Guide to Implementors
Network Working Group
Internet-Draft
<draft-ietf-calsch-imp-guide-00.txt>
4-Oct-99
Expires: <date + 6 months>

Bob Mahoney/MIT
Alexander Taler/CS&T

Implementors' Guide to Internet Calendaring

Status of this Memo

Abstract

This document describes the relationship between the various internet calendaring and scheduling protocols defined by RFC 2445 (iCalendar), RFC 2446 (iTIP), and RFC 2447 (iMIP), as well as the works in progress,"iCalendar Real-time Interoperability Protocol" (iRIP), and "Calendar Access Protocol" (CAP). It's intention is to provide a context for these protocols, assist in their understanding, and ultimately help implementors in the design of their internet calendaring and scheduling systems.

This document also describes issues and problems which are not solved by these protocols, and could be targets for future work.

1. Introduction

The calendaring and scheduling protocols are intended to provide for the needs of individuals attempting to obtain information and schedule meetings across the internet, organizations attempting to provide information on the internet, as well as organizations looking for a calendaring and scheduling solution to deploy internally.

It is the intent of this document to provide guidance for implementors of calendaring and scheduling products in determining which of the various existing protocol documents are applicable to their work, as well as providing some background information and pointers to the less obvious implications of the available choices.

Problems not solved by these protocols, as well as security issues to be kept in mind, are discussed at the end of the document.

1.1 Terminology

This memo uses much of the same terminology as [ICAL], [ITIP], [IMIP], [IRIP] and [CAP]. The following definitions are provided as introductory, the definitions in the protocol specifications are the canonical ones.

Calendar
: A collection of events, todos, journal entries, etc. A calendar could be the content of a person's or a resource's agenda; it could also be a collection of data serving a more specialized need. Calendars are the basic storage containers for calendaring information.

Calendar Access Rights
A set of rules for a calendar describing who may perform which operations on that calendar, such as reading and writing information.

Calendar Service
A running server application which provides access to a collection of calendars.

Calendar Store
A data store of a calendar service. A calendar service may have several calendar stores, and each store may contain several calendars, as well as properties and components outside of the calendars.

Calendar User
An entity (often a human) which accesses calendar information.

Calendar User Agent (CUA)
Software used by the calendar user which communicates with calendar services to provide the user access to calendar information.

Component
A piece of calendar data such as an event, a todo or an alarm. Information about components is stored as properties of those components.

Property
A property of a component, such as a description or a start time.

2. Requirements

2.1 Fundamental Needs

The following examples illustrate people's basic calendaring and scheduling needs:

a] A busy musician wants to maintain her schedule on an internet-based agenda which she can access from anywhere.

Need: Read and manipulate one's own calendar.

b] A software development team wishes to share agenda information by using a group scheduling product in order to more effectively schedule their time.

Need: Share calendar information with users using the same calendar service.

c] A teacher wants his students to be able to book time slots

during his office hours.

Need: Schedule calendar events and todos with users using the same calendar service.

d] A movie theatre wants to publish its schedule so that prospective customers can easily access it.

Need: Share calendar information with users using other calendar services, possibly from different vendors.

e] A social club wants to be able to organise events more effectively by booking time with its members.

Need: Schedule calendar events and todos with users using other calendar services, possibly from different vendors.

2.2 Protocol requirements

The first three needs can be satisfied through proprietary solutions, but the last two cannot. From these needs we can establish that protocols are required for accessing information in a calendar store, and for scheduling events and todos. In addition these protocols require a data format for representing calendar information.

These roles are filled by the following protocol requirements.

- [ICAL] is the data format

  [ICAL] provides data format for representing calendar information which the other protocols can use. [ICAL] can also be used in other contexts such as a drag and drop format or an export/import format.

  All the other protocols depend on [ICAL], so all elements of a standards-based calendaring and scheduling systems will have to interpret [ICAL].

- [ITIP] is the scheduling protocol

  [ITIP] describes the messages used to schedule calendar events. These messages are represented in [ICAL], and have semantics that include such things as being an invitation to a meeting, an acceptance of an invitation or the assignation of a task.

  [ITIP] messages are used in the scheduling work flow, where users exchange messages in order to organize things such as events and todos. CUAs generate and interpret [ITIP] messages at the direction of the calendar user.

  [ITIP] is transport-independent, but has two specified transport

bindings, [IMIP] is a binding to email and [IRIP] is a real-time binding. In addition [CAP] will provide a second real-time binding of [ITIP], allowing CUAs to perform calendar management as well as scheduling over a single connection.

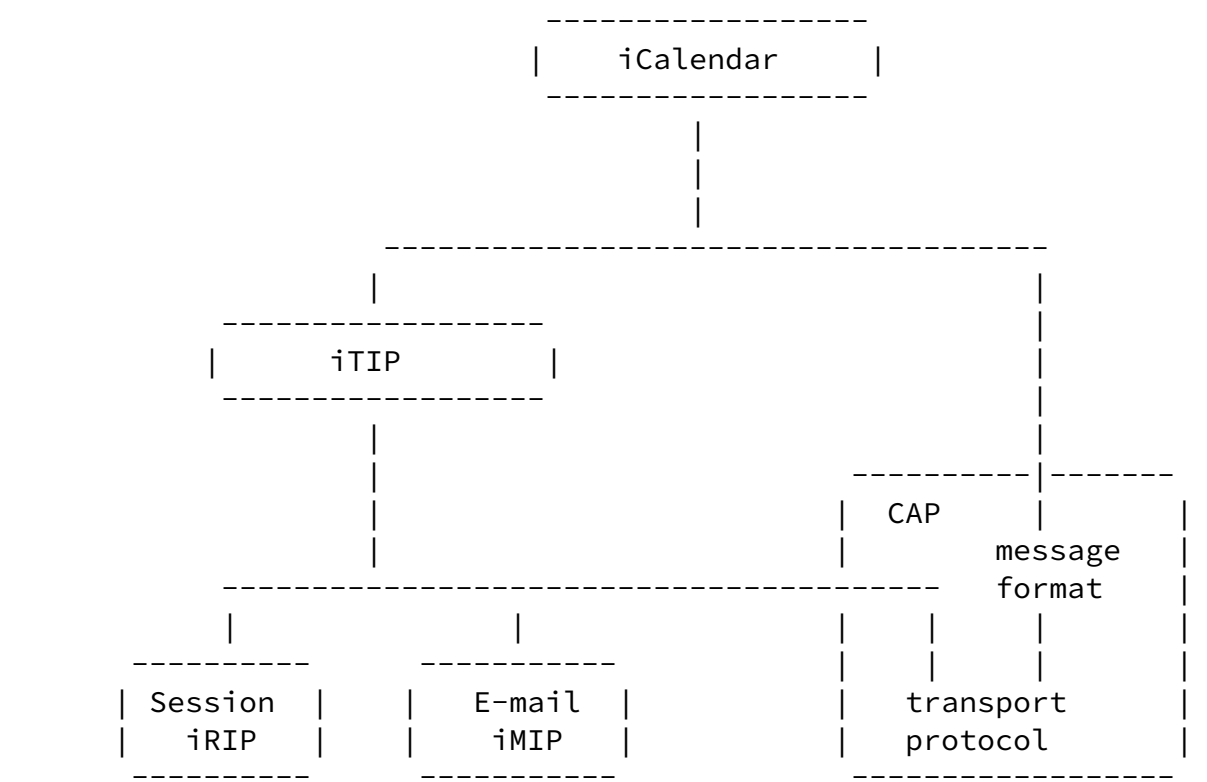
Both CUAs and calendar services may have [ITIP] interpreters.

- [CAP] is the calendar management protocol

  [CAP] describes the messages used to manage calendars. These messages are represented in [ICAL], and have semantics such as being a search for data, being data in response to a search or the being the creation of a meeting.

  [CAP] also provides a real-time binding for the calendar management messages. Although other bindings, such as an email binding, could be defined, this is not done because it is inappropriate for this protocol.

The following diagram describes the implementation dependencies between the protocols. A calendar system using these standards will implement at least one of the leaves of the tree. The calendar management message and transport protocol parts of CAP are separated in the diagram to highlight its relationship to ITIP.



3. Solutions

3.1 Examples

Returning to the examples of section 2.1, they can be solved using the protocols in the following ways:

a] The musician who wishes to access her agenda from anywhere can use a [CAP] enabled calendar service accessible through the internet. She can then use whichever [CAP] clients are available to access the data.

A proprietary system could also be employed which provides access through a web-based interface, but the use of [CAP] would be superior in that it would allow the use of third party tools, such as PDA synchronization tools.

b] The development team can use a calendar service which supports [CAP] and then each member can use a [CAP]-enabled CUA of their choice.

Alternatively, each member could use an [IMIP]-enabled CUA, and they could book meetings over email. This solution has the drawback that it is difficult to examine the other agendas, making organizing meetings more difficult.

Proprietary solutions are also available, but they require that all people use clients by the same vendor, and disallow the use of third party applications.

c] The teacher can set up a calendar service, and have students book time through any of the [ITIP] bindings. [CAP] or [IRIP] provide real-time access, but could require additional configuration. [IMIP] would be the easiest to configure, but may require more email processing.

If [CAP] access is provided then determining the state of the teacher's schedule is straightforward. If not, this can be determined through [ITIP] free-busy requests. Non-standard methods could also be employed, such as serving up ICAL, HTML, XML through HTTP.

A proprietary system could also be used, but would require that all students be able to use software from a specific vendor.

d] For publishing a movie theatre's schedule [CAP] provides the most advanced access and search capabilities. It also allows easy integration with its customer's calendar systems.

Non-standard methods such as serving data over HTTP could also be employed, but would be harder to integrate with customer's systems.

Using a completely proprietary solutions would be very difficult since it would require every user to install and use proprietary

software.

e] The social club could distribute meeting information in the form of [ITIP] messages. This could be done over email using [IMIP], or [IRIP] depending on the recipient. Meeting invitations, as well as a full published agenda could be distributed.

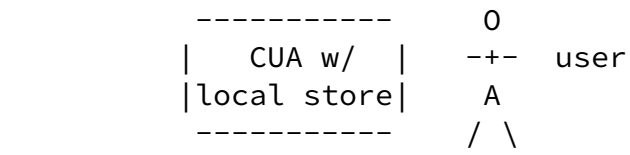
Alternatively, the social club could provide access to a [CAP] enabled calendar service, however this solution would be more expensive since it requires the maintenance of a server.

3.2 Systems

The following diagrams illustrate possible example systems and usage of the protocols. [ed. More coming]

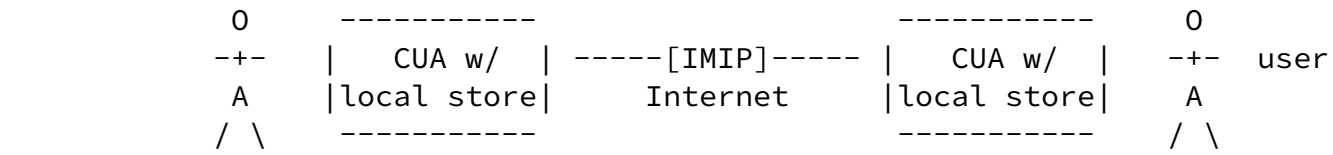
3.2.1 Standalone single-user system

A single user system which does not communicate with other systems need not employ any of the protocols. However, it may use [ICAL] as a data format in some places.



3.2.2 Single-user systems communicating

Users with single-user systems may schedule meetings with each other using [ITIP]. The easiest binding of [ITIP] to use is [IMIP], since it messages can be held in their mail queue, which we assume to already exist. [IRIP] or [CAP] would require at least one user to run a listening server.



4. Open Issues

Many issues are not currently resolved by these protocols, and many desirable features are not yet provided. Some of the more prominent ones follow.

4.1 Scheduling people, not calendars

Meetings are scheduled with people, however people may have many calendars, and may store these calendars in many places. There may

also be many routes to contact them. These protocols do not attempt to provide unique access for contacting a single person. Instead, 'calendar addresses' are booked, which may be email addresses or individual calendars. It is up to the users themselves to orchestrate mechanisms to ensure that the bookings go to the right place.

4.2 Administration

These protocols do not address the issues of administering users and calendars on a calendar service. This must be handled by proprietary mechanisms for each implementation.

4.3 Notification

People often wish to be notified of upcoming events, new events, or changes to events. These protocols do not attempt to address these needs in a real-time fashion. Instead, the ability to store alarm information on events is provided, which can be used to provide client-side notification of upcoming events. To organize notification of new or changed events clients will have to poll the data store.

5. Security considerations

5.1 Access Control

There has to be reasonable granularity in the configuration options for access to data through [CAP], so that what should be released to requestors is, and what shouldn't isn't. Details of handling this are described in [CAP].

5.2 Authentication

Access control must be coupled with a good authentication system, so that the right people get the right information. For [CAP] this means requiring authentication before any data base access can be performed, and checking access rights and authentication credentials before releasing information. In [IMIP], this may present some challenges, as authentication is often not a consideration in store-and-forward protocols.

Authentication is also important for scheduling, in that receivers of scheduling messages should be able to validate the apparent sender. Since scheduling messages are wrapped in MIME, signing and encryption is available for free. For messages transmitted over mail this is the only available alternative. It is suggested that developers take care in implementing the security features in [IMIP], bearing in mind that the concept and need may be foreign or non-obvious to users, yet essential for the system to function as they might expect.

The real-time protocols provide for the authentication of users, and the preservation of that authentication information, allowing for validation by the receiving end-user or server.

5.3 Using email

Because scheduling information can be transmitted over mail without any authentication information, email spoofing is extremely easy if the receiver is not checking for authentication. It is suggested that implementors consider requiring authentication as a default, using mechanisms such as are described in Section 2 of [IMIP].

The use of email, and the potential for anonymous connections, means that 'calendar spam' is possible. Developers should consider this threat when designing systems, particularly those that allow for automated request processing.

5.4 Other issues

The current security context should be obvious to users. Because the underlying mechanisms may not be clear to users, efforts to make clear the current state in the UI should be made. One example is the 'lock' icon used in some web browsers during secure connections.

6. Acknowledgements

Thanks to the following who have participated in the development of this document:

Eric Busboom, Pat Egen, David Madeo, Shawn Packwood.

7. Bibliography

[ICAL] [RFC-2445] Calendaring and Scheduling Core Object Specification
[ITIP] [RFC-2446] iCalendar Transport-Independent Interoperability Protocol
[IMIP] [RFC-2447] iCalendar Message-Based Interoperability Protocol
[IRIP] draft-ietf-calsch-irip iCalendar Real-time Interoperability Protocol
[CAP] draft-ietf-calsch-cap Calendar Access Protocol

[RFC-1847] Security Multiparts for MIME
[RFC-2045] MIME Part 1: Format of Internet Message Bodies
[RFC-2046] MIME Part 2: Media Types
[RFC 2047] MIME Part 3: Message Header Extensions for Non-ASCII Text
[RFC-2048] MIME Part 4: Registration Procedures
[RFC-2049] MIME Part 5: Conformance Criteria and Examples

8. Author's Addresses

Alexander Taler

CS&T
3333 Graham Boulevard, 5th Floor
Montreal, QC H3R 3L5
Tel: (514) 733-8500
Email: alext@cst.ca

Bob Mahoney
MIT
E40-327
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-0774
Email: bobmah@mit.edu

9. Full Copyright Statement