Chapter 3
A Publish/Subscribe-Based Service Bus for Integrating and Streamlining Event-Driven IoT Services

ABSTRACT
In IoT scenarios, numerous things and services are connected and coordinated via distributed events. Hence, a service bus needs to be established to streamline these events to enable the efficient and stable coordination of IoT services as an integrated service system. However, without an awareness of the coordination requirements of the application, the publish/subscribe-based service bus will not be optimally utilized to deliver real-time and coherent sensor events, and at the same time, service concurrency and scalability cannot be maximally realized. In this chapter, a service-oriented publish/subscribe middleware is proposed as a base for the construction of a distributed, ultra-scale, and elastic service bus for IoT applications. In order to establish this publish/subscribe service bus, the service coordination logic is then extracted from an event-driven business process, and the coordination logic is translated into the event matching and routing functions of the publish/subscribe middleware.

INTRODUCTION
In some applications, the communication foundation needs to cooperate with the computation systems. Cloud computing in data centres is based on this idea, enabling elastic and scalable computation with high levels of

DOI: 10.4018/978-1-5225-7622-8.ch003
utility to be realised. Particularly in IoT scenarios, numerous things and real-time services need to connect and coordinate with each other, meaning that cooperation becomes very important in supporting the scalability and interconnection of IoT services. In the work of Bakken et al. (2011), a special publish/subscribe middleware was proposed for smart grids, involving an adjustment of the communication fabric for higher-layer IoT services. SDNs (O. M. E. Committee, 2012) are similar, and allow an application to control all the operations in the network. These two examples do not involve the redesign of upper applications. In the current work, the communication fabric for IoT services acts as a service bus. In order to enable a service bus and IoT service systems to cooperate, three problems must be addressed:

1. Whether the architecture of the service bus supports the higher-layer applications in utilising its fine-grained functionalities and adjusting it flexibly to adapt to changes.
2. Whether the interaction functionalities can be separated from the upper application and can be assumed by the communication fabric to realise decoupling-based scalability for the application’s systems.
3. Whether the higher-layer applications are aware of the underlying communication fabric, enabling adaptation such as adjusting their behaviour and moving.

In the same way as GridStat and SDN enabled a rethinking of the foundations of communication, the work of Li et al. (2010) and Hensa et al. (2014) involved the rethinking of higher-level applications to allow accommodation of the underlying functionality. These authors partitioned a single business process into multiple pieces, which were deployed and executed according to the states of the underlying infrastructure. However, they did not consider two-layer cooperation.

In the current work, the underlying communication fabric which works on the service (SOAP) protocol is designed, followed by the distributed service containers. These two components form the distributed publish/subscribe-based service bus for IoT services, where service routing such as addressing service endpoints and delivering service invocations is supported, and its routing and management capabilities are encapsulated in the form of application interfaces, allowing bus clients to access its functionalities. This bus is called the unified message space (UMS).
Network Optimization Using Evolutionary Algorithms in Multicast Transmission
www.igi-global.com/chapter/network-optimization-using-evolutionary-algorithms/16873?camid=4v1a

The IPv6 Protocol
www.igi-global.com/chapter/ipv6-protocol/16862?camid=4v1a