Calendaring Interoperability over HTTP (CIH)

Abstract

Calendaring Interoperability over HTTP (CIH) is a technique for exchanging calendaring information between scheduling systems using the Hypertext Transfer Protocol (HTTP). This allows users to schedule meetings with anyone else, no matter what scheduling software they use.

This document is a tentative exploration of whether CIH is feasible and what the pros and cons are relative to a brand new protocol like the Calendaring Interoperability Protocol (CIP). The primary benefit of CIH over CIP and motivation for this project is that we avoid implementing a whole new protocol, such as writing and debugging the protocol, convincing vendors to write new code to support it, and then convincing users to deploy it.

Table of Contents

1.0 Introduction

The Internet Calendaring and Scheduling Core Object Specification (iCalendar) [1] describes a way to represent calendaring and scheduling objects and verbs as MIME objects. This includes distributing and responding to events and todos and sending free time queries and responses. One of the most interesting parts of iCalendar(and probably one of the best) is that these objects can include a Profile property that indicates what action should be performed (proposing an event, cancelling one, changing one, responding to one, etc.). This allows the objects to be sent

around by email and transferred into scheduling systems, either manually or by integrating the scheduling systems with the email system.

However, there are many times when email is not an ideal transport for calendaring and scheduling. For instance, email delivery can be slow and sometimes unreliable. Gateways can mangle messages. Users can fail to check their email for long periods of time. Most email systems will require manual effort to transfer scheduling info from email to the scheduling system and back. Many scheduling operations can and should be performed better with a direct, real-time connection between scheduling systems.

For these reasons, a protocol that allows iCalendar objects to be exchanged in real time would be ideal. A simple protocol will do. We're just exchanging MIME objects. The details of the object format and the actions implied by the objects (date syntax, attendee list, etc.) are considered in the iCalendar spec.

Several protocols have been suggested to address this problem. The CIP (Calendaring Interoperability Protocol) [2] is a simple protocol with a few verbs that correspond to the operations necessary to accomplish calendaring interoperability. HTTP is also being considered for this role and that is the focus of this document.

This document provides a tentative exploration of whether Calendaring Interoperability over HTTP (CIH) is feasible and what the pros and cons are relative to a brand new protocol like the CIP.

2.0 Overview of CIH

CIH uses the overall architecture of CIP, but reimplements it using a subset of the HTTP 1.1 protocol specification [3]. CIH servers are HTTP servers and therefore must meet all requirements for HTTP servers laid out in the HTTP 1.1 specification. CIH is merely a convention which describes how certain HTTP commands should be handled when certain MIME content types are used.

2.1 Performing Basic CIH Operations

All CIH operations are performed by using the HTTP POST command to send MIME objects from the client to a specified URL on the server and receive MIME objects in the response. The MIME objects in question are those described by the iCalendar spec and their semantics and syntax are described thoroughly in that document.

HTTP is used in this context mostly as a real-time transfer mechanism for MIME objects. Since its capabilities are a superset of those provided by SMTP email, this does not require significant changes to the iCalendar spec.

2.2 Mapping an Attendee Address to a URL for CIH

The default value of an attendee as defined in iCalendar is an RFC 822 address (that is, user@hostname). While iCalendar allows the value to have other data types, such as URL, it is desirable to have an easy way to map a single RFC 822 address into a URL that can be used for CIH. For the address hanna@world.std.com, the CIH URL is http://cih.world.std.com/hanna. The algorithm is to prepend cih. to the host name and use the user name as an abs_path. If this generates an invalid URL or an error occurs when CIH communications are attempted, email communication SHOULD be used.

A few notes here. First, this does not mean that all URLs that support CIH must have this form. It just provides a convenient method for mapping an RFC 822 address into an HTTP URL for use with CIH. Second, if a better scheme emerges for server location (SLP or whatever), we can support that and provide backward compatibility with this scheme via redirects.

3.0 Pros and Cons of CIH (vs CIP)

The primary benefit of CIH over CIP and the motivation for this project is that we avoid implementing a whole new protocol, such as writing and debugging the protocol,

convincing vendors to write new code to support it, and then convincing users to deploy it. There are three phases to this benefit: definition, implementation, and deployment.

Some would argue that defining CIH will be easier than defining a whole new protocol. This is not clear, as there is also a counter-argument that HTTP is so complicated that defining CIH will be quite a task.

For vendors with lots of HTTP experience and code, deployment of CIH will be easier than CIP. For vendors without much HTTP experience, CIH will be more complex than CIP and probably take more time to implement, even if HTTP code can be licensed and integrated into their products.

For deployment, the balance is clearly in favor of CIH. Users and administrators are already familiar with HTTP and a great deal of infrastructure has been built up to support HTTP already. Problems may arise, however, if CIH uses a subset of HTTP or takes advantage of obscure or cutting-edge features of HTTP that aren't widely implemented. These is also a risk of customer backlash against using HTTP for a purpose different from its original intent.

3.1 Performance Concerns

Concerns have been raised that CIH will be substantially slower than CIP. This section provides some of the arguments pro and con.

While HTTP 1.0 is a connectionless protocol, HTTP 1.1 uses persistent connections by default. This avoids the overhead associated with bringing up and shutting down TCP connections all the time.

Although HTTP 1.1 uses persistent connections, it is still a stateless protocol in that each request is considered without regard to the requests that preceded it (unless cookies are used). The stateful nature of CIP can allow it to optimize some operations by establishing a state once and then performing several commands. However, it is not clear that this will provide substantial performance improvements.

In the area of authentication and encryption, SSL avoids reauthentication with every request.

Several HTTP features, such as compression via content codings may provide sophisticated methods of boosting throughput and therefore performance over CIP.

4.0 Security Considerations

Since calendaring interoperability may involve the exchange of sensitive information, any calendaring interoperability mechanism must provide for encryption and authentication. Several techniques such as SSL and HTTP Access Authorization are available for use with HTTP as described in [3] and [4]. Without such techniques, CIH clients and servers are wide open to forged or snooped meeting proposals or responses.

5.0 References

[1] F. Dawson, D. Stenerson, _Internet Calendaring and Scheduling Core Object Specification_, Internet Draft, draft-ietf-calsch-ical-00.txt, February 1997.

[2] A. Ganguly, S. Hanna, P. O'Leary, B. Spencer, _Calendaring Interoperability Protocol_, Internet Draft, draft-ietf-calsch-cip-00.txt, November 1996.

[3] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, _Hypertext Transfer Protocol_, RFC 2068.

[4] A. Frier, P. Karlton, P. Kocher, _The SSL Protocol Version 3.0_, Internet Draft, draft-ietf-tls-ssl-version3-00.txt, November 1996.

6.0 Open Issues

Define iCalendar objects to be used in CIH responses (other than FREE-BUSY/REPLY). This will have the benefit of

allowing responses (i.e. operation confirmations) by email if that's desired.

Should have a non-destructive way to confirm that a given URL supports CIH before you start PUTting things to it.

Otherwise, you might just be dumping things in a directory on an HTTP server. Perhaps a GET that is expected to return a certain value.

7.0 To Be Done

Add examples and lots more detail.

8.0 Author's Address

Steve Hanna
On Technology Corporation
hanna@world.std.com
(617)
692-3153

Anik Ganguly
OnTime
Campbell Services Inc.

http://www.ontime.com