

CERN Alerter - RSS based system for information broadcast to all CERN offices

Rafal Otto

CERN, Geneva, Switzerland

E-mail: Rafal.Otto@cern.ch

Abstract. Nearly every large organization uses a tool to broadcast messages and information across the internal campus (messages like alerts announcing interruption in services or just information about upcoming events). These tools typically allow administrators (operators) to send "targeted" messages which are sent only to specific groups of users or computers, e/g only those located in a specified building or connected to a particular computing service. CERN has a long history of such tools: CERNVMS's SPM_ "MESSAGE" command, Zephyr [2] and the most recent the NICE Alerter based on the NNTP protocol. The NICE Alerter used on all Windows-based computers had to be phased out as a consequence of phasing out NNTP at CERN. The new solution to broadcast information messages on the CERN campus continues to provide the service based on cross-platform technologies, hence minimizing custom developments and relying on commercial software as much as possible. The new system, called CERN Alerter, is based on RSS (Really Simple Syndication) [9] for the transport protocol and uses Microsoft SharePoint as the backend for database and posting interface. The windows-based client relies on Internet Explorer 7.0 with custom code to trigger the window pop-ups and the notifications for new events. Linux and Mac OS X clients could also rely on any RSS readers to subscribe to targeted notifications. The paper covers the architecture and implementation aspects of the new system.

1. History

Every organization needs to have an easy and fast way of broadcasting messages to all computers on the campus network. Such systems should be used in case of emergency where some service provider would like to inform all users about some burning issue and when service provider needs to be sure that the message has been read by the users. Very often email is used for that purpose but it is not fulfilling completely the requirements above. A user might simply have an email client switched off or just might not watch it constantly, which means that the message will not reach the user in a reasonable time (many users check email just once per day, i.e. at the end of the working day).

For several years at CERN a system called Zephyr [2] had been used in order to broadcast urgent messages around the campus. Once a message has been sent by the service provider, it was displayed in a popup window on the screen of each client computer. The system used a push mechanism of sending messages, meaning that the server had to initiate the connection with each client to whom it wanted to pass the message. With time this proprietary tool (developed in MIT) became difficult to maintain (several thousands of lines of C++ code). In addition, the number of workstations at CERN increased with time and the system started to have scalability problems.

At the end of 2004 a project was initiated in the Internet Services group of the CERN IT department to provide a replacement for Zephyr. In the summer of 2005 the new NICE Alerter based

on NNTP protocol was deployed CERN-wide on Windows workstations. Due to the lack of resources, a Linux client for the new system had never been developed and in practice Zephyr remained in production for Linux computers.

Finally, it has been decided to stop the NNTP infrastructure at CERN and a new replacement had to be found. Therefore, at the beginning of 2007 the new CERN Alerter system has been proposed, developed and deployed and the NNTP infrastructure at CERN could be phased out in July 2007. This time the solution is based on RSS (Really Simple Syndication) [9] as a transport protocol. We tried to use as much as possible out-of-the-box products and minimize custom development. Details of the implementation are covered by the section 3.

2. Functionality

The functionality of the CERN Alerter could be easily defined by the experience gained with all previous systems. The important factor for the design was the requirement to have a system based on standards so it could be easily used on multiple client platforms. Therefore, RSS has been selected as transport protocol since for all operating systems one can find many implementations of the RSS reader client. It is also included in most of the recent web browsers including Internet Explorer [6], Mozilla Firefox [8] and Safari [1].

For the Windows platform we developed a client, which is based on the RSS storage of Internet Explorer 7 and is using Internet Explorer as GUI, however it adds a pop-up functionality (see section 2.2.).

Sections: 2.1. 2.2. and 2.3. list the attributes of the message sent to the users. Most of them are standard attributes as described in the RSS 2.0 specification [9]. The rest of them are not in the standard set of RSS attributes, however, they are still compliant with the RSS specification as it allows defining custom ones. All standard RSS readers will just ignore them and interpret only those, which are standard. Using a standard RSS reader instead of a dedicated one, means that the part of functionality determining behavior of the message (see section 2.2.) will be lost but those determining content and scope of the message will remain fully functional.

2.1. Message content

Each message has several attributes determining its content. On Windows clients the message is then displayed in an Internet Explorer window where the content is visualised (see Figure 2). The following attributes are standard (as described in the RSS specification):

- Title – title of the message
- Body – body of the message, which is in HTML format
- Publication Date – date of the creation of the message
- Author – normally it is a service provider posting the message
- Enclosure – each message can have some multimedia files attached, which is displayed at the bottom of the window (e.g. the picture of shrimps on the Figure 2). It can be as well a video file, which would be played within the window once the window is popped-up. All standard RSS readers would normally show that a message has some file attached.

In addition, there is a Severity attribute, which determines the icon displayed together with the message (see top right corner of the Figure 2).

2.2. Message behaviour

The CERN Alerter Windows client distinguishes three types of messages: Alerts, Important Notices and Information Messages. Depending on the type, CERN Alerter behaves differently (see Figure 1). There are three attributes, which have define the behaviour of the message:

- Message Type – alert, important notice or information message
- Valid From Date – from that moment on the message can be seen by the users, before that time, the message is just recorded in the database and stays invisible for users

- Expiration Date – after that time message will not trigger any special action, it will just stay in the list of historical messages.

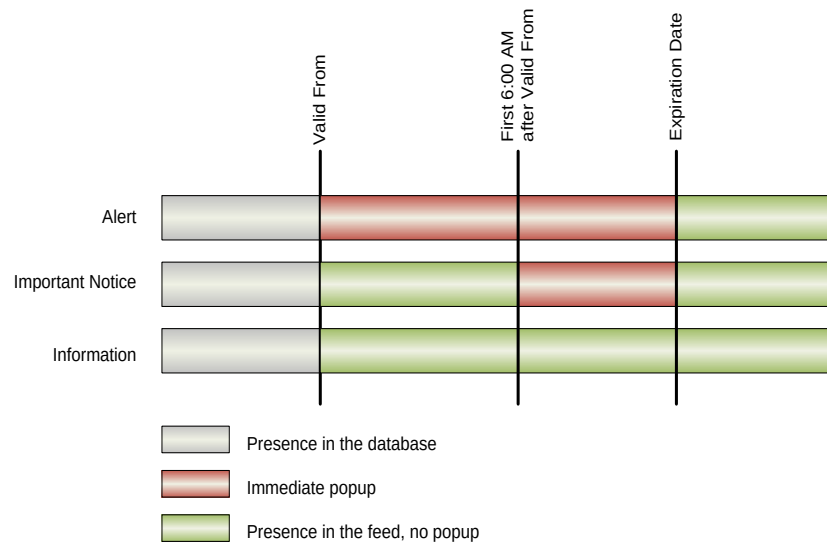


Figure 1 The message behaviour

A non-windows client (and standard RSS reader) ignores message type and the expiration date (message of any type will be displayed in the same way even if the expiration date has passed). However, the “Valid From Date” remains respected as it simply is resolved on the server side and if the message is not yet valid, it does not appear in the RSS feed, therefore is not visible for any RSS reader.

Sections: 2.2.1. 2.2.2. 2.2.3. describe therefore behavior of the message on Windows computers.

2.2.1. Alert

This type of the message should be sent in case of an urgent message, which needs to be broadcast. Typically it would be a message announcing disturbance in some service affecting many users. Once an alert message is received by the client computer within the validity period (between “Valid From Date” and “Expiration Date”), a window with an alert message is immediately displayed on the screen (see Figure 2). It is a standard Internet Explorer 7 window; therefore the content can be printed, saved as HTML etc.



Figure 2 Example of the alert message

2.2.2. Important Notice

The idea of an important notice is that a service provider needs to be sure that the message reached and has been read by the user; however it is not so urgent and the user should not be disturbed immediately.

Once an important message is received by the client computer, it first displays a notification balloon (see Figure 3). Then, the client waits until 6AM and then pops-up a window with an alert (like on Figure 2). It pops-up a window with an alert immediately after the logon as well if a logon has taken place after the closest 6AM following the “Valid From Date”.

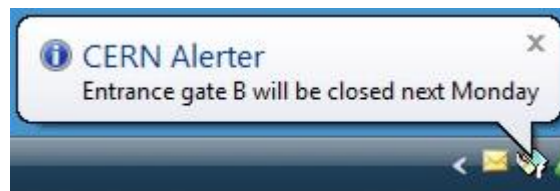


Figure 3 Notification balloon

2.2.3. Information Message

This type of message never pops-up any window. It is present in the RSS feed and therefore in the history of all the messages ever sent, although to read the message a user needs to open the history and then read it. In other words, a user is never forced to read such a message. It is intended to be used in case of non-critical announcements. Optionally each user can switch on notification balloons for information message, which will display balloons like one in Figure 3

2.3. Message scope

Any user logged on any Windows computer at CERN is automatically subscribed to the dedicated RSS feed. It means that there are as many different feeds as combinations of active users and computers. This is a task of the front-end application (see Section 3.2.) to generate those feeds out of the source feed given by the back-end (see Section 3.1.).

Each message can be therefore targeted to a group of computers or users. It means that for example a message, which is targeted to users of the IT department, appears only in the feeds of users affiliated to the IT department. In addition a constraint can be added which send a message only to users located in building 4 (see Figure 1). At the moment messages can be targeted using buildings, departments, operating system of the client computer and a few other CERN-specific attributes.

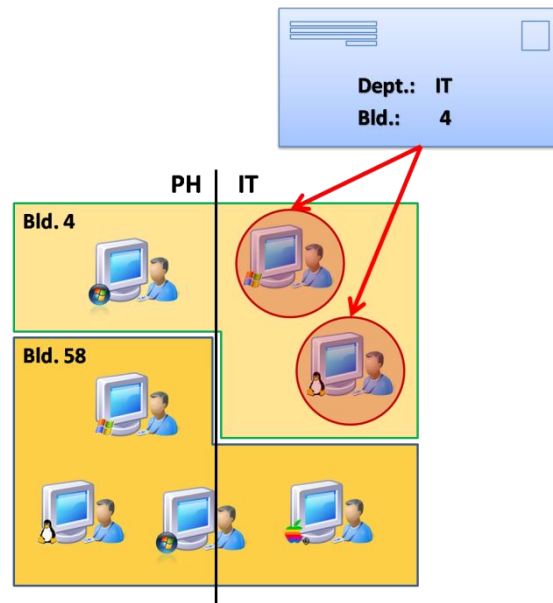


Figure 4 Message targeting

2.4. Message submission process

A message can be submitted only by the IT Manager on Duty (MOD). This authority is responsible for moderating messages coming from service providers and to give advice in the way and timing in which a message needs to be sent. Any service provider can therefore send an email to MOD requesting a message to be posted. The MOD after verification can post a message using a dedicated and protected web page [3], where all attributes of the message can be specified in a user-friendly manner.

3. Architecture

One of the most important factors determining the design of the system was the idea of minimizing custom development and relying on standard and supported software. The next sections describe the architecture, which can be seen as well on the Figure 5.

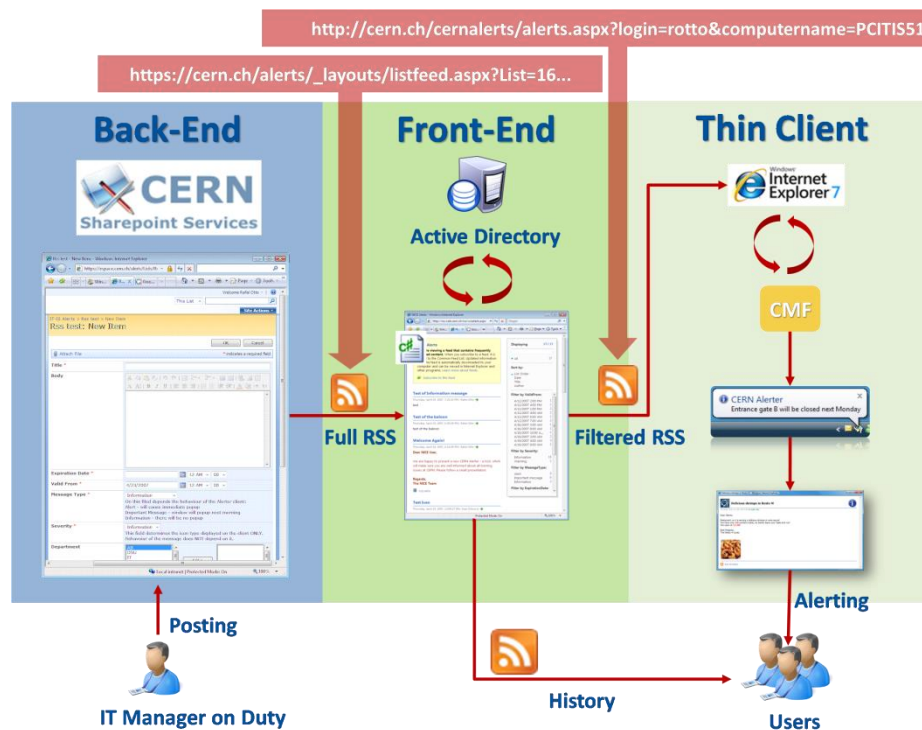


Figure 5 The Architecture of CERN Alerter

3.1. Back-End

The back-end server of the CERN Alerter system is running Microsoft SharePoint Services [7]. MS SharePoint offers both: the database for all the messages and the web interface for posting new messages. The IT Manager on Duty (MOD) visits the SharePoint site and posts a message using a standard SharePoint interface. The content of the message can be specified there as well as all the attributes determining behaviour and scope of the message. When the message is submitted, it is stored in the SharePoint database and it is exposed in form of the single RSS feed containing all messages ever posted. We call it “Full RSS Feed”.

3.2. Front-End

The front-end server is running the web application (written in C#), which provides a targeting functionality for the Full RSS Feed. As an input it takes the Full RSS Feed from the back-end server and as an output it generates RSS feeds dedicated for each combination of the user and computer name. Those feeds have all URLs in form of:

[http://cern.ch/cernalerts/alerts.aspx?login=\[USERNAME\]&computername=\[COMPUTERNAME\]](http://cern.ch/cernalerts/alerts.aspx?login=[USERNAME]&computername=[COMPUTERNAME])

Each of the feeds contains only messages which are targeted to the specific user and computer. The feeds are generated on request, so in fact the RSS feeds are generated only for those users who have active session on any of the Windows computers in the campus network.

In order to properly target messages, the application needs to check several user and computer attributes in Active Directory using the LDAP protocol. For example, if one of the messages is targeted to users or computers in building 4, the targeting application checks the location of the user and computer in Active Directory and adds a message to the final feed only if user or computer is located in the building 4.

At CERN we have in average around 3000 active computers and normally there is a single user session on each. The CERN Alerter client checks the front-end application for the new messages in the feed every 5 minutes. As a result the front-end server receives in average around 10 requests per second. For the web server it is not a lot but if for each request it would query Active Directory using

LDAP for user and computer properties, the domain controllers would be overloaded. Therefore, the front-end application uses a cache to store that information. The cache is cleaned once per day, which means that if a user changes the department he is affiliated, the change will be propagated on the alerting system within a maximum of 24 hours.

3.3. Windows Client

The CERN Alerter client makes use of the Internet Explorer 7.0 RSS store. First, it assures that the user, who is currently logged on the computer is subscribed to the dedicated RSS feed. Secondly, it periodically (each 5 minutes) forces Internet Explorer to check for new messages. Finally, it is responsible for interpreting the attributes of the messages like specified in section 2.3. 2.2. and depending on the result, it will popup a window with the alert message or display a notification balloon.

The client is implemented as a .Net 2.0 assembly in form of a DLL. It exposes methods as a plug-in module which is then consumed at CERN by the Computer Management Framework (CMF) client [4]. The CMF is a system developed in-house at CERN and is used for the desktop management and application deployment. The CMF client is running on each computer at CERN partially as a Windows service and partially as an application running in each user context. The CERN Alerter plug-in is running in an independent thread of the user-part of the CMF client.

3.4. Non-Windows Clients

The CERN alerter system is based on RSS which makes the system easily accessible from other platforms, not only Windows computers. It is enough then to install any RSS reader (all recent web browsers can act as RSS reader including Firefox [8] and Safari [1]). In order to see all messages ever sent by the central services one should then subscribe to the following RSS feed:

<http://cern.ch/cernalerts/alerts.aspx>.

As already noted in section 2.3. optionally a user name and a computer name can be added to the URL in order to filter messages and see only those concerning specified user and computer:

<http://cern.ch/cernalerts/alerts.aspx?login=MYLOGIN&computername=MYCOMPUTER>.

Acknowledgements

The author would like to thank to Ruben Domingo Gaspar Aparicio. All his effort in collecting requirements and implementing the previous NNTP-based alerter has been used as a starting point for the new system. Special thanks to Emmanuel Ormancey for his help and support.

References

- [1] Apple Inc, Safari RSS, <http://www.apple.com/macosx/features/safari/>
- [2] Belville, S.Pop, D., Zephyr at CERN, CERN CN/DCI/165, July 1993
- [3] CERN, CERN Alerter web site, <https://cern.ch/alerts>
- [4] Deloose, I., CERN, Computer Management Framework, Presentation at HEPiX 2006 Spring Workshop, April 2006, <http://hepix.caspar.it/spring2006/TALKS/7apr.deloose.nice.pdf>
- [5] Gaspar Aparicio, R. D., CERN, Nice Alerter Presentation, May 2005, <https://cern.ch/NICE-Techmeetings/Document%20Library/1/2005/2005-05-02%20NewsREader2.ppt>
- [6] Microsoft Corp., Internet Explorer 7 – RSS Feeds Quick Tour, <http://www.microsoft.com/windows/IE/ie7/tour/rss/>
- [7] Microsoft Corp., Windows SharePoint Services TechCenter, <http://www.microsoft.com/technet/windowsserver/sharepoint/>
- [8] Mozilla Corp., Live Bookmarks, <http://www.mozilla.com/en-US/firefox/livebookmarks.html>
- [9] Winer, D., RSS 2.0 Specification, July 2003, <http://cyber.law.harvard.edu/rss/rss.html>