mainly designed through composition of abstract services. The aim of the proposed framework is to verify that running services in a composition are, and stay, interoperable over time. The framework validates the interoperability between services through the use of two interoperability models, in a model-driven approach: a specification model (defined by service developers), and an application model (defined by application developers that compose services).

5 Concluding remarks—a call to collaborate

The aims of this special issue (discussed above) fall into two broad categories: examining emerging ideas and capturing on the state-of-the-art, and highlighting the grand challenge around middleware and building a community around this challenge. In terms of the former, the special issue reflects the state-of-the-art rather nicely. There is a wealth of interesting results emerging in dealing with the Internet of Things and responding to the new capabilities of the cloud. Overall, though, this is not enough. Rather than seeking middleware solutions for IoT and (by definition) separate solutions for cloud computing, there is a need to address the holistic problem and this implies the community must address system of system issues as a fundamental part of the requirements. Without this, middleware will become part of the problem rather than part of the solution as disparate middleware technologies emerge for each of the different problem domains.

In terms of the latter, our dominant wish is that people read this special issue and get stimulated about the underlying challenges associated with complex distributed systems and also that different communities come together to address these challenges. Distributed systems is at the heart of many exciting developments in society and yet the underlying technologies we offer are clearly inadequate going forward. Fresh insights perspectives are required, and collaboration is at the core of making this work. We invite you to join us in this quest.

Acknowledgments The guest editors are grateful to the Annals of Telecoms for this opportunity to publish a special issue on *Middleware for Internet distribution in the context of cloud computing and the Internet of Things*. They thank all the authors for submitting quality articles to this special issue. Last but not least, they are thankful to all the members of the Program Committee for their time and expertise. Their relevant comments have played a key role in the quality of this special issue.

References

1. Weiser M (1991) The computer for the 21st Century. In: Scientific American, special issue on communications, computers, and networks 265.3, pp 66–75
2. Hughes D, Taconet C, Leriche S (eds.) (2014) Proceedings of the 1st ACM workshop on middleware for context- aware applications in the IoT, M4IoT@Middleware 2014, Bordeaux, France, December 9, 2014. ACM
3. Hughes D, Taconet C, Leriche S (eds) (2015) Proceedings of the 2nd ACM workshop on middleware for context- aware

applications in the IoT, M4IoT@Middleware 2015, Vancouver, Canada, December 8, 2015. ACM
4. Stankovic JA (2014) Research directions for the Internet of Things. In: IEEE Internet of Things Journal 1.1, pp 3–9
5. Satyanarayanan M et al (2009) The case for VM-based cloudlets in mobile computing. In: IEEE pervasive computing 8, pp 14–23. ISSN: 1536-1268
6. France RB, Rumpe B (2007) Model-driven development of complex software: a research roadmap. In: International conference on software engineering, ISCE 2007, workshop on the future of software engineering, FOSE 2007, May 23–25, 2007, Minneapolis, pp 37–54
7. Schneider S (2013) Understanding the protocols behind the Internet of Things. [Online; Accessed 29 Jan 2016]. http://electronic-design.com/iot/understanding-protocolsbehind-internet-things
8. Object-Management-Group (2015) Data Distribution Service 1.4. OMG formal/15-04-10
9. Armbrust M et al (2010) A view of cloud computing. In: Commun. ACM 53.4, pp 50–58
10. Sakr S et al (2011) A survey of large scale data management approaches in cloud environments. In: IEEE communications surveys and tutorials 13.3, pp 311–336
11. Elkhatib E, Walraven S (eds) (2014) CCB '14: Proceedings of the 2nd international workshop on CrossCloud systems. ACM, Bordeaux. ISBN: 978-1-4503-3233-0
12. Feiler P et al (2006) Ultra-large-scale systems: the software challenge of the future. Software Engineering Institute, Carnegie Mellon University
13. COMPASS consortium (2011) The COMPASS project: comprehensive modelling for advanced systems of systems. [Online; Accessed 29 Jan 2016]. http://www.compass-research.eu/
14. DANSE consortium (2011) The DANSE project: designing for adaptability and evolution in system of systems engineering. [Online; Accessed 29 Jan 2016]. http://www.danse-ip.eu/home/
15. Road2Sos consortium (2013) The Road2Sos Project: roadmap for system of systems engineering. [Online; Accessed 29 Jan 2016]. http://road2sos-project.eu/
16. T-Area-SoS consortium (2013) The T-Area-SoS project: trans-atlantic research and education agenda on systems of systems. [Online; Accessed 29 Jan 2016]. https://www.tareasos.eu/
17. Blair GS et al (2015) Holons: towards a systematic approach to composing systems of systems. In: Proceedings of the 14th international workshop on adaptive and reflective Middleware, ARM@Middleware 2015, Vancouver, BC, Canada, December 7–11, 2015, pp 5:1–5:6