

Understanding and Using the VOEvent Network

Robert B. Denny
DC-3 Dreams, SP

The VOEvent network carries VOEvent messages from authors to subscribers. It consists of a backbone of interconnected brokers, plus additional value added publishing and delivery services with their own entry and exit points. Authors originate VOEvent messages, and subscribers receive them. How they get from one to the other is the subject of this chapter.

Background

VOEvent messages carry time-sensitive information about transient astronomical events. Depending on the type of event, the time sensitivity can range from seconds to days. The network has been designed to deliver messages in seconds.

Messages originate with authors, and terminate with subscribers. There are relatively few authors and many subscribers. The VOEvent network offers a variety of ways to both publish and subscribe.

The VOEvent Backbone

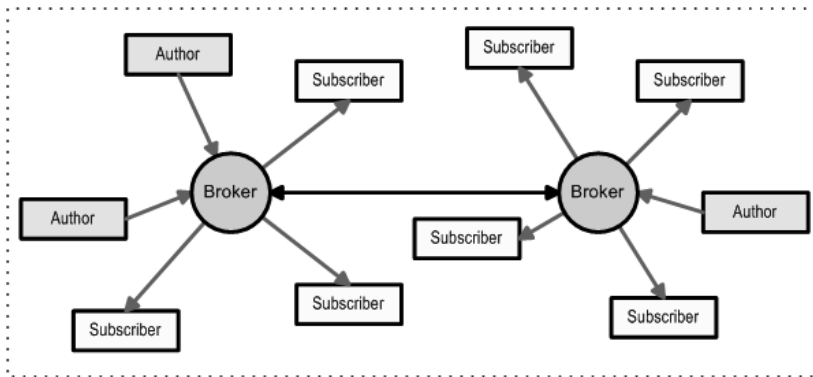


Figure 1. The VOEvent Backbone

The core component of the VOEvent network is its backbone. The backbone is formed by a set of brokers which are interconnected. Each broker provides both publish and subscribe services.

When a message is published to a broker, it is immediately distributed by that broker to all of its locally connected subscribers and to one or more other brokers to which it is interconnected. Those brokers, in turn, immediately distribute the message to their subscribers and to all of the brokers to which they are interconnected. This all happens in seconds.

The backbone uses the Internet and a TCP socket-based protocol designed for the purpose and intentionally made simple. The protocol is described in the IVOA Note VOEvent Transport Protocol 1.1 [1].

Value-Added Services – SkyAlert

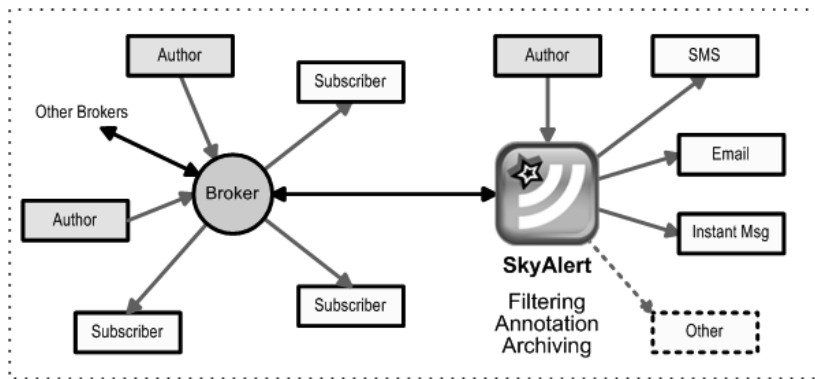


Figure 2. The Caltech SkyAlert System

The VOEvent network is not limited to the services provided by the backbone. At the California Institute of Technology, a team has created the SkyAlert system. It also accepts VOEvent messages from authors and delivers them to subscribers while acting as a broker on the backbone. What sets it apart from backbone brokers is the wide variety of value-added services it provides to subscribers. SkyAlert is described in more detail in Section 3.1 of this book.

Subscribing to the VOEvent Network

Before deciding how you want to subscribe, you should consider your present and future needs. If you want to receive personal notification of events, then SkyAlert is currently the right way to go. Be on the lookout for future systems which may also deliver personal alerts. SkyAlert provides event delivery by SMS, Email, instant messaging (XMPP), on the sky layer of the Google Earth web service, on the Microsoft Worldwide Telescope, and even Facebook. The detail contained in these postings varies widely, so be sure your chosen delivery method contains the information you need. Go to the SkyAlert home page [3] for details.

If you plan to drive a robotic observatory with event triggers, you'll probably want to subscribe directly to the backbone. You will need to implement the VOEvent Transport protocol as a subscriber. You'll also need to parse the VOEvent XML messages to extract the trigger data contained therein. For more information on this, see the section **Reading and Making VOEvent Messages** below. Many of the messages on the backbone will be of no interest, so once you have extracted the data from the XML, you'll need to discard uninteresting messages.

Publishing to the VOEvent Network

Publishers must construct VOEvent messages which are XML documents that conform to the IVOA Technical Specification *Sky Event Reporting Metadata (VOEvent)* [4]. Within this specification is a great deal of freedom to include data specific to the type(s) of events you may wish to publish, as well as external references to other VOEvent messages and supporting data for your trigger. For more information on this, see the section **Reading and Making VOEvent Messages** below.

Once you have constructed a VOEvent message, you currently have two choices for publishing. You can publish directly to the backbone, or you can publish to an instant-messaging (XMPP) service that is part of the Caltech SkyAlert system.

To publish to the backbone, you'll need to implement the VOEvent Transport protocol as a publisher per the VOEvent Transport Note [1]. Publishing to the backbone permits you to digitally sign your messages, assuring their authenticity and integrity from you to each subscriber who cares to check your signature. The signature method supported by VOEvent Transport is described in the IVOA Note *A Proposal for Digital Signatures in VOEvent Messages* [2], and referenced by the VOEvent Transport Note[1].

If you wish to publish via SkyAlert, you'll need to implement the XMPP-based Jabber protocol plus Publish/Subscribe, and have a channel assigned to you by Caltech. The XMPP protocol provided by SkyAlert can encrypt your messages, and requires you to authenticate with the server before it will accept message from you.

Reading and Making VOEvent Messages

VOEvent messages are XML documents which adhere to *Sky Event Reporting Metadata (VOEvent)*[4]. This specification and the corresponding XML schema refer to other XML specifications including *Space-Time Coordinate Metadata for the Virtual Observatory* [5] (STC). Taken as a whole, the specification is rather complex. Fortunately, in practice, the VOEvent messages in actual use represent a small subset of the possibilities.

In order to make it easier to read and make VOEvent messages, the author has written a reduced schema[7] which is suitable as input to binding tools such as JAXB, Liquid and several binding tools available for Python. By using one of these tools, all of the data in the VOEvent message is mapped to objects with corresponding properties. This approach is a real time.

There is also a public web-based VOEvent validator which will do a deep check of a VOEvent message[8]. The check is performed against the previously mentioned reduced schema. Not only is XML structure checked, but also ordering and spelling of tags plus correctness of the data in many of the elements and attributes. While developing your tools and pipelines which produce VOEvent messages, use this web site to check your messages for correctness.

Operating a Broker

If your organization has multiple publishers and subscribers on-site, or if you have a solid connection to the internet and wish to assist the VOEvent network, or if you want the visibility of having a node on the backbone for whatever reason, you may wish to operate your own broker.

Developing a conforming broker is not a simple undertaking, and it requires a skilled development team. A badly behaving or unreliable broker is a liability to your publishers and subscribers, and could adversely affect the rest of the backbone.

Available Software

This author has developed Dakota, a freely available cross-platform and open source receiver/broker and message sender package. It is described in the IVOA Note *Dakota: A Cross Platform VOEvent Broker*[6].

The lightweight broker can serve as a message receiver since it receives and stores all incoming VOEvent messages on its disk. The message sender is a separate command-line tool that reads a VOEvent message from a disk file and delivers it to a specified broker, with optional digital signature. The receiver/broker will check any digital signature it encounters. Dakota contains no tools for constructing or parsing VOEvent (XML) messages.

Dakota runs on most recent Linux distributions, a few BSD variants, Mac OS X (Leopard or later), and Windows XP/Vista/7. Feel free to contact the author for details at rdenny@dc3.com.

References

- [1] [VOEvent Transport Protocol 1.1](http://www.ivoa.net/Documents/VOEventTransport)
<http://www.ivoa.net/Documents/VOEventTransport>
- [2] [A Proposal for Digital Signatures in VOEvent Messages](http://ivoa.net/Documents/latest/VOEventDigiSig.html)
<http://ivoa.net/Documents/latest/VOEventDigiSig.html>
- [3] [Welcome to SkyAlert](http://www.skyalert.org/)
<http://www.skyalert.org/>

- [4] Sky Event Reporting Metadata (VOEvent)V1.11
IVOA Technical Specification
<http://ivoa.net/Documents/latest/VOEvent.html>
- [5] Space-Time Coordinate Metadata for the Virtual Observatory
<http://ivoa.net/Documents/latest/STC.html>
- [6] Dakota: A Cross Platform VOEvent Broker
<http://ivoa.net/Documents/DakotaBroker/>
- [7] VOEvent Mini-Schema
Documentation : <http://voevent.dc3.com/schema/>
Schema : <http://voevent.dc3.com/schema/VOEventMini-v1.1.xsd>
- [8] Online VOEvent Message Validator
<http://voevent.dc3.com/vomsgchk/>