Extensible Architecture for High-Performance, Scalable, Reliable Publish-Subscribe Eventing and Notification

Krzysztof Ostrowski, Cornell University, USA
Ken Birman, Cornell University, USA
Danny Dolev, The Hebrew University of Jerusalem, Israel

ABSTRACT

Existing Web service notification and eventing standards are useful in many applications, but they have serious limitations that make them ill-suited for large-scale deployments, or as a middleware or a component-integration technology in today’s data centers. For example, it is not possible to use IP multicast, or for recipients to forward messages to others, scalable notification trees must be setup manually, and no end-to-end security, reliability, or QoS guarantees can be provided. We propose an architecture that is free of such limitations and that may serve as a basis for extending or complementing the existing standards. The approach emerges from our work on QuickSilver, a new, extremely modular and extensible platform for high-performance, scalable, reliable eventing.

Keywords: architecture; eventing; extensible; multicast; notification; publish-subscribe; reliable; scalable

INTRODUCTION

Motivation
Notification is a valuable, widely used primitive for designing distributed systems. The growing popularity of RSS feeds and similar technologies shows that this is also true at the Internet scales. The WS-Notification (Graham et al., 2004) and WS-Eventing (Box et al., 2004) standards have been offered as a basis for interoperation of heterogeneous systems deployed across the Internet. Unlike RSS, they are subscription-based and, hence, free of the scalability problems of polling, and they support proxy nodes that could be used to build scalable notification trees. Nonetheless, they embody restrictions that make them unsuitable as a middleware technology in large-scale systems:
• **No forwarding among recipients:** Many content distribution schemes build overlays within which content recipients participate in message delivery. In current Web services notification standards, however, recipients are passive (limited to data reception). For example, given the tremendous success of BitTorrent for multicast file transfer, one could imagine a future event notification system that uses a BitTorrent-like protocol for data transfer. But BitTorrent depends on direct peer-to-peer interactions by recipients.

• **Not self-organizing:** While both standards permit the construction of notification trees, such trees must be manually configured and require the use of dedicated infrastructure nodes (“proxies”). Automated setup of dissemination trees by means of a protocol running directly between the recipients is often preferable, but the standards preclude this possibility.

• **Weak reliability:** Reliability in the existing schemes is limited to per-link guarantees resulting from the use of TCP. In many applications, end-to-end guarantees are required, and often of strong flavor, for example, to support virtually synchronous, transactional, or state-machine replication. Because receivers are assumed passive and cannot cache, forward messages, or participate in multiparty protocols, even weak guarantees of these sorts cannot be provided.

• **Difficult to manage:** It is hard to create and maintain an Internet-scale dissemination structure that would permit any node to serve as a publisher or as a subscriber, for this requires many parties to maintain a common infrastructure and agree on standards, topology, and other factors. Any such large-scale infrastructure should respect local autonomy, whereby the owner of a portion of a network can set up policies for local routing, availability of IP multicast, and so forth.

• **Inability to use external multicast frameworks:** The standards leave it entirely to the recipients to prepare their communication endpoints for message delivery. This makes it impossible for a group of recipients to dynamically agree upon a shared IP multicast address, or to construct an overlay multicast within a segment of the network. Yet such techniques are central to achieving high performance and scalability, and can also be used to provide QoS guarantees or to leverage emergent technologies.

In this article, we propose a principled approach to Web service notification in large-scale systems, free of the limitations listed above, which is modular and highly extensible. The design presented here is a basis for QuickSilver (Ostrowski & Birman, 2007a; Ostrowski, Birman & Dolev, 2007b, 2007c), a novel, reliable, and extremely scalable platform for publish-subscribe eventing and notification, under development at Cornell. While this architecture is inspired by our prior work on QuickSilver, it is designed to be generic, and it is compatible, in general, with a wide range of existing protocols.

**Model**

We employ the usual terminology, where

---

*Figure 1. Publishers and subscribers register for a topic with the subscription manager*
Related Content

A Semi-Automatic Approach to Composite Web Services Discovery, Description and Invocation
[www.igi-global.com/article/semi-automatic-approach-composite-web/3051?camid=4v1a](www.igi-global.com/article/semi-automatic-approach-composite-web/3051?camid=4v1a)

DiALog: A Distributed Model for Capturing Provenance and Auditing Information
[www.igi-global.com/article/dialog-distributed-model-capturing-provenance/42107?camid=4v1a](www.igi-global.com/article/dialog-distributed-model-capturing-provenance/42107?camid=4v1a)

Web Service Business Context: The Normative Perspective
[www.igi-global.com/article/web-service-business-context/3039?camid=4v1a](www.igi-global.com/article/web-service-business-context/3039?camid=4v1a)
An Efficient Service Discovery Method and its Application
www.igi-global.com/chapter/efficient-service-discovery-method-its/59932?camid=4v1a