

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 12, 2014

K. Beck
N. Cam-Winget
D. McGrew
Cisco Systems
July 11, 2013

Using the Publish-Subscribe Model in the Interface to the Routing System
[draft-camwinget-i2rs-pubsub-sec-00](#)

Abstract

In the Publish-Subscribe model, subscribers express their interest in an event, or a pattern of events, and are subsequently notified of any event generated by a publisher that matches their registered interest. The model is well suited for communication in large-scale and loosely coupled distributed systems. This document describes how the model fits into Interface to the Routing System (I2RS) and Software Defined Networking (SDN) architectures, and analyzes its advantages, its security requirements, and its use in providing security within I2RS.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 12, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

Internet-Draft

Using Pub-Sub in I2RS

July 2013

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	4
2.	Publish-Subscribe Models	4
2.1.	Terminology	4
3.	An I2RS Publish-Subscribe Model	5
3.1.	Role of the Message Broker	6
4.	Proposed Taxonomy based on Use Cases	6
5.	Security Requirements	11
6.	Security Considerations	14
7.	IANA Considerations	15
8.	Acknowledgements	15
9.	References	15
9.1.	Normative References	15
9.2.	Informative References	15
	Authors' Addresses	16

1. Introduction

This document describes the use of a publish-subscribe (or pub-sub) model to facilitate communication and authorized access for the different types of programmatic interfaces and types of applications that may be available in the I2RS and SDN architectures. It also analyzes the advantages in both scalability and security in its use within I2RS. The security requirements of a pub-sub system are given special attention, to ensure that the system meets the requirements when it is used to provide security.

2. Publish-Subscribe Models

There are two classical publish-subscribe models: topic- or content-based [[manyfaces](#)][[pub.sub.evolution](#)]. In a topic-based model, information is exchanged through a set of predefined "topics" or subjects; where a publisher is responsible for defining the classes of information to which a subscriber registers. In a content-based model, information is only shared to a subscriber if the specific information matches the subscriber's criteria.

In some important scenarios, a publish-subscribe model has significant advantages over a client-server model. When a source generates data intermittently, a client-server model can be used by having the client poll the server. But this method suffers from overhead in both processing and bandwidth, and it introduces a latency between the time the data is generated at the source, and the time that it is transported to the client. In contrast, a publish-

subscribe model avoids the extra processing, bandwidth, and latency by establishing a channel by which the source asynchronously communicates its data.

[2.1.](#) Terminology

- o Publisher: defines an entry point or handle by which a protocol or programmatic interface and its capabilities such as its time delivery capabilities, protocol transport and security properties can be used.
- o Subscriber: defines an interested routing element or application requiring access to a protocol or programmatic interface.
- o Message Broker (MB): the authorization agent and broker used to manage the supported protocols and interfaces inclusive of their (access and transport) capabilities and manages the authorization of subscribers.

[3.](#) An I2RS Publish-Subscribe Model

Use cases and framework architectures for I2RS and SDN define the need for interfaces for acquiring information about the routing system as well as manipulating and controlling the topology and behavior of such routing system. The uses cases and frameworks described in [[I-D.amante-i2rs-topology-use-cases](#)] and [[I-D.ward-i2rs-framework](#)], respectively, describe the need for interfaces with varying capabilities. The characteristics of these capabilities include:

- o Time delivery sensitivity: interfaces or operations to be executed in the I2RS may come with different time constraints. Per [section 3.5](#) of [[I-D.ward-i2rs-framework](#)], it is necessary in some cases to define when an operation is to be handled. In particular, there are operations that initially require synchronization of state.
- o Support for multiple protocols or implementation layers: it is expected that there would be more than a single mechanism defined to acquire, manipulate and control the routing system.
- o Secure, authorized communications: as the application(s) control

the behavior of the routing system, the application must be authorized to manipulate and control the routing system, and that system must check that the application has the appropriate authorizations.

- o Support for a range of data delivery content: especially in interfaces where information (such as topology data or security monitoring or auditing data) is being conveyed, the size or amount of data to be transmitted can be very large. Conversely, interfaces that control routing may transmit very short packets.

Given the different characteristics and presence of multi-protocol support, a publish-subscribe model can be used as a means to facilitate secure authorized communications. Publishers can define the characteristics and capabilities supported by the particular interface through the message broker from which subscribers can register. Furthermore, as suggested by [\[I-D.amante-i2rs-topology-use-cases\]](#) and [\[I-D.atlas-i2rs-policy-framework\]](#), a "Policy manager" or "Policy Framework" is needed to ensure authorized communications. The message broker of a publish-subscribe model can behave as the authorizing agent and determine if a subscriber is authorized to register to specific subscriptions. Similarly, the message broker can also decide whether a publisher is authorized to provide the protocols and interfaces it is attempting to publish.

The use of the publish-subscribe pattern handles scalability issues especially given the many to many relationships. It is expected that an application will establish connections to more than one interface; similarly, an interface will be communicating with many applications. Given the many to many expected communications, the need to manage the connections and its security properties can be diminished through the use of the publish-subscribe model. The publish-subscribe model reduces the number of relationships to $[P+S]$ (where P are the number of publishers and S are the number of subscribers) versus the potential for having to manage $P*S$ relationships.

[3.1.](#) Role of the Message Broker

In using the publish-subscribe model, there is a Message Broker that mediates between the publishers and subscribers. The Message Broker

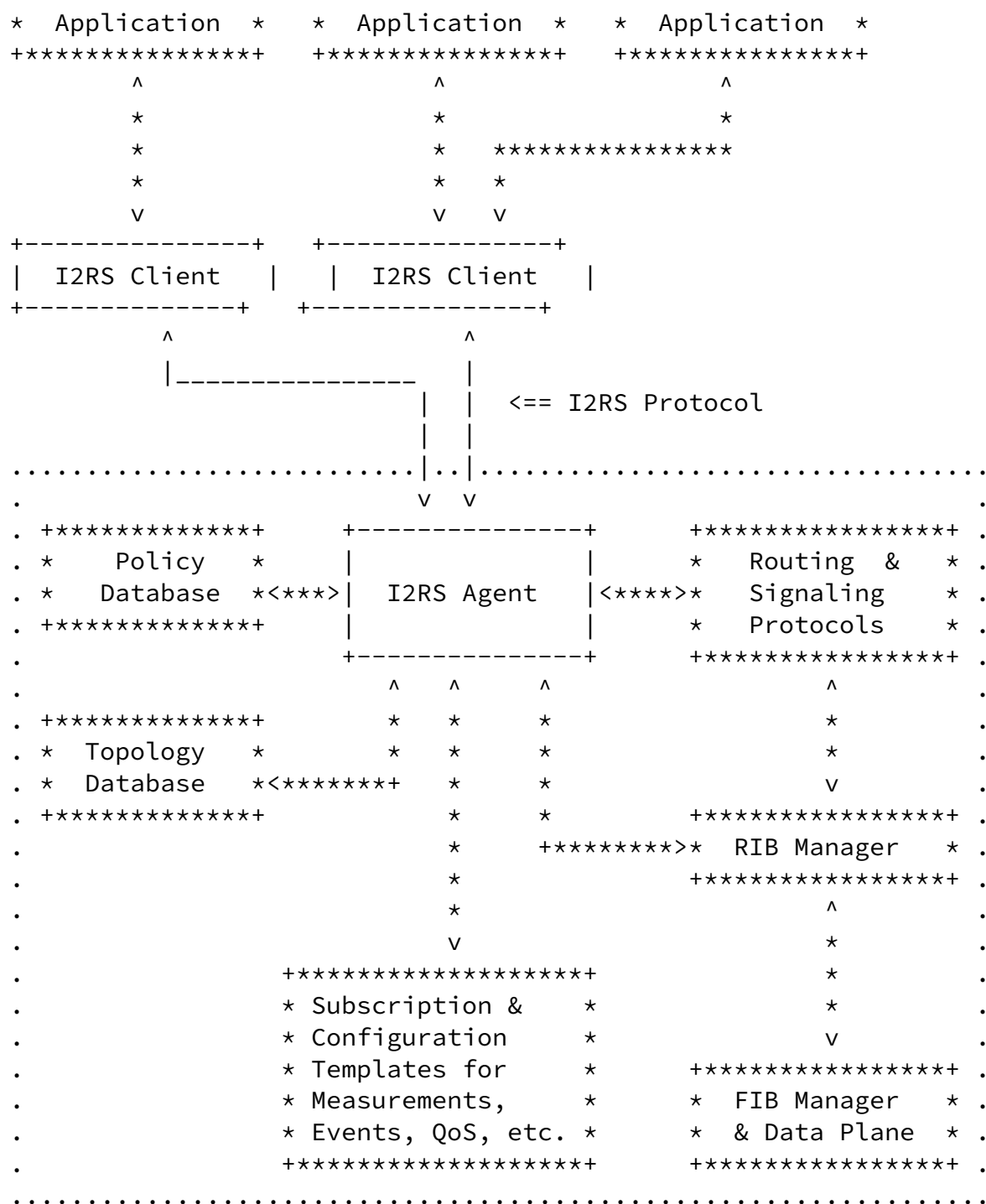
as the intermediary, allows publishers to post their information while allowing subscribers to register to the types of information it wants to receive. As the intermediary, the Message Broker can also provide filtering capabilities to allow for a publisher to post its information once but filter according to what subscribers may be interested or is authorized to receive in a subset of what a publisher may post.

In the I2RS and Software Defined Networking (SDN) use case, the Message Broker (MB) can establish the authorizations of both publishers and subscribers. When a publisher registers with the MB, the publisher and MB authenticate each other and the MB authorizes the publisher as being authoritative for a particular topic (or if the MB is not authorized, its registration is rejected). When a subscriber registers with the MB, the subscriber and MB authenticate, and the MB authorizes the subscriber to receive the topic (or its registration is rejected).

4. Proposed Taxonomy based on Use Cases

Suggested architectures of the I2RS system contains I2RS Clients [[I-D.amante-i2rs-topology-use-cases](#)] [[I-D.atlas-i2rs-problem-statement](#)] (also called Commissioners [[I-D.atlas-i2rs-policy-framework](#)]) at the application level, I2RS Agents at the network level with the I2RS protocol as the interface between the two. The specific details and nature of the Clients or Commissioners and Agents require more investigation. This discussion assumes little about them and focuses on the I2RS Protocol and how the publish-subscribe model can address many of the stated requirements and use cases, and bring additional benefits such as scalability and security.

A suggestive network architecture diagram from [[I-D.atlas-i2rs-problem-statement](#)] below:



<--> interfaces inside the scope of I2RS
 +--+ objects inside the scope of I2RS

<*> interfaces NOT within the scope of I2RS
 +*** objects NOT within the scope of I2RS

.... boundary of a router participating in the I2RS

I2RS Architectural Model

The I2RS Agent communicates with the network platform on which it resides presumably largely with a platform specific interface, i.e. CLI or REST, and with existing protocols where appropriate. There are opportunities to facilitate the publish-subscribe model within the network platform also. Here the applicability would most likely be to new areas of functionality where no existing broadly adopted standards are used; additionally where such existing standards might be extended by adoption of publish-subscribe. It is unlikely the standards themselves would be modified but rather publish-subscribed might be layered on top to better facilitate sharing of state maintained by such standards.

As suggested, the standards used would be publish-subscribed as a new layer to improve the overall system scalability and facilitate operations such as:

- o Discovery: to address the many versions of interfaces and protocols supported, a discovery mechanism may be introduced by which I2RS Agents register as publishers or subscribers. As a publisher, an interface, schema or protocol may be "advertised" with its capabilities such as the supported versions, security and data transport properties.
- o Security: it is imperative that I2RS Agents be authenticated and authorized to employ the different interfaces and protocols. To address the many-to-many relationships, the use of a publish-subscribe model as a new layer on top helps address the security requirements in a scalable manner as well.

Taxonomy is still being developed for I2RS and there is inconsistent terminology being used to date. The terminology used here and its relationship to terminologies of others is as follows:

Application

I2RS application that uses the I2RS interface to interact with the routing system. This may include a Client which actually implements the application side of the I2RS interface. This has also been called the Commissioner.

Router

The network element that supports the I2RS interface acts as a northbound API. This may include an I2RS Agent or Server.

Areas where publish-subscribe can be deployed to satisfy stated requirements and use cases:

Asynchronous Router to application notifications

In the publish-subscribe model, routers register as publishers of notifications and applications interested in receiving notifications register as subscribers. The I2RS Agent in the router need only publish a notification to publish-subscribe and is unburdened from maintaining which, if any, application is interested in the notification. Similarly, the application need only express interest in receiving notifications and is unburdened from monitoring about routers. The publish-subscribe mechanism handles the asynchronous notification connecting router notifications with interested applications. Note that an application can also act as a publisher and an I2RS agent can act as a subscriber.

Many to Many

[I-D.amante-i2rs-topology-use-cases] and [[I-D.ward-i2rs-framework](#)] both discuss the need for the I2RS interface to support multiple applications interfacing with multiple routers and the required capability of each application to be made aware of changes made by another. With publish-subscribe routers register to publish change notifications, applications register to receive change notifications and the publish-subscribe mechanism handles the change notifications connecting router notifications with interested applications.

Topology

[I-D.amante-i2rs-topology-use-cases] and [[I-D.ward-i2rs-framework](#)] discuss the requirement for applications to monitor network topology and changes to the topology whether made by devices appearing or disappearing

due device reboot or failure, modifications by other applications or by some other autonomic mechanism and the limitations of existing protocols to satisfy this requirement. Here, device agents, applications and other entities modifying topology would register as publishers of topology info, with publish-subscribe handling distributing change notifications to interested applications. This particular use case highlights the need for an initial synchronization to enable a subscriber to learn the current

network topology as well as an asynchronous method to learn the changes and updates to that topology as they occur.

RIB Updates

[I-D.ward-i2rs-framework] and others describe the need for router RIB updates to be available to I2RS applications. This is another case where routers can register as publishers of RIB changes with publish-subscribe handling the distribution of these changes to interested applications.

Policy Management

[I-D.atlas-i2rs-policy-framework] discusses the need for applications to monitor policy changes, including those made by other applications. This would be a subset of the Many to Many publish-subscribe case.

Other cases which it is clear would be well handled by publish-subscribed include Events and Configuration Changes. Use cases from [I-D.keyupate-i2rs-bgp-usecases] would include:

BGP Error Notifications. Notification of Routing Events.

BGP Protocol Statistics. Routing agents could publish BGP errors, other BGP events and BGP statistics on the router.

Tracing Dropped BGP Routes. Routers can publish learning of BGP routes which would enable applications the monitor the propagation of routes through the AS.

5. Security Requirements

This section describes security requirements as needed to sustain an I2RS or SDN. These requirements are based on the use cases defining the need for multiple protocol (using multiple layers) that need to act at different time sensitivities. It is expected that applications will need to gain appropriate authorization to use one or more of these protocols within an I2RS or SDN.

In access control models, it is common to describe access control on data in terms of the entities that are permitted to read that data, and the entities that are permitted to write that data. These models naturally apply to a publish-subscribe model: a subscriber to a topic is authorized to read data on that topic, and a publisher is authorized to write data on it. In a pub-sub model, there may be

Beck, et al.

Expires January 12, 2014

[Page 11]

Internet-Draft

Using Pub-Sub in I2RS

July 2013

many subscribers to a topic, and there may be more than one publisher on a topic as well.

Published data requires the following protections, which are stated in terms of a specific topic:

Authentication: it must not be possible for any entity other than the publisher to create a message that a subscriber will accept as authentic. It must not be possible for one subscriber to create messages that are accepted by the other subscribers to the same topic. When there are multiple publishers on a particular topic, it must be possible for the subscribers to authenticate the actual publisher of each message.

Anti-replay: if a subscriber receives the same exact message twice (e.g. because an attacker has copied and then re-injected the message), it must be detectable, and the subscriber must reject the replayed message.

Ordering: it must not be possible for any entity to re-order the authentic messages in such a way that the subscriber would accept the messages in a sequence other than the one intended by the publisher. If there are multiple publishers and the system does not ensure that they are delivered in a particular order, then subscribers (and the applications that use them) must not rely on

any particular ordering.

Confidentiality: it should not be possible for any entity other than a subscriber to read the messages.

Authenticity, anti-replay, and ordering are required, because those protections are essential to prevent the manipulation of the routing system. Confidentiality may not always be necessary, but it is strongly recommended that it be available within the pub-sub system. Anti-replay and ordering can be easily achieved whenever authentication is available, through the use of sequence numbers and/or timestamps.

There are different cryptographic techniques that can be used to provide the security services outlined above. One method is to use pairwise communications security, such as TLS or IPsec, between each subscriber and the MB and between each publisher and the MB (in the case that all communication goes through the MB). Mutual authentication between the MB and the publishers and subscribers is required. The minimum configuration that is necessary is that each publisher, and each subscriber, be configured with the information needed to authenticate the MB. Because each communication channel is separately protected, all of the needed security services are

provided and separation is enforced between different publishers and subscribers. The advantage of this method is its simplicity. It does have disadvantages when there are many subscribers and/or publishers: cryptographic operations will need to be performed for each subscriber, and if the MB is compromised, then the security of the entire system is compromised. If the MB functionality is distributed to multiple nodes in the network, that may help scalability, but at the expense of security, since it creates additional points in the system whose compromise will undermine its security.

In some cases the communication may go directly between a publisher and a subscriber, instead of through the MB. Those cases have similar advantages and disadvantages. In these cases, the MB must provide the subscribers and publishers with the information that they need to authenticate each other, and to check the authorizations of the other side of the secure communications channel. This authentication and authorization information can be provided by the

MB on a per-session basis, or it could be persistent across multiple sessions. If the data can persist for a long time, then it is important to have a method by which the MB can revoke the authorizations.

Another method is to use cryptography on the messages themselves. Digital signatures can be used to provide authentication of each message, in which each publisher has a private key, and each subscriber has the corresponding public key. Confidentiality can be provided using a group key that is shared by each publisher and the set of corresponding subscribers. This method has the advantage that cryptographic operations need only be done once on each method, thus enabling the system to scale well when there are large numbers of subscribers. It also reduces the number of keys used, and the amount of session state that needs to be maintained by the MB (or by the publishers, in the case of direct communication). The compromise of the MB does not directly compromise the entire system in this case (though if the MB is authoritative regarding which public keys should be trusted, an attacker who compromises it can always perpetrate a man-in-the-middle attack).

Security requirements using the publish-subscribe model include:

- o REQ1: Mutual authentication between the Publishers and the Message Broker, and the Subscribers and the Message Broker, is REQUIRED. Authentication to the Message Broker MUST be established as the minimum to determine authorization as either a publisher or subscriber.

- o REQ2: A Message Broker must exist to determine whether the routing element or application is authorized to access the particular protocol or interface (e.g. whether it is allowed to publish or subscribe).
- o REQ3: A Publisher SHOULD define the security properties of its protocol or program interface (e.g. how its messages are secured, using TLS for instance). Its transport SHOULD provide confidentiality and MUST provide message authentication.
- o REQ4: A Publisher SHOULD describe the latency with which it can

deliver messages, and a Subscriber SHOULD verify that the latency is acceptable. This is needed to protect applications from attacks that block the timely delivery of critical information.

- o REQ5: The I2RS interface's communication channel must provide confidentiality and message authentication.
- o REQ6: When there are multiple subscribers, it should be possible to provide cryptographic authentication in such a way that no subscriber can pose as a publisher for which it subscribed.
- o REQ7: Versioning MUST be supported. Backwards compatibility of interfaces greatly simplifies the system, but cannot always be expected. Version negotiation SHOULD be provided, and can be facilitated through the publish-subscribe layer as the Message Broker must account for the existence of multiple versions of interfaces and protocols.
- o REQ8: A discovery mechanism, when used, must be secured. At a minimum, it must be possible to configure an element with information that enables it to authenticate the provider of the discovery service, or the discovered data, and reject data from untrusted sources. A discovery service SHOULD have the ability to authenticate its clients and choose to withhold information from a client based on its authorizations.

[6.](#) Security Considerations

As the interfaces and frameworks being defined within I2RS and SDN are purposed to inform, manipulate and control topology or behavior of a routing system they must be secured through proper authentication and authorization. Section [Section 5](#) defines the security requirements to address appropriate control access, privacy and authenticity.

[7.](#) IANA Considerations

This memo includes no request to IANA.

[8.](#) Acknowledgements

The authors thank Allan Thomson and Anthony Grieco for valuable review and comments on this draft.

[9.](#) References

[9.1.](#) Normative References

[I-D.amante-i2rs-topology-use-cases]

Amante, S., Medved, J., Previdi, S., and T. Nadeau, "Topology API Use Cases", [draft-amante-i2rs-topology-use-cases-00](#) (work in progress), February 2013.

[I-D.atlas-i2rs-policy-framework]

Atlas, A., Hares, S., and J. Halpern, "A Policy Framework for the Interface to the Routing System", [draft-atlas-i2rs-policy-framework-00](#) (work in progress), February 2013.

[I-D.atlas-i2rs-problem-statement]

Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Problem Statement", [draft-atlas-i2rs-problem-statement-00](#) (work in progress), February 2013.

[I-D.ward-i2rs-framework]

Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Framework", [draft-ward-i2rs-framework-00](#) (work in progress), February 2013.

[9.2.](#) Informative References

[manyfaces]

Eugster, P., Felber, P., Guerraoui, R., and A-M. Kermarrec, "The many faces of publish/subscribe", ACM Computing Surveys, Volume 35, Issue 2, pages 114-131, 2003.

[pub.sub.evolution]

Schiper, A., "The Evolution of Publish/Subscribe

Communication Systems", Springer Volume 2584 of Lecture Notes in Computer Science, pages 137-141, 2003.

Authors' Addresses

Ken Beck
Cisco Systems

Email: kebeck@cisco.com

Nancy Cam-Winget
Cisco Systems

Email: ncamwing@cisco.com

David McGrew
Cisco Systems

Email: mcgrew@cisco.com

